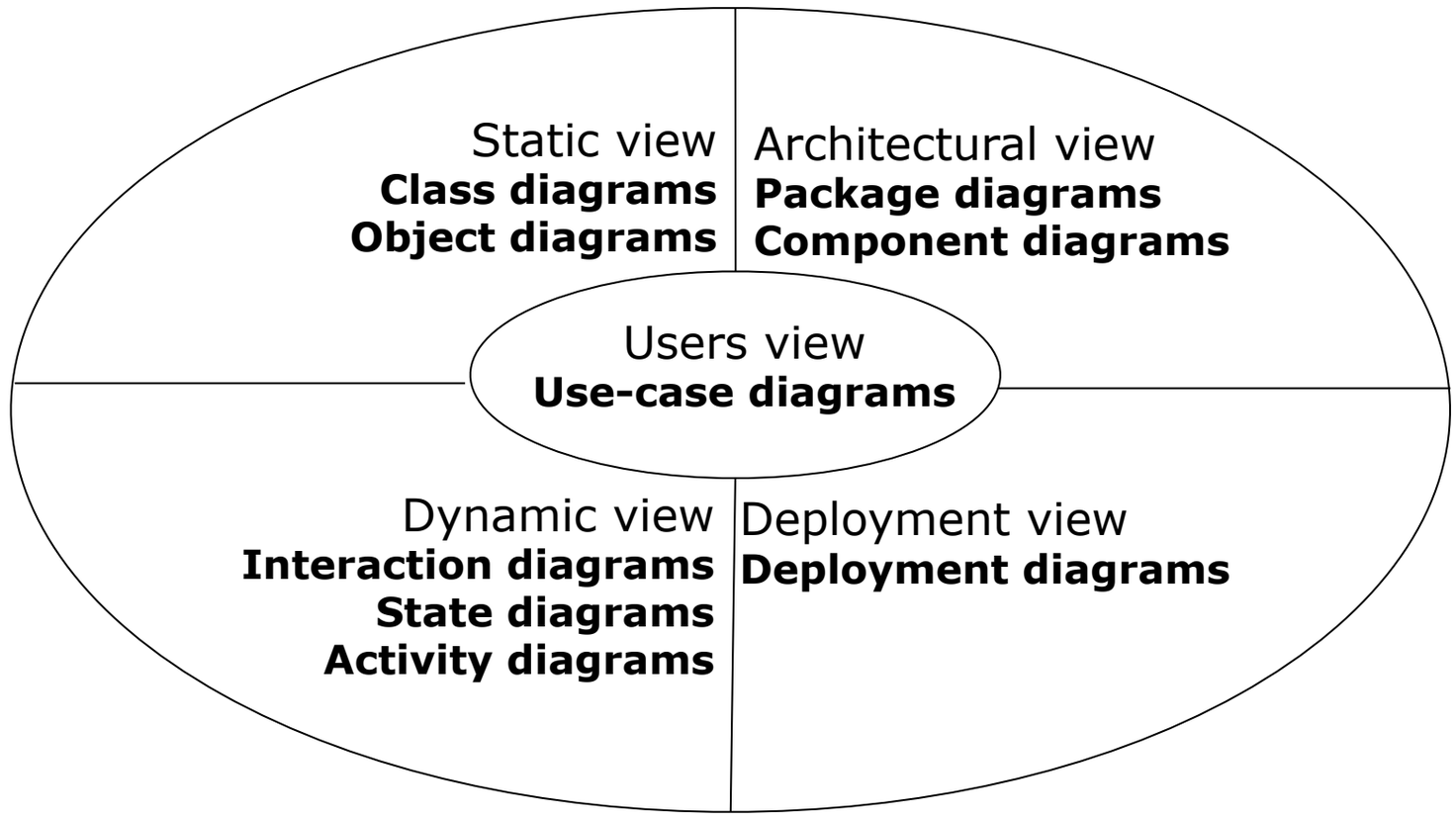
Requirement modelling

- Use-case diagrams



Software Development Activities

Requirements Gathering

Define requirement specification

Analysis

Define the conceptual model

Design

Design the solution / software plan

Implementation

Code the system based on the design

Integration and Test

Prove that the system meets the requirements

Deployment

Installation and training

Maintenance

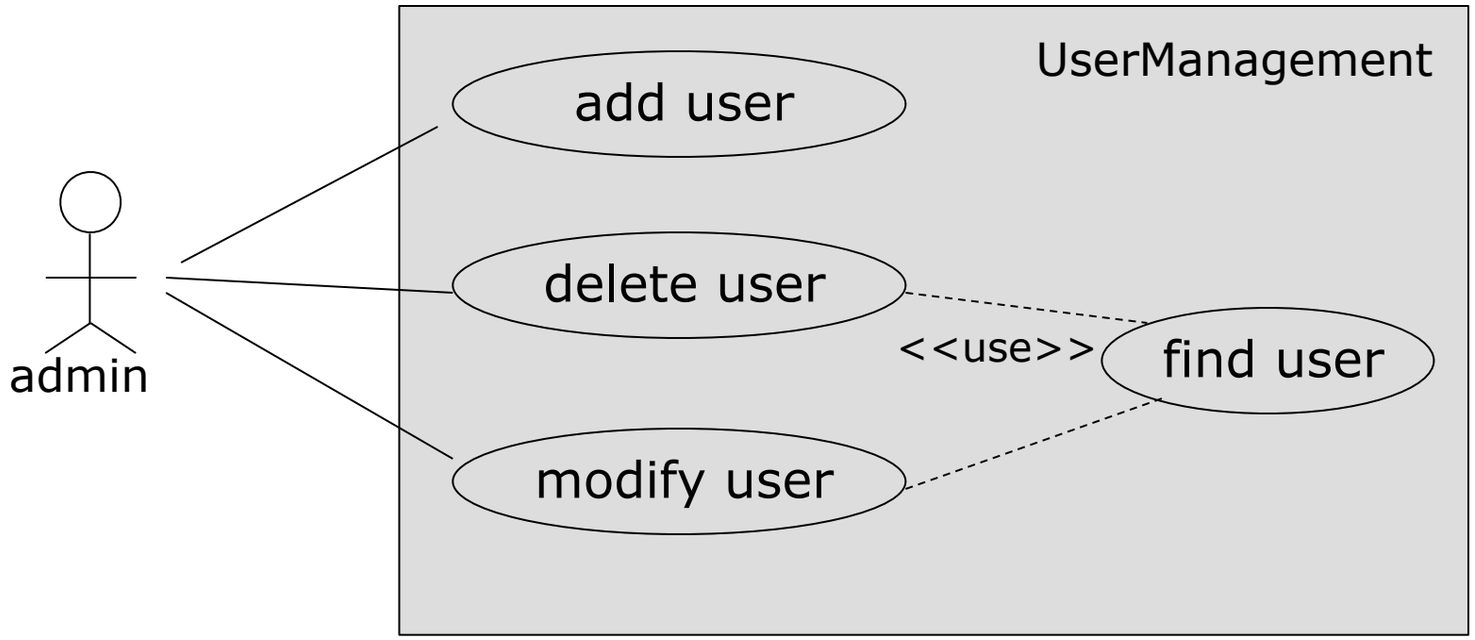
Post-install review
Support docs
Active support

Requirement

- Requirements are capabilities and conditions to which the system - and more broadly, the project - must conform
- Requirement analysis is about describing problems

Use-case diagram

- The first step in requirement analysis is to determine use-cases of the system
- Use-case diagrams
  - allow to **represent the functionalities of the system in the users view**
  - allow to delimit the boundary of the system



User-centred design

- The development of a system should always be centred around the needs of users
  - Understand who are the users
  - Understand the tasks performed by the users
  - Make sure that users are involved in the decision-making process
  - Design the interface well following the needs of the user
  - Users will need to evaluate prototypes and return their comments

Cash register at the supermarket

Interest of user-centred design

- Meets the actual requirements
- Reduce costs related to changes or maintenance
- Allow to better define the properties in the development
- Reduce learning time
- Reduce training and supporting costs
- Allow efficient use
- Making the system more attractive and better suited to its market

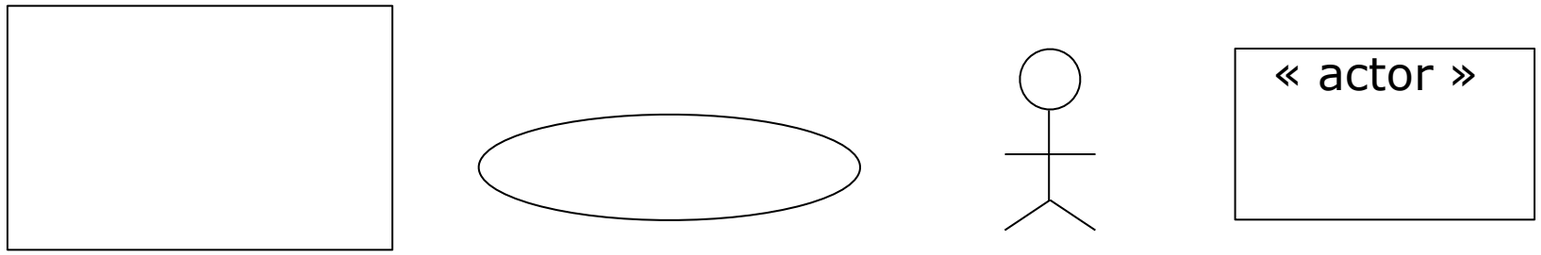
Determining users' characteristics

- Good questions
  - What are their goals?
  - How will they use the software?
  - What is their level of computer literacy?
  - What are their psychological characteristics?
  - What are their habits?

Use-case diagrams

- A use-case diagram consists of three parts
  - The system
  - The use-case
  - The actor

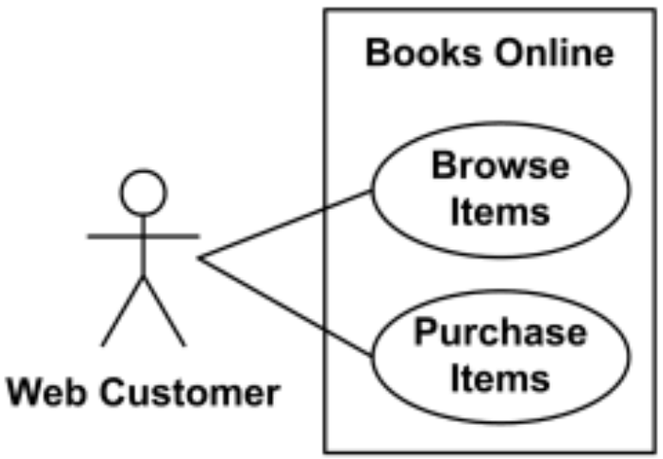
- Graphical representation



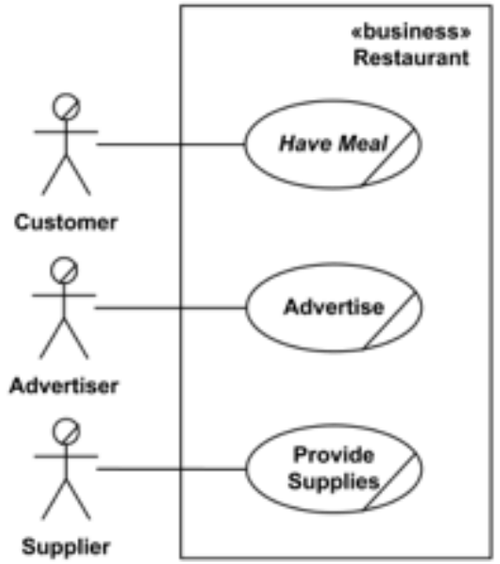
System Use-case Actor Actor

System

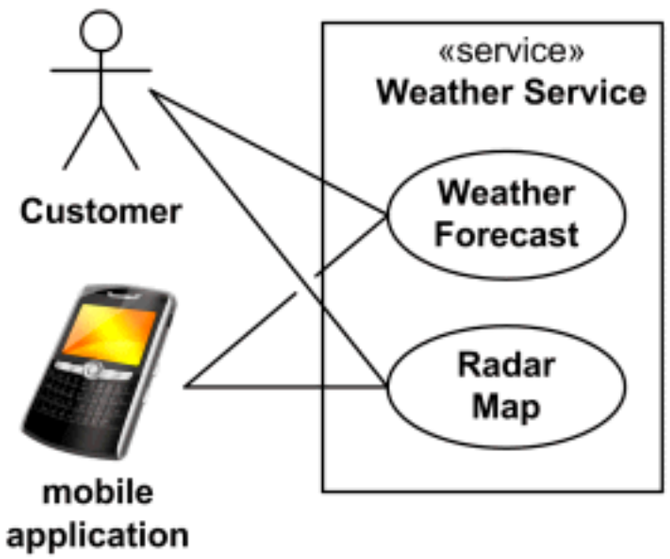
- The system can be any system, not only the software system
- It defines the boundary of the system in a clear and precise manner
  - Not too ambitious
  - Only determine basic functionalities
  - Build a well defined architecture
  - Additional functionality can be added during development



"Books Online" system



"Restaurant" system



"Weather Service" system

Use-case

- A use-case is a typical interaction or a typical sequence of interactions between the system and its environment
- The objective of a use-case is to model the system
  - according to the perspective of user interacting with the system
  - to accomplish their objectives
- A use-case may can be either large or small
- Example: developing a tool for text processing
  - Some possible use-cases
    - Create a new document
    - Modify an existing documents
    - Delete a document
    - Input new text, …

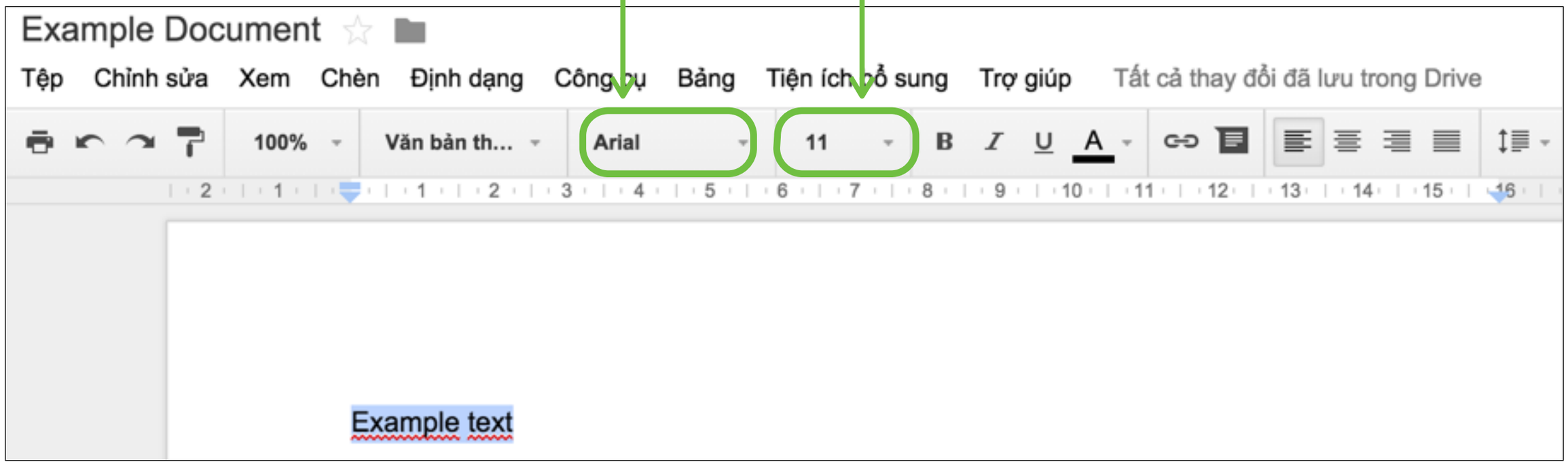
Text processing tool

Use-case

- A use-case needs to
  - always correspond to a high level objective
  - describe the interaction between the user and the system, not the operations that the system should perform
  - cover all the steps to follow in performing a given task
  - be written, to the possible extent, independently of the user interface
  - include only the interactions with the system

Use-case

- **Objectives** and **interactions**
  - **Objectives** of users: what users expect from the system
  - **Interactions** with the system: mechanisms to meet those objectives
  - Define the objectives then determine the interactions to achieve objectives
  - Example
    - Objective: define the document style
    - Interactions: choose the font, choose sizes, choose the page layout, ...



Use-case

- Example: developing of an ATM system
- Some interactions in the following scenario
  - Insert the card
  - Enter the PIN code
  - Choose the amount to be withdrawn
  - Confirm the amount
  - Take the card
  - Take the money
  - Take the receipt

- Are all interactions use-cases?

Use-case

- Example (continue)
  - The answer is no
  - Since some interactions such as “confirm the amount” do not meet a goal of the user
  - The goal of the user in this case is to **withdraw money**: this is a use-case

Actors

- An **actor** is a **role** played by the user or an external entity during interaction with the system
- Who or what uses the system
- Actors communicate with the system by sending and receiving messages
- Example
  - Develop a system of cash register at the supermarket
    - Possible actors
      - Client
      - Cashier
      - Manager
      - Inventory manager

Actors

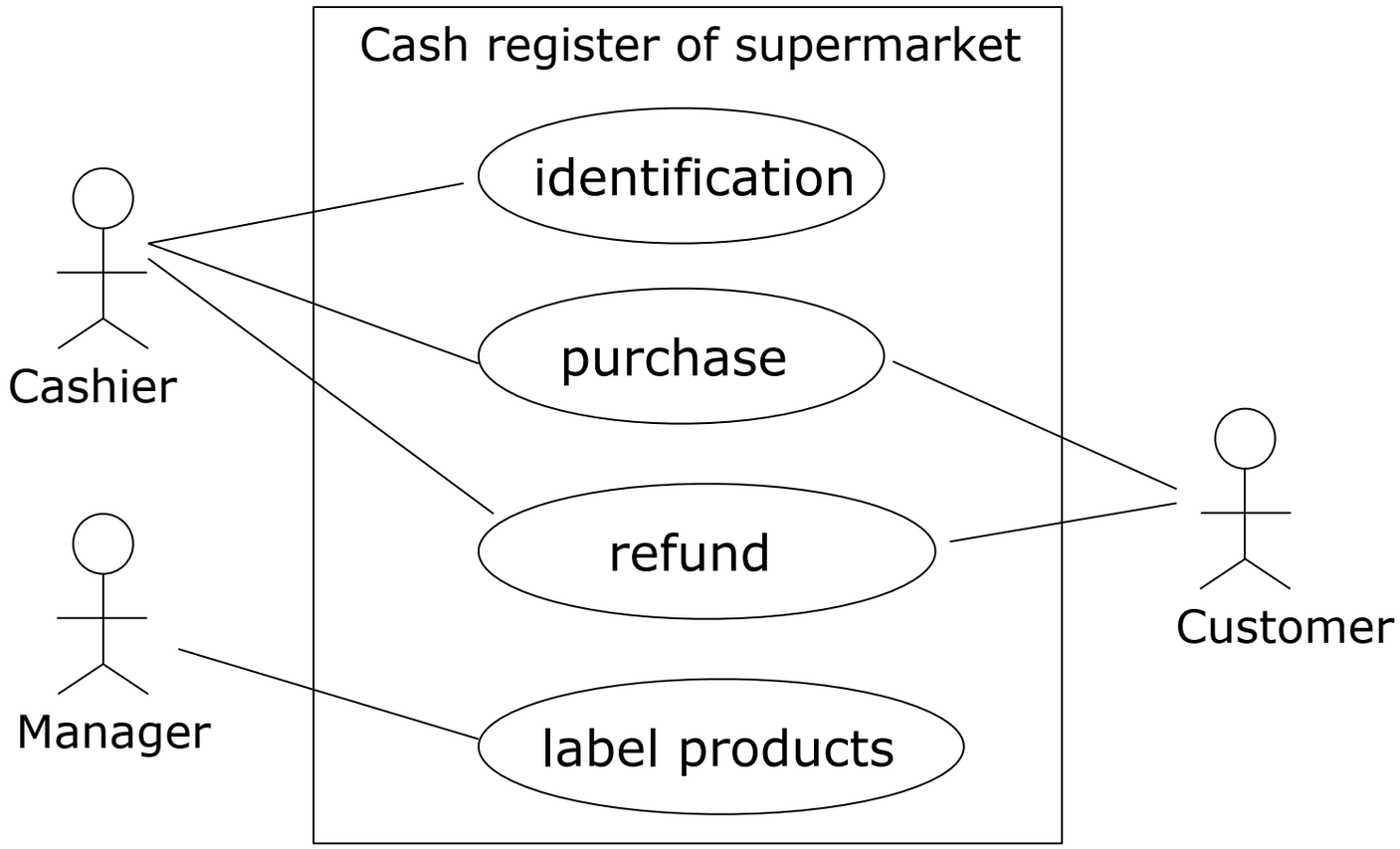
- Distinguishing two notions: **actor** and **user**
  - Multiple users may correspond to a single actor
    - Different cashiers play the same role in the system
  - A user may correspond to several actors
    - A user can simultaneously be a cashier and a manager in the system

cashier and manager

cashier and customer

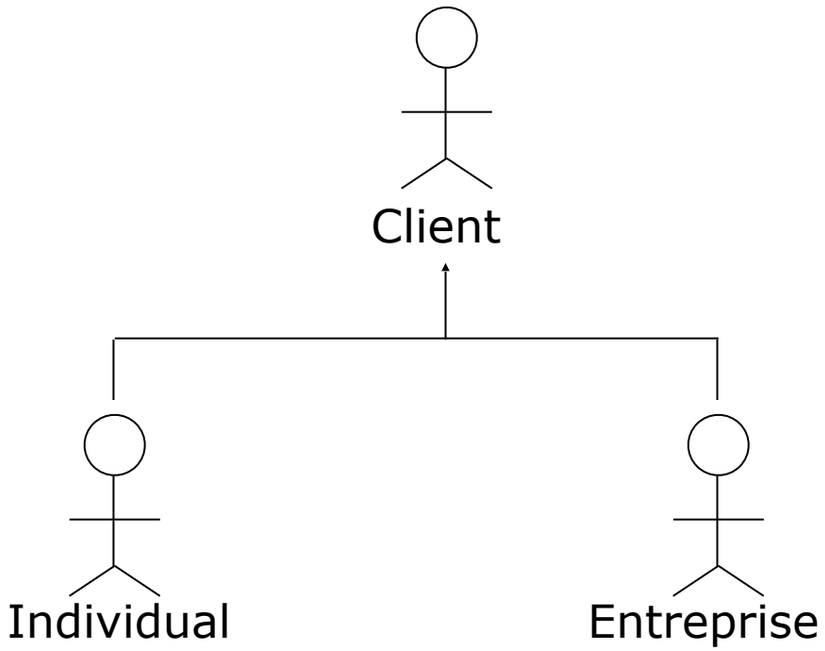
Actors

- Questions for identifying the system actors
  - Who will use the main features of the system?
  - Who will need the support of the system to perform its tasks?
  - Who should update, administer and maintain the system?
  - Does the system interacts with other systems?
  - Who or what has interests on the results of the system?



Relations between the actors

- Inheritance between actors



Use-case specification

- Typical specification of a use-case
  - **Use-case**: name of a use-case often begins with a verb
  - **Actors**: list of stakeholders concerning the use-case
  - **Objective**: objective of the use-case
  - **Description**: a brief description of treatment to achieve

- Example
  - **Use-case**: purchase of products
  - **Actors**: Client, Cashier
  - **Objective**: describe a purchase of products by the customer with cash payment
  - **Description**: The clients comes in the box with the selected products. The cashier encodes products, announces the total. The customer pays. The cashier registers the payments.

Use-case specification

- The use-case specification may add
  - the references concerning the specification of the requirement
  - the pre- and post-conditions of the use-case

- Example
  - **Use-case**: purchase of products
  - **Actors**: Client, Cashier
  - **Objective**: describe a purchase of products by the customer with cash payment
  - **References**: R1.2, R3.4
  - **Pre-conditions**: the cashier is identified and authorised
  - **Post-conditions**: the purchased is registered, the payment is made, the receipt is printed
  - **Description**: The clients comes in the box with the selected products. The cashier encodes products, announces the total. The customer pays. The cashier registers the payments.

Use-case specification

- A use-case can be specified by adding **scenarios**
- A scenario describes the specific actions of the actors in the system
- A scenario consists of principal interactions and exceptional interactions
- The actions can be divided into two flows
  - Flow of actions concerning the actors
  - Flow of actions concerning the systems
- Example
  - A scenario for “purchase products” use-case

Use-case specification

- Principal interactions of "purchase products" scenario

Actions of actor	Actions of system
• The customer comes to the cash desk with the products to buy	
• The cashier encodes the identifier of each product If a product has more than one item, the cashier inputs the number of items	• The cash desk displays the description and price of the product This number is displayed
• After having encoded all of the products, the cashier signals the end of the purchase	• The cash desk calculates and displays the total amount that the customer has to pay
• The cashier announces the total amount to the customer	
• The customer pays	
• The cashier input the amount of money paid by the customer	• The cash desk displays the balance

Use-case specification

- Principal interactions of “purchase products” scenario (continue)

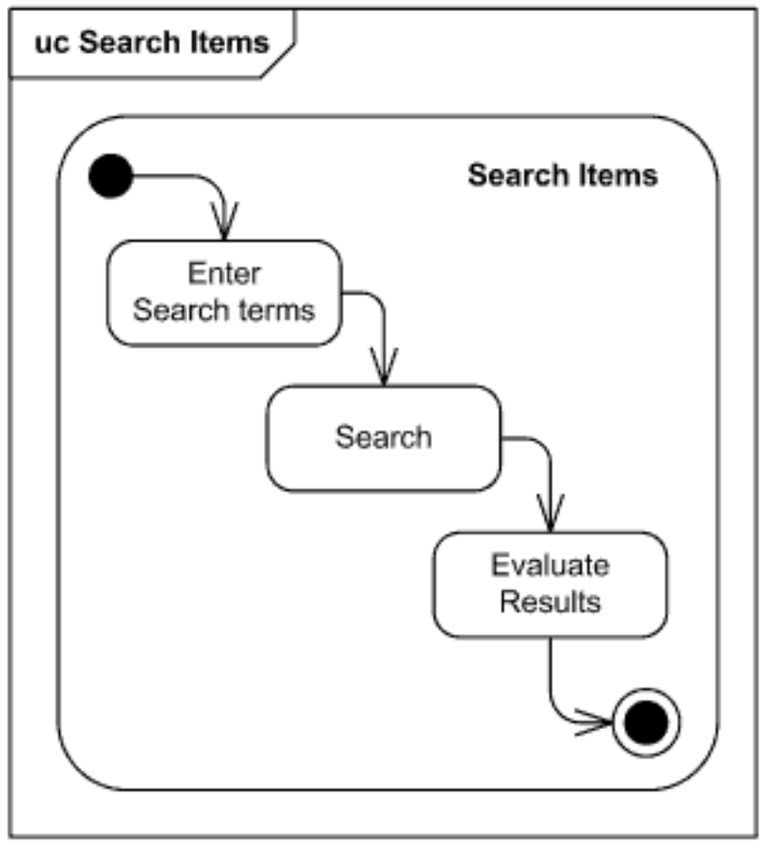
Actions of actor	Actions of system
	• The cash desk prints the receipt
• The cashier gives change to the customer and the receipt	• The cash desk saves the purchase
• The customer leaves the cash desk with the bought products	

- Exceptional interactions of “purchase products” scenario

Actions of actor	Actions of system
• The customer doesn’t have enough money. The cashier cancel the purchase	• The product identifier is not correct, the system displays the error

Use-case specification

- Remarks
  - The use-case's specification format is only a proposal. Therefore, it is not strict
  - The interactions are described in more detail for important use-cases
  - Use-case's interaction can also be described using activity diagram, state diagram or interaction diagram



Use-case's interactions described in activity diagram

Use-cases identification techniques

- **Software Developer write requirements specification themselves**
  - Lack of human reactions (future users of the system)

- **Interview**

User interview

User interview

Use-cases identification techniques

- Workshop *(Organise meetings)*
  - Meeting of all the concerned people of the system to be developed
    - Customers, Users, Software developers
    - Everyone gives their ideas
      - List all the possible actors, use-cases
      - Analyse and describe briefly each use-case
    - Model the use-cases and actors

Workshop

- Remarks
  - Don’t try to search for all the use-cases
    - Other use-cases can appear in the development process

Relationships between use-cases

- Two types of relationship between use-cases
  - Extension
  - Inclusion

- "extension" relationship
  - Used to specify the **optional interactions**
    - These are **exceptional cases**
  - The case where a use-case is similar to another but it includes **additional actions**
  - The extending use-case must list all the actions in the main use-case and also the supplementary actions

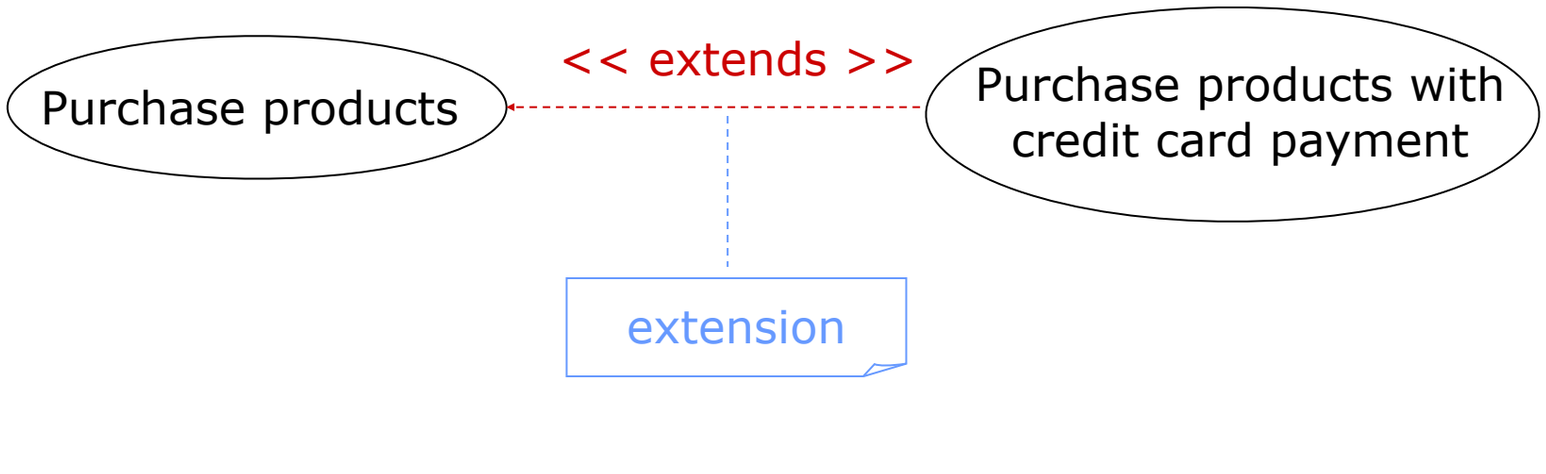
Relationships between use-cases

- “extension” relationship
  - Example: “purchase product with payment by credit card” use-case
    - **Use-case**: purchase products
    - **Actors**: Customer, Cashier
    - **Objective**: describe a purchase of products by the customer with payment by credit card
    - **Description**: The customer comes to checkout with selected products. The cashier encodes products, announces the total amount. The customer gives his credit card. The cashier inserts the credit card into the system. The customer types the PIN code. The system verifies the card and then deducts the total of the card.

  - This use-case is a variation of the “purchase products” use-case but adds actions relating to the use of credit card.

Relationships between use-cases

- "extension" relationship
  - "Purchase products with credit card payment" use-case is an extension of the "Purchase products" use-case
  - Notation



  - **Remarks**: If a use-case is associated with an actor, all extensions are also associated with this actor. This is expressed implicitly in the use-case diagrams.

Relationships between use-cases

- "inclusion" relationship
  - describes a series of joint actions in several cases of different usages
  - if several use-cases share **the same sequence of actions** and this common part is intended to meet a clearly defined goal then the part is described in a separate use-case
  - helps to avoid repeating the same details in different use-cases

Relationships between use-cases

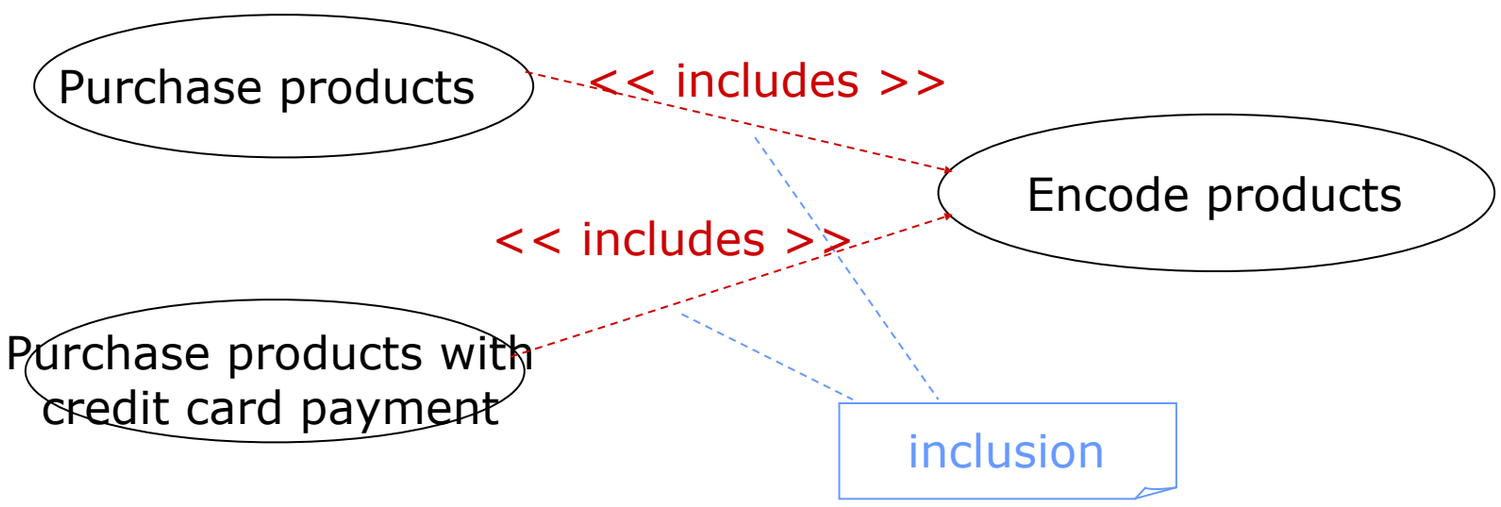
- Example of "inclusion" relationship
  - Suppose we have two use-cases "purchase product" and "purchase products with credit card payment"
    - Both use-cases have the same sequence of actions of encoding products that can be described by the "encode products" use-case

Actions of actor	Actions of system
• The customer comes to the cash desk with the products to buy	
• The cashier encodes the identifier of each product If a product has more than one item, the cashier inputs the number of items	• The cash desk displays the description and price of the product This number is displayed
• After having encoded all of the products, the cashier signals the end of the purchase	• The cash desk calculates and displays the total amount that the customer has to pay
• The cashier announces the total amount to the customer	

actions of encoding products

Relationships between use-cases

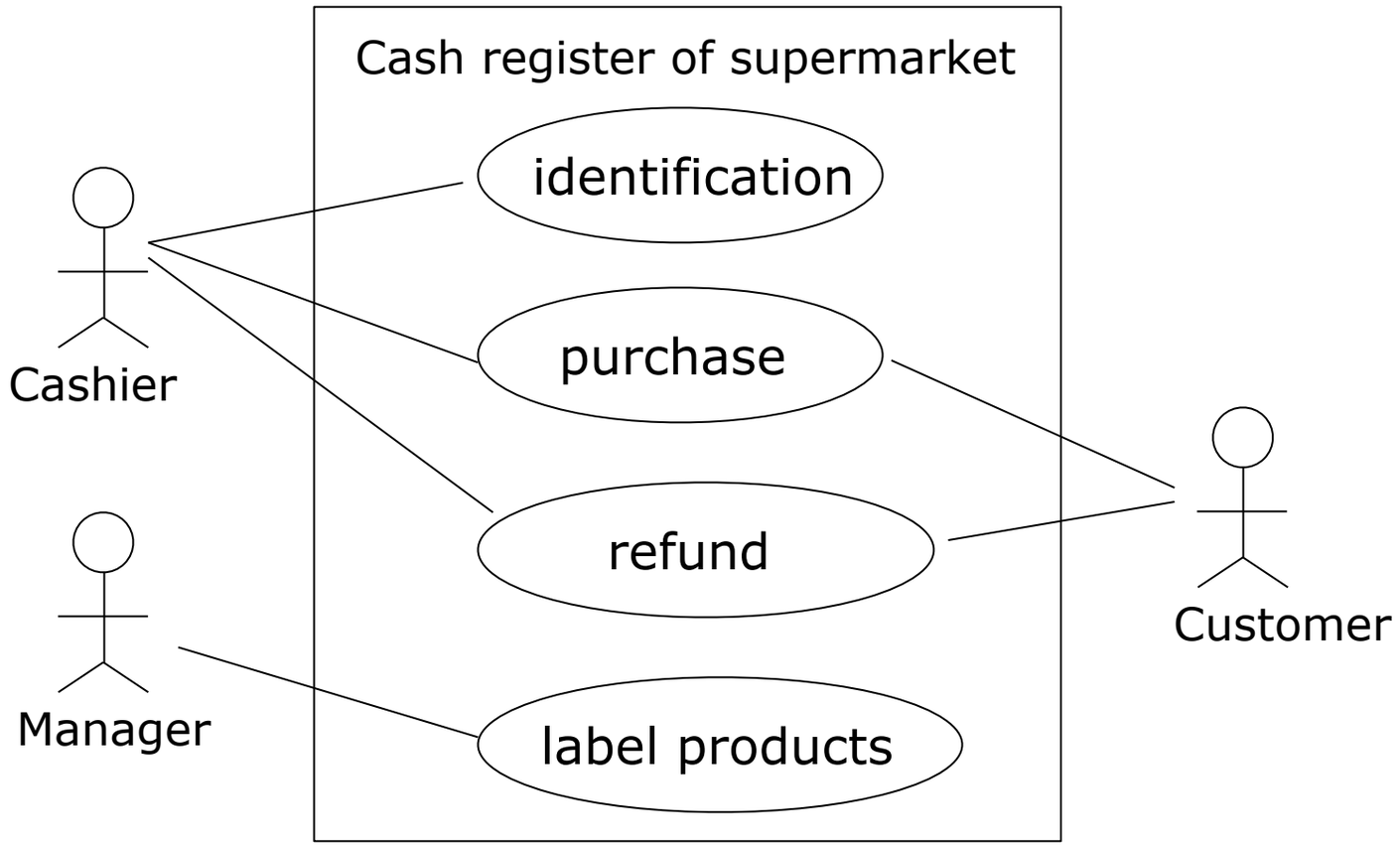
- "inclusion" relationship
  - Example (continue)
    - "encode products" use-case
      - **Use-case**: encode products
      - **Actor**: Customer, Cashier
      - **Objective**: describe the encoding of the products bought by a customer
      - **Description**: The customer comes to checkout with the selected products. The cashier encodes products, announces the total amount to the customer.
  - Notation



Building use-case diagrams

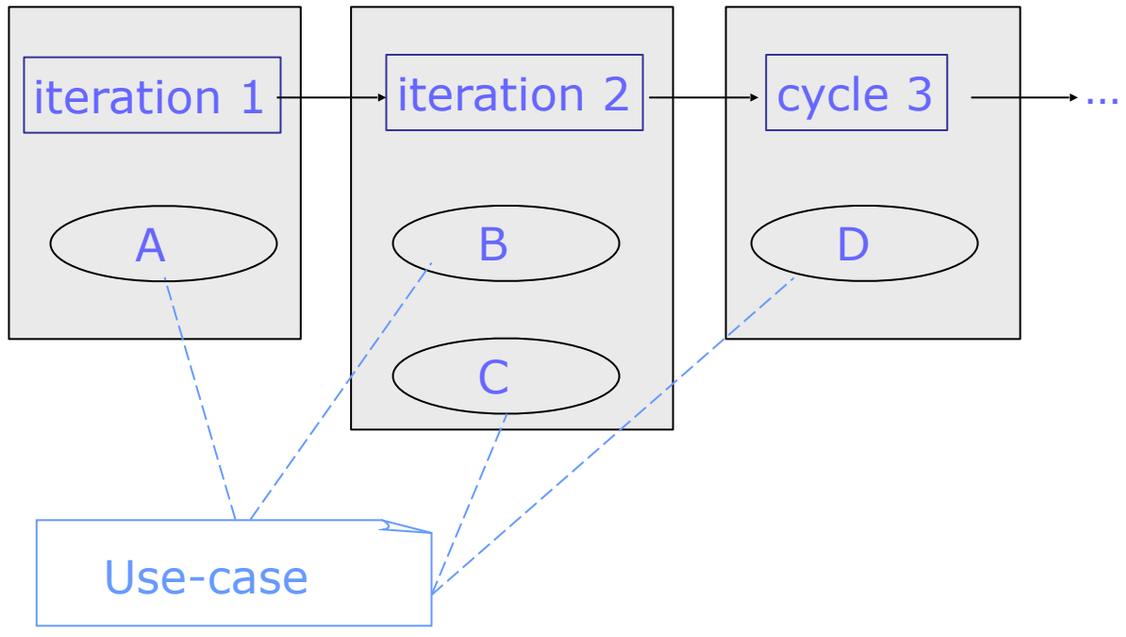
- A use-case diagram describes the relationships between the use-cases and actors of the system

- The steps to build a use-case diagram
  - Define the limits of the system
  - Identify the actors
  - Identify the use-cases
  - Define the relationships between use-cases
  - Verify the diagrams



Classification of use-cases

- Assign the use-cases to iterations of development process



- How to assign use-cases to iterations of development process?
  - Use-cases should be implemented in the order of importance. For example:
    - Use-cases that may contain risks
    - Use-cases that build the architecture of the software
    - Use-cases that realise a large part of the system functionality
    - Use-cases that require new technology or significant research
    - Use-cases that have great interests by the customer