

# 今天的内容

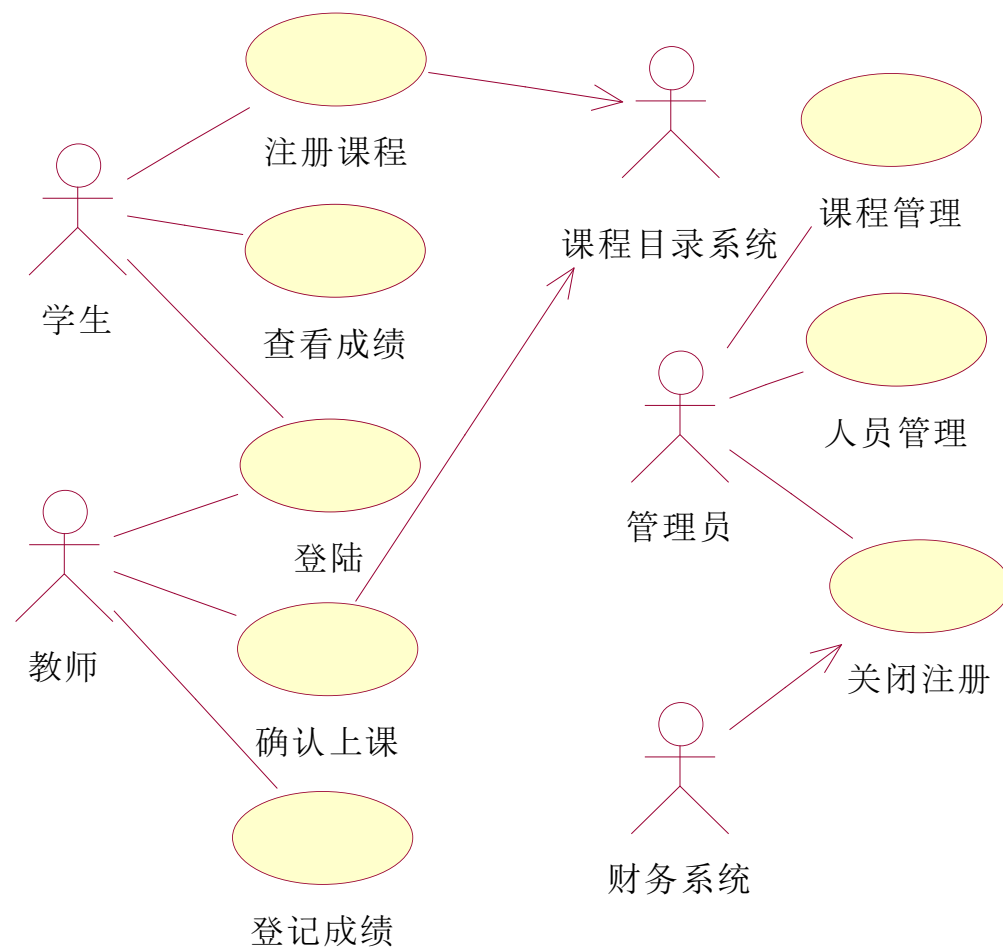
- 结合一个案例，讲：基于OO的需求分析的步骤和制品
  - 我的课程，案例，一直用这个：选课系统
- UML 中的用例图和活动图
  - Usecase diagram & activity diagram
  - Exercises by the students
- 时序图和协作图
  - Sequence diagram & collaboration

# Step-by-step process of OO based system requirement

## 基于面向对象的需求分析的步骤

### 步骤1 写问题陈述

- Problem statement (PS)
- vs personal statement
- 写项目的简单的背景和意义，主要的功能，本系统主要的外部环境（与该系统交互的其它软硬件系统和用户），主要的非功能需求
- 主要是用文字来描述
- 概述性的



# OO based system requirement

## 步骤2 用例析取、画出用例图

- 根据PS，将整个系统模块化，以便后续析取类
- 画出本系统的用例图
- 要图文并茂
- 所有的图都要有名称和编号

# OO based system requirement

## 步骤3 写用例规约 (usecase specification)

- 针对用例图中的每个用例，写其流程
- 用活动图配合文字
- 每个用例都要有
- 在你们的项目作业中，写3-5个（未来练习目标）

# OO based system requirement

## 步骤4 写补充规约 (supplementary Doc)

- 写面向全局性的非功能需求，例如安全性、可用性、可靠性

。 。 。 。 。

## 步骤5 写术语表 (Glossary)

- 解释本项目领域知识
- 把所有需要存储的“**名词**”纳入术语表
  - ✓ 以便后续构建数据类

# 思考

- 绘制用例图应该注意什么问题？
  - 用例的粒度
  - 客户没有讲出的角色、功能、其它需要交互的软硬件
  - 用例命名规范
- 写用例规约需要注意的事情？
  - 备选事件流要尽量考虑完整
  - 输入和输出特征、参数类型等，要写完整
  - 活动图中同步条的使用要考虑合理性

# Requirements Overview

## 软件需求分析

### Objectives

- Describe the basic Requirements concepts and how they affect Analysis and Design
- Demonstrate how to read and interpret artifacts of Requirements that are used starting point for Analysis and Design



# Overall artifacts in whole A&D based on OO (1)

1st iteration



1. **Project Description**
2. **Requirement: UseCase, Glossary, UseCase Specification, Supplement Specification.**
3. **Analysis and Design: Architecture Document, Design Pattern, Key Abstraction, UseCase Realization**

# Overall artifacts in whole A&D based on OO (2)



2nd iteration

4. **UseCase Analysis: Use-Case Realization**  
Interaction; Diagram for at least one of the usecase flows of events; VOPC class diagram, containing the analysis classes, their stereotypes, responsibilities, attributes, and relationships; Analysis class to analysis mechanism map.

# Overall artifacts in whole A&D based on OO (3)

2nd iteration



**5. Designing Element: Design classes, subsystems, their interfaces and their relationships with other design elements; Mapping from the analysis classes to the design elements; The location of the design elements (e.g. subsystems and their design classes) in the architecture (i.e., the package/layer that contains the design element). For each subsystem, an interface realization class diagram; Table mapping analysis classes to design elements; Table listing design elements and their “owning” package.**

# Overall artifacts in whole A&D based on OO (4)



2nd iteration

6. **Running Time Architecture: Class diagram showing the Processes, Mapping of classes and subsystems to processes; Process relationships, Design element relationships to support process relationships.**
7. **Distribution: Deployment diagram depicting, Nodes, Connections and What processes run on what nodes,.**

# Overall artifacts in whole A&D based on OO (5)

3rd iteration

8. UseCase Design: Class diagram (VOPC) that includes the design elements that must collaborate to perform the use case, and their relationships.
9. Subsystem Design: Interface realizations, Interaction diagram for each interface operation; Class diagram containing the subsystem design elements that realize the interface responsibilities and their relationships; Class diagram that shows the subsystem and any dependencies on external package(s) and/or subsystem(s) (subsystem dependencies class diagram).

# Overall artifacts in whole A&D based on OO (6)



**3rd iteration**

**10. Class Design, Produce the following: An updated VOPC, including the relationship refinements (generalization, dependency, association).**

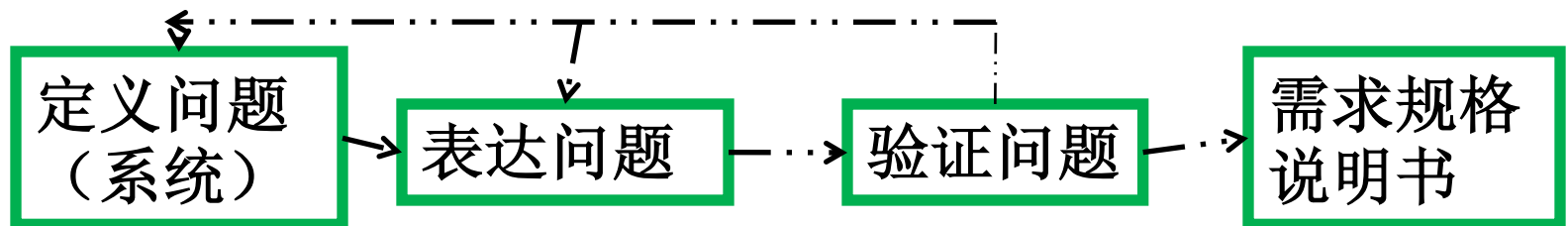
**11. Data Base Design**

# 需求分析的步骤和制品与编程的方法论有关

- 如果采用**基于结构化的编程语言**，例如C，那么需求分析的步骤和制品应按照“结构化需求分析的要求”来写，具体内容见“基于结构化的需求分析”章节
- 如果采用**基于面向对象的编程语言**，例如C++/JAVA，那么需求分析的步骤和制品应按照“面向对象需求分析的要求”来写
  - 包括：**问题陈述、用例模型（内含用例图和用例规约）、术语表、补充规约**

# 需求分析的一般性过程

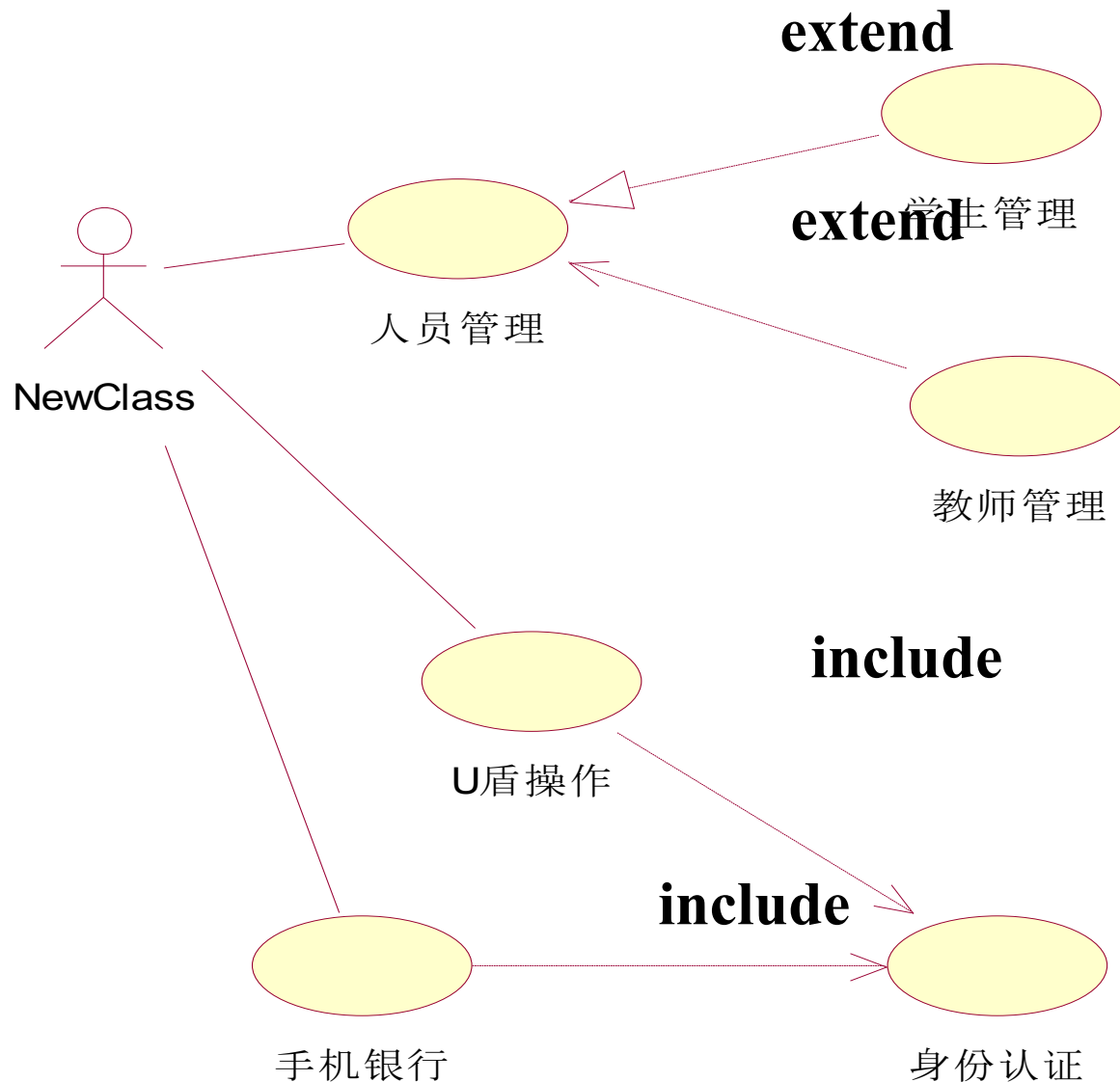
- 无论采用基于结构化或者基于面向对象，需求分析的任务就是回答“系统要做什么”、“what to do”
- 需求分析的任务包括表述清晰功能需求、非功能需求，又是也需要表述出“业务需求”
- 需求分析的一般性过程为：





# Usecase diagram 中include vs extend 的建模

- **IF** 父代用例有几个子用例，则**extend**
  - **Such as**,人员管理包含{教师管理+学生管理}，用**extend**
- **IF** 几个父用例有一个共有的流程，用**include**
  - **Such as**, 手机银行和网银，都需要 “身份认证”



注意：同一簇模型中采用的图形符号要相同

# 需求规约 (1) (这个是基于结构化的)

I. 引言	A.系统参考文献
	B.整体描述
	C.软件项目约束
II. 信息描述	A.信息内容表示
	B.信息流表示: i 数据流 ii 控制流
III. 功能描述	A.功能划分
	B.功能描述: i 处理说明 ii 限制/局限 iii 性能需求 iv 设计约束 v 支撑图 C.控制描述 i 控制规约 ii 设计约束
IV. 行为描述	A.系统状态
	B.事件和响应
V. 检验标准	A.性能范围
	B.测试种类
	C.期望的软件响应
	D.特殊的考虑
VI. 参考书目	
VII. 附录	

# 需求规约（II）

- 1.引言：**陈述软件目标，在基于计算机的系统语境内进行描述。
- 2.信息描述：**给出软件必须解决问题的详细描述，记录信息内容和关系、流和结构。
- 3.功能描述：**描述解决问题所需的每个功能。其中包括，为每个功能说明一个处理过程；叙述设计约束；叙述性能特征；用一个或多个图形来形象地表示软件的整体结构和软件功能与其他系统元素间的相互影响。

# 需求规约 （III）

4. 行为描述：描述作为外部事件和内部产生的控制特征的软件操作。
5. 检验标准：描述检验系统成功的标志。即对系统进行什么样的测试，得到什么样的结果，就表示系统已经成功实现了。它是“确认测试”的基础。
6. 参考书目：包含了对所有和该软件相关的文档的引用，其中包括其他的软件工程文档、技术参考文献、厂商文献以及标准。
7. 附录：包含了规约的补充信息，表格数据、算法的详细描述、图表以及其他材料。

# 基于面向对象的软件需求分析

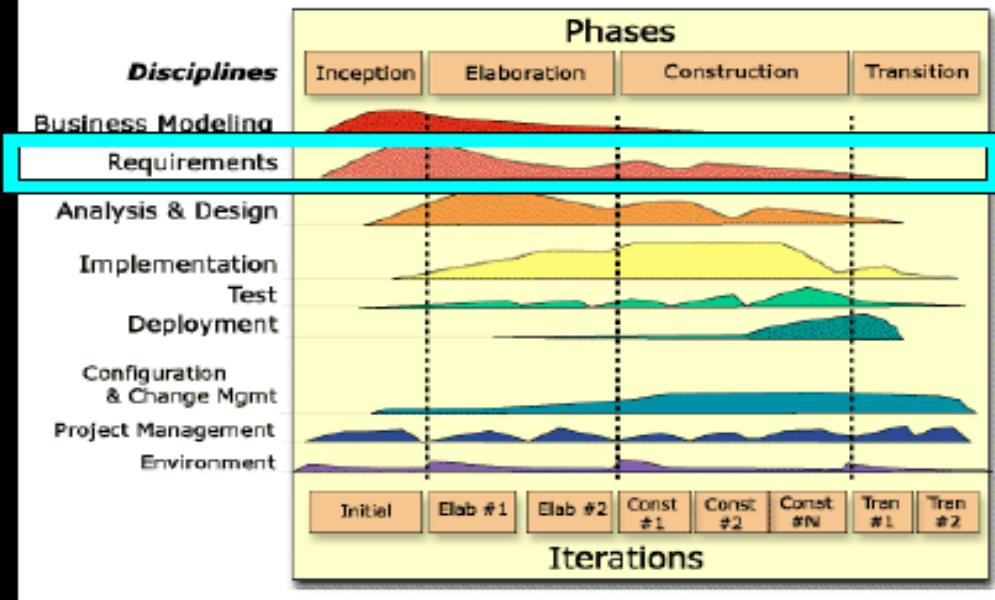
## 主要内容

- RUP中的需求流程
- 基于OO的需求分析的步骤和制品
  - ✓ 撰写问题陈述
  - ✓ 绘制并且撰写用例模型
  - ✓ 完成术语表
  - ✓ 补充规约
- 检查点
- 案例实践

# Requirements in Context

The purpose of Requirements is to:

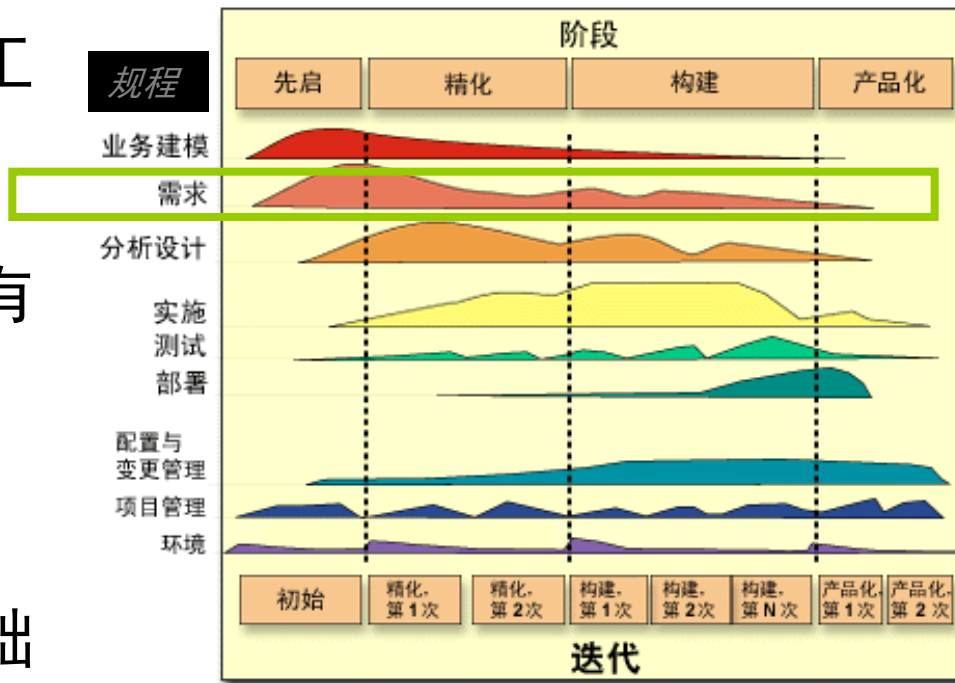
- Establish and maintain agreement with the customers and other stakeholders on what the system should do.
- Give system developers a better understanding of the requirements of the system.
- Delimit the system.
- Provide a basis for planning the technical contents of the iterations.
- Provide a basis for estimating cost and time to develop the system.
- Define a user interface of the system.



# RUP规程中的需求

## 需求的目的是

- 与客户和其他涉众在系统工作内容方面达成并保持一致
- 让开发人员对系统的需求有更好的理解
- 划分系统的边界
- 为迭代的技术内容提供基础
- 为估算开发系统所需成本和时间提供基础
- 定义系统的用户界面

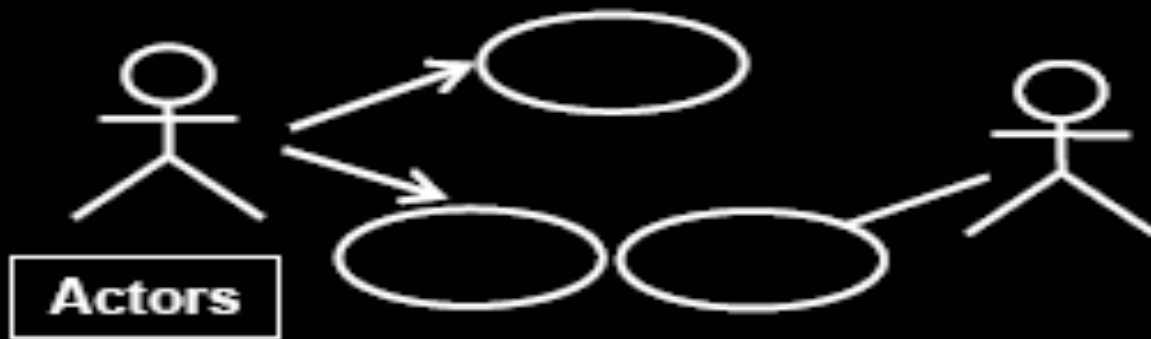




# Relevant Requirements Artifacts

需求分析的所有制品  
(没有包括问题陈述)

## Use-Case Model

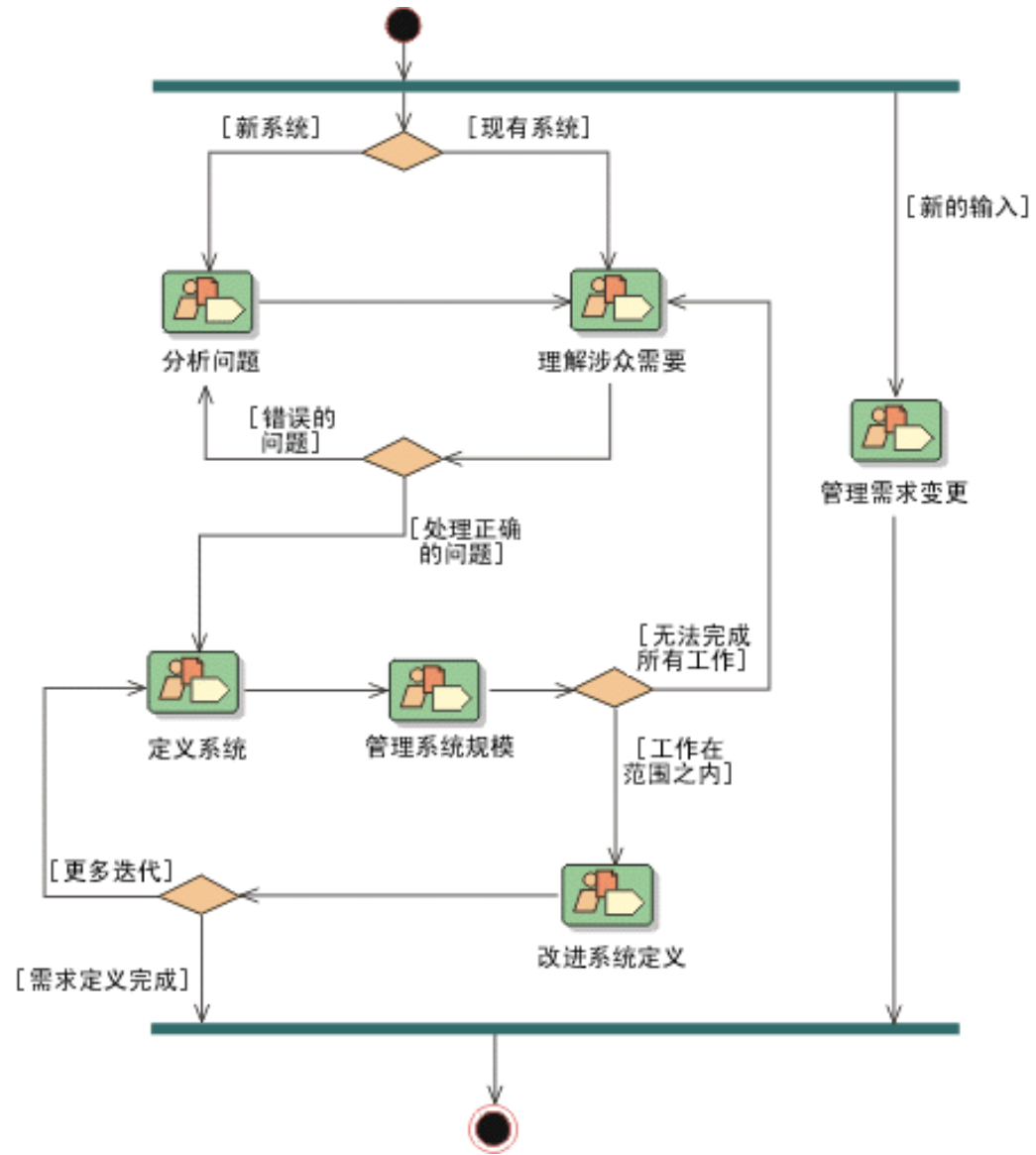


Glossary



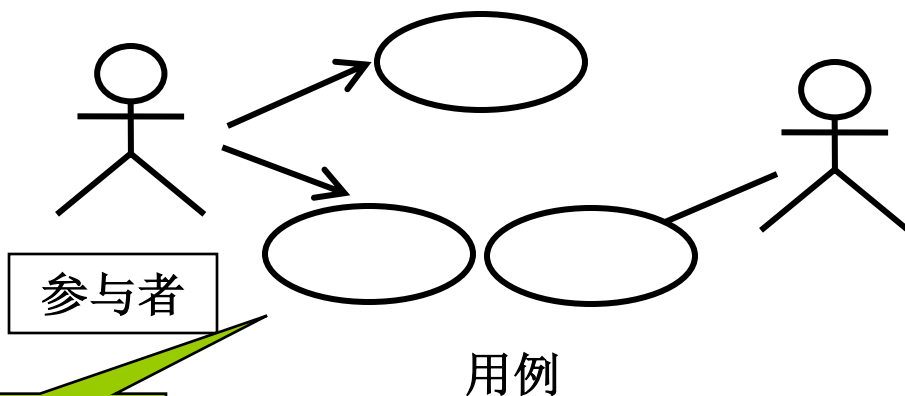
Supplementary  
Specification

# 需求工作流程



# 相关需求制品

## 用例模型



用例图描述  
参与者与用  
例间的关系

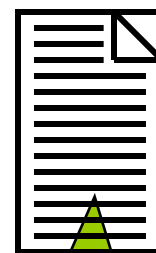
用例规约描述  
每一个用例的  
细节信息

用例规约



术语表

记录一些系  
统需求相关  
的术语



补充规约

记录一些全局性的功  
能需求、非功能性需  
求和设计约束等

# Case Study: Course Registration Problem Statement



课程注册需求文档

Review the problem statement provided in the  
Course Registration Requirements Document.

# 案例学习：课程注册系统

- 浏览课程注册系统的问题陈述文档



课程注册需求文档

# 第六章 软件需求

## 主要内容

RUP中的需求流程

用例模型

术语表

补充规约

检查点

案例实践

# What Is System Behavior?

- **System behavior is how a system acts and reacts.**
  - ◆ **It is the outwardly visible and testable activity of a system.**
- **System behavior is captured in use cases.**
  - ◆ **Use cases describe the system, its environment, and the relationship between the system and its environment.**

# Major Concepts in Use-Case Modeling

- ♦ An actor represents anything that interacts with the system.



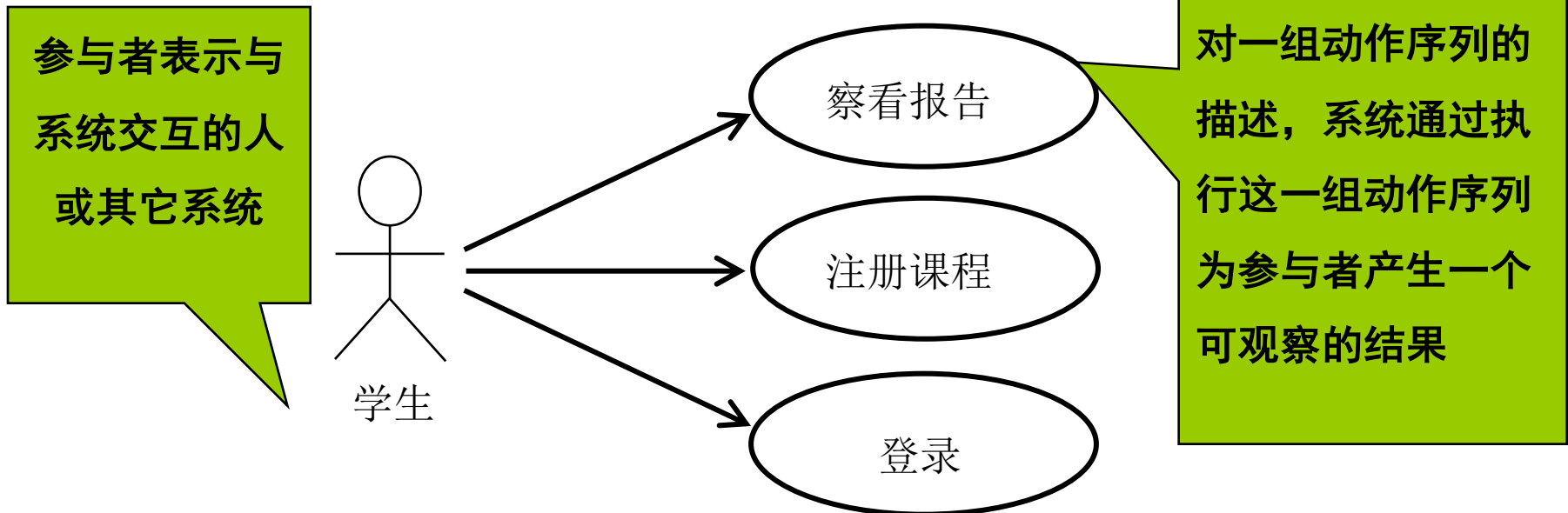
- ♦ A use case is a sequence of actions a system performs that yields an observable result of value to a particular actor.

UseCase



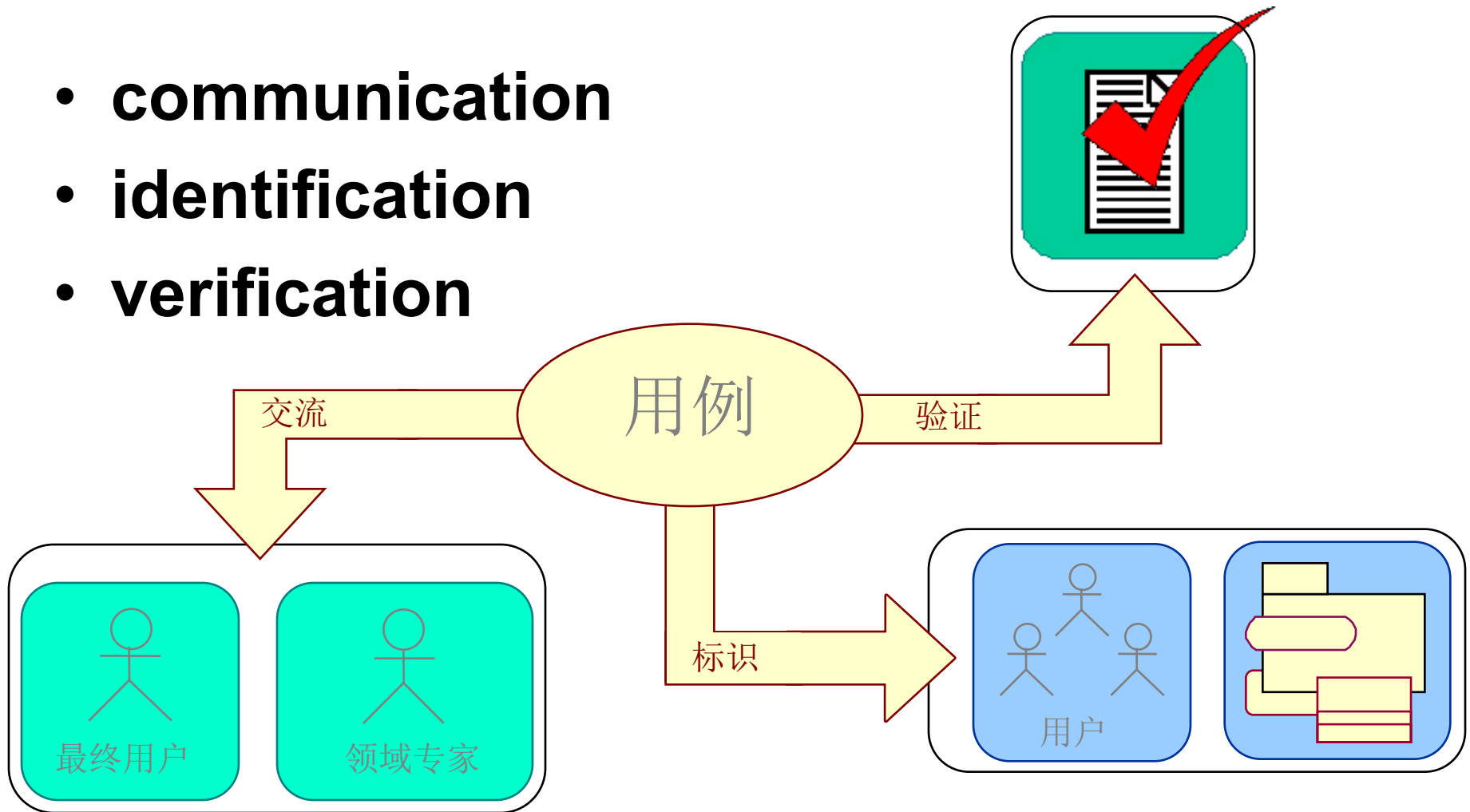
# What Is a Use-Case Model?

- A model that describes a system's functional requirements in terms of use cases
- A model of the system's intended functionality (use cases) and its environment (actors)



# What Are the Benefits of a Use-Case Model?

- **communication**
- **identification**
- **verification**



# Construct usecase model

- **Process of describing system requirements and function by used is called usecase modeling**
- **Step-by-step process**
  - **Ascertain roles**
  - **Ascertain use cases**
  - **Glossary**
  - **Checkup usecase model**

# Ascertain the roles

- **Finding out the roles by questionnaire**
  - **Who use the main function of the system?**
  - **Who or which system receive?**
  - **Who provide the data?**
  - **Which system are related to?**
  - **Who manage or maintain the system?**

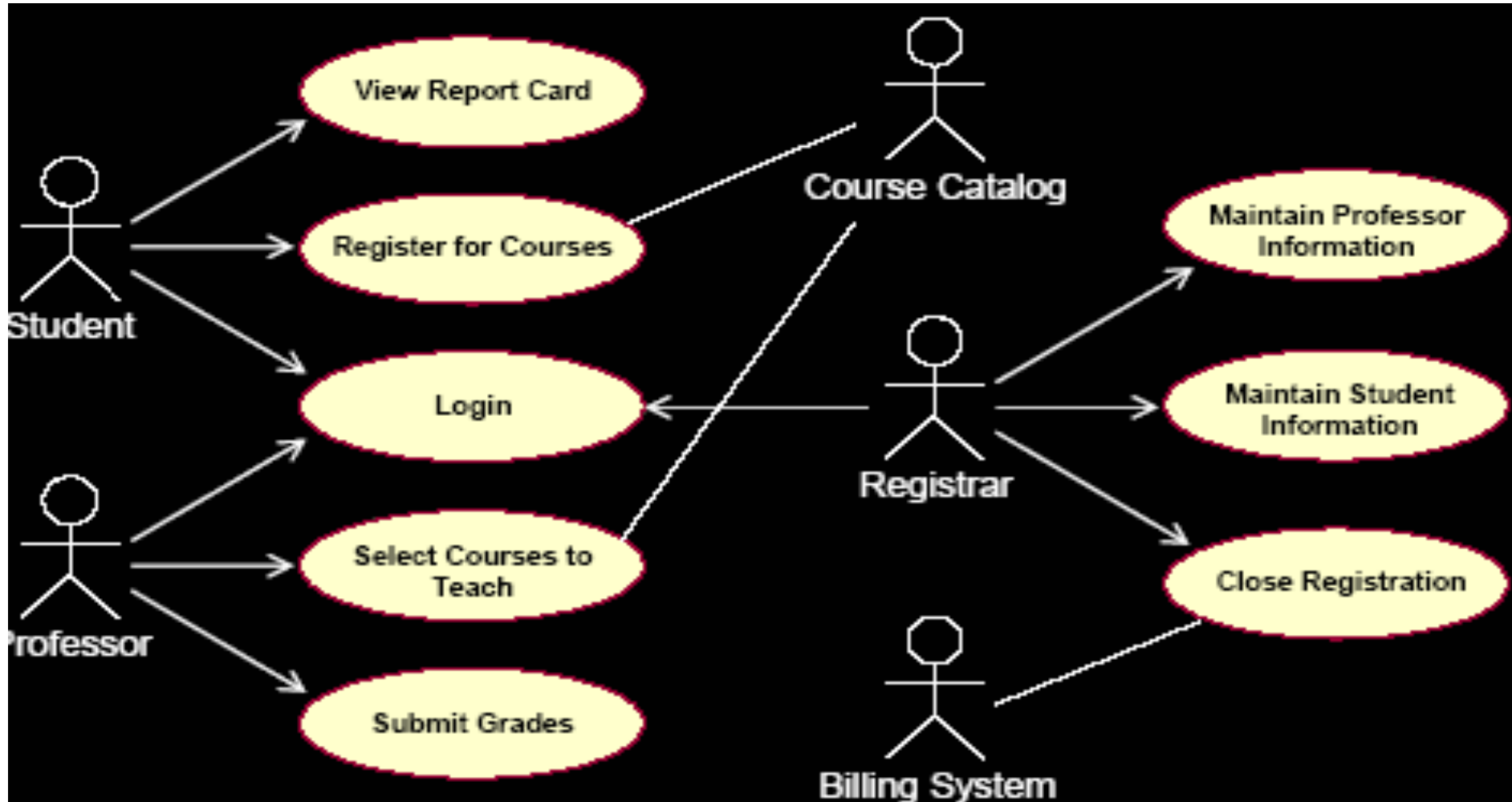
# Case study: roles in usecase

- **student——register course**
- **professor——select course to teach**
- **manager——maintain student and professor information**
- **Billing system——obtain some valuable information**
- **Course register system——maintain the course information**

# Ascertain usecase

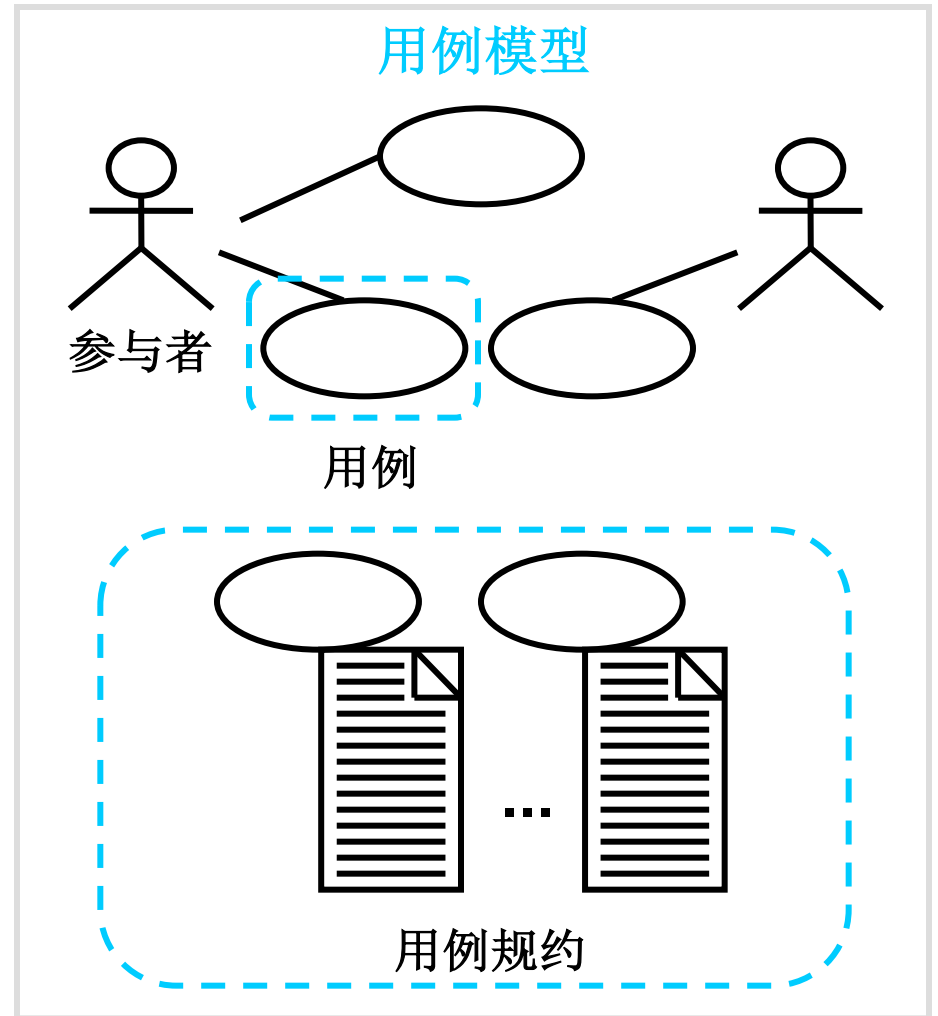
- **Finding out the usecase by questionnaire**
  - **Why the roles will use the system?**
  - **Will the roles will set up, modify, delete, brows, archive some data, and how to do those things ?**
  - **Do the roles inform of the system about outside information ?**
  - **Does the system give notice about the system to the roles?**

# How Would You Read This Diagram?



# Use-Case Specifications

- Name
- Brief description
- Flow of Events
- Relationships
- Activity diagrams
- Use-Case diagrams
- Special requirements
- Pre-conditions
- Post-conditions
- Other diagrams





# Use-Case Flow of Events

- Use-Case Flow

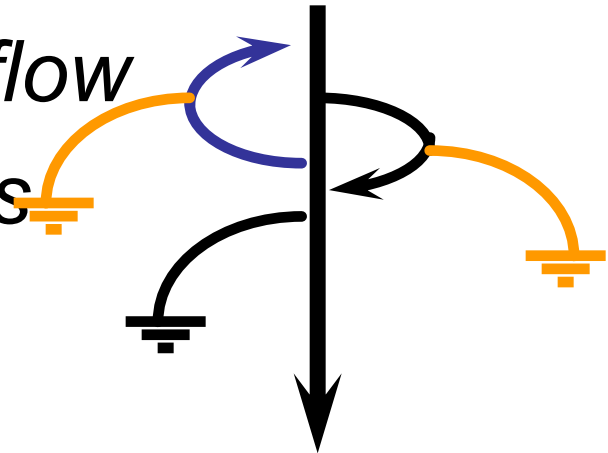
- *Has one normal, basic flow*

- *Several alternative flows*

- Regular variants

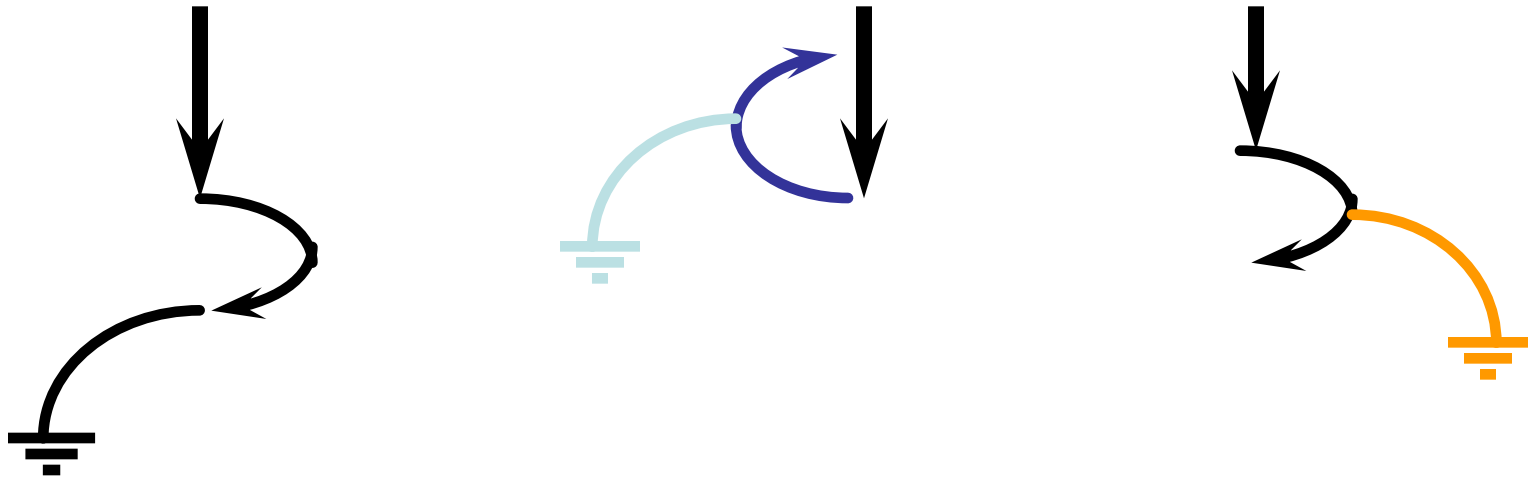
- Odd cases

- Exceptional flows for handling error situations



# What Is a Scenario?

- A scenario is an instance of a use case.



# What Is an Activity Diagram?

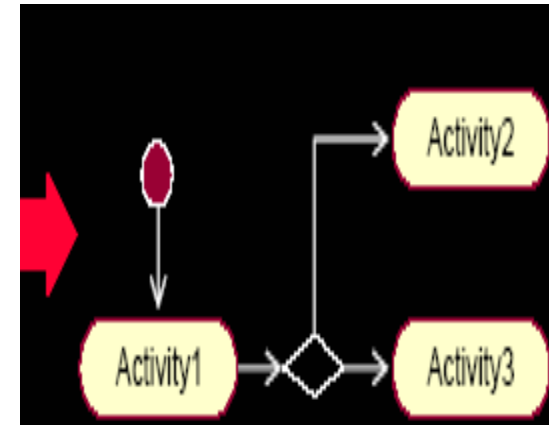
- An activity diagram in the Use-Case Model can be used to capture the activities in a use case.
- It is essentially a flow chart, showing flow of control from activity to activity.

## ***Flow of Events***

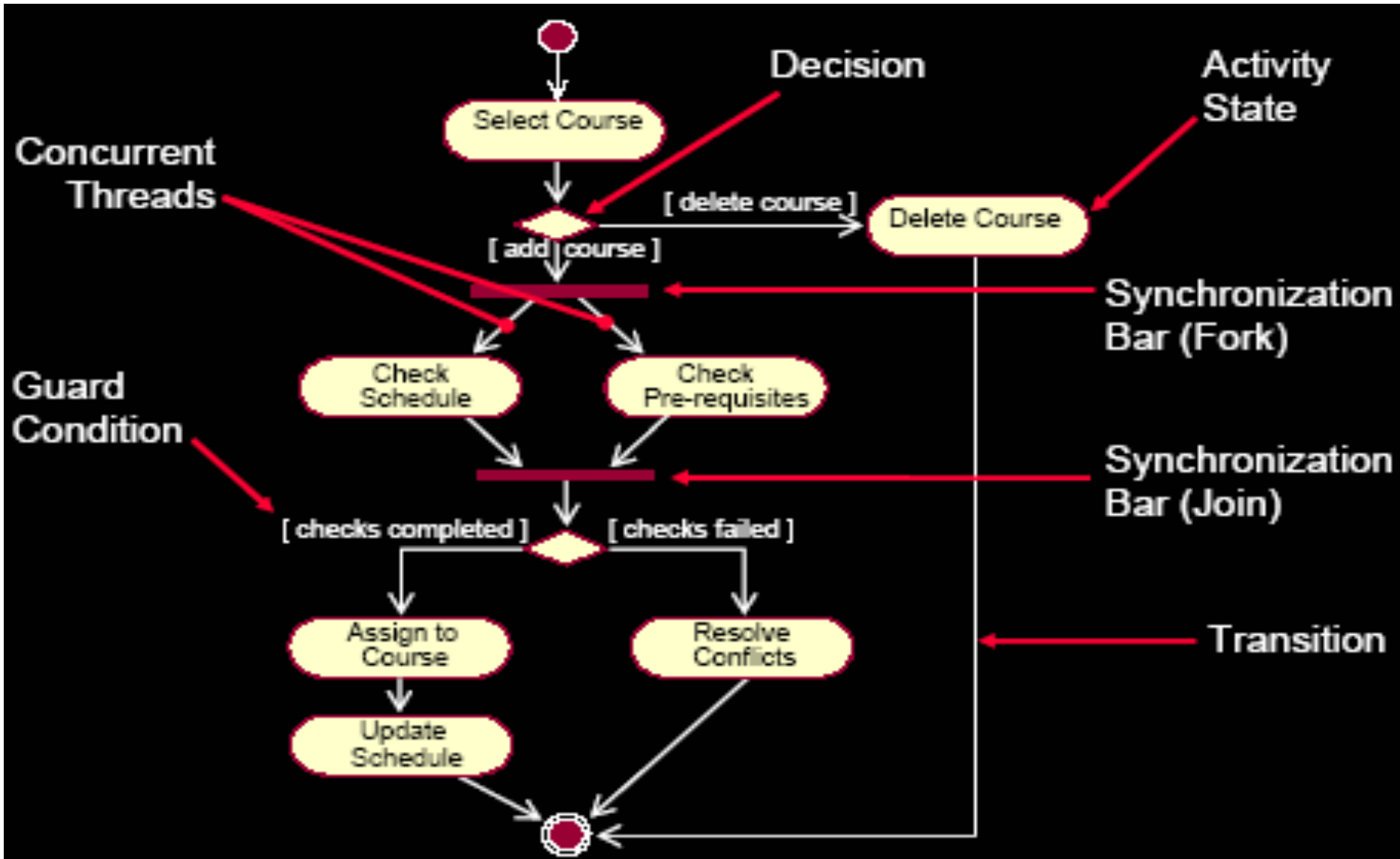
This use case starts when the Registrar requests that the system close registration.

1. The system checks to see if registration is in progress. If it is, then a message is displayed to the Registrar and the use case terminates. The Close Registration processing cannot be performed if registration is in progress.

2. For each course offering, the system checks if a professor has signed up to teach the course offering and at least three students have registered. If so, the system commits the course offering for each schedule that contains it.



# Example: Activity Diagram



# 登录系统的用例规约

## 简要说明:

学生在进行课程注册前必须先登录该系统；教授也需要先登录才能进行选择课程等操作；登记员也必须登录后才能进行有关信息的维护

## 事件流:

略

## 特殊需求:

无

## 前置条件:

学生/教授/登记员有正确的用户名和密码

## 后置条件:

成功登录后，学生/教授/登记员可分别进行相应权限内的操作；如果登录失败则不能进行任何操作

# 第六章 软件需求

## 主要内容

RUP中的需求流程

用例模型

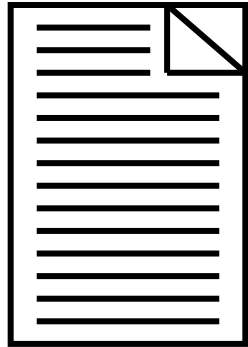
术语表

补充规约

检查点

案例实践

# Glossary



术语表



## 1. Introduction

This document is used to define terminology specific to the problem domain, explaining terms, which may be unfamiliar to the reader of the use-case descriptions or other project documents. Often, this document can be used as an informal *data dictionary*, capturing data definitions so that use-case descriptions and other project documents can focus on what the system must do with the information.

## 2. Definitions

The glossary contains the working definitions for the key concepts in the Course Registration System.

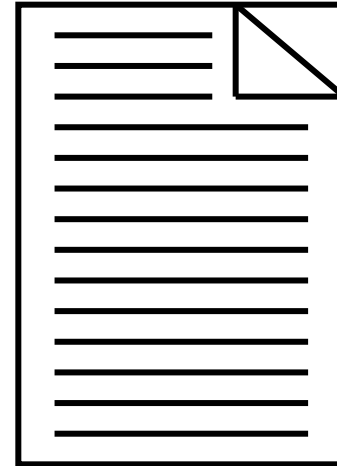
**2.1 Course:** A class offered by the university.

**2.2 Course Offering:** A specific delivery of the course for a specific semester – you could run the same course in parallel sessions in the semester. Includes the days of the week and times it is offered.

**2.3 Course Catalog:** The unabridged catalog of all courses offered by the university.

# Case Study: Glossary

- Review the Glossary  
provided in the Course  
Registration Requirements  
Document



Glossary



# 第六章 软件需求

## 主要内容

RUP中的需求流程

用例模型

术语表

补充规约

检查点

案例实践

# Supplementary Specification

- Functionality
- Usability
- Reliability
- Performance
- Supportability
- Design constraints

描述全局性的功能需求

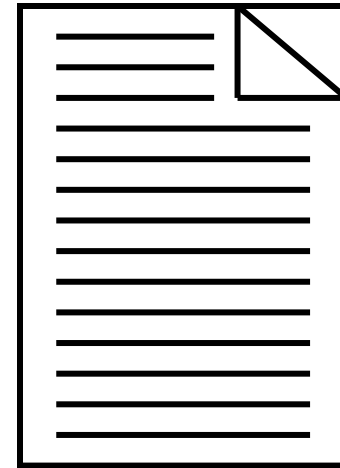
记录所有可用性相关的需求，如系统的使用者所接受的培训时

包括可用性，一些故障频率和严重标准如

包括响应时间、吞吐量、资源

包括编码标准、命名约定、类库、如何来对系统进行维护操作和相应的维护

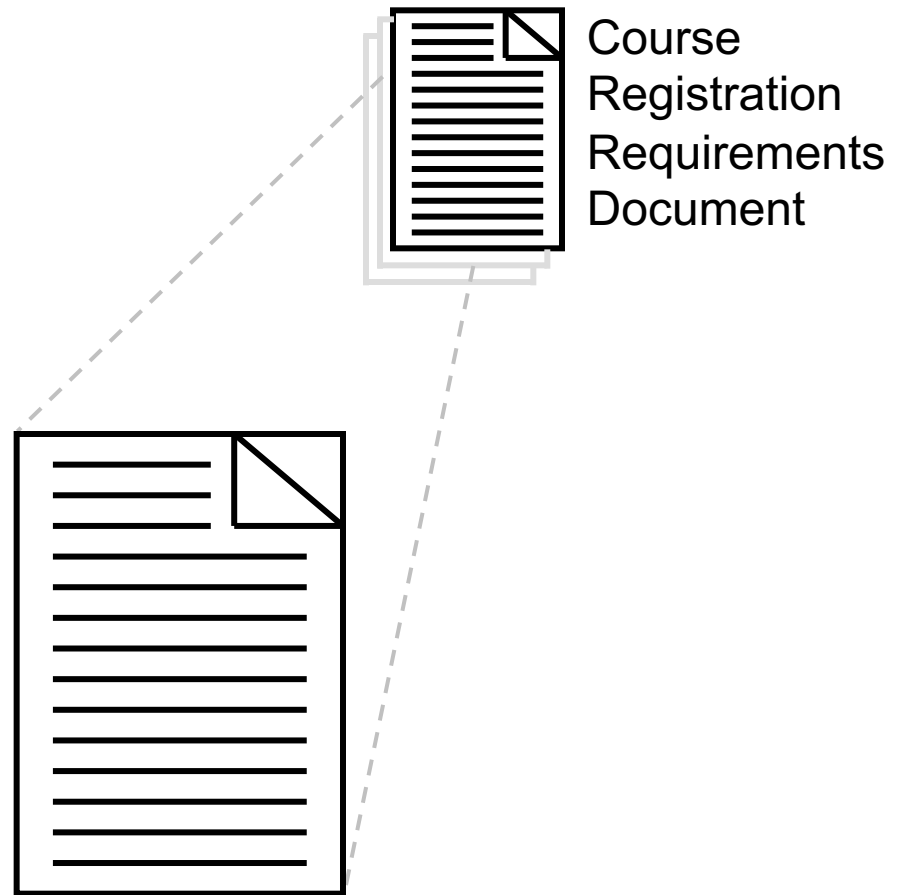
代表已经批准并必须遵循的设计决定，其中包括软件开发流程、开发工具、系统构架、编程语言、第三方构件类库、运行平台和数据库系统等



补充规约

# Example: Supplementary Specification

- Review the Supplementary Specification provided in the Course Registration Requirements Document.



# 第六章 软件需求

## 主要内容

RUP中的需求流程

用例模型

术语表

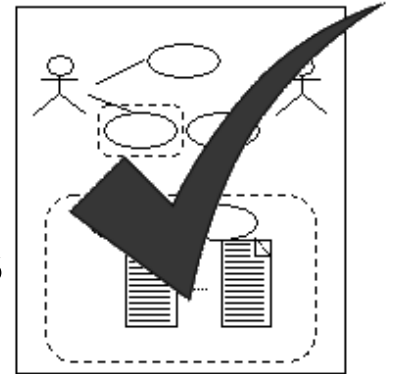
补充规约

检查点

案例实践

# Checkpoints: Requirements: Use-Case Model

- **Is the Use-Case Model understandable?**
- **By studying the Use-Case Model, can you form a clear idea of the system's functions and how they are related?**
- **Have all functional requirements been met?**
- **Does the Use-Case Model contain any superfluous behavior?**
- **Is the division of the model into usecase packages appropriate?**



# Checkpoints: Requirements: Actors

- **Have all the actors been identified?**
- **Is each actor involved with at least one use case?**
- **Is each actor really a role? Should any be merged or split?**
- **Do two actors play the same role in relation to a use case?**
- **Do the actors have intuitive and descriptive names? Can both users and customers understand the names?**



# Checkpoints: Requirements: Use-Cases

- **Is each use case involved with at least one actor?**
- **Is each use case independent of the others?**
- **Do any use cases have very similar behaviors or flows of events?**
- **Do the use cases have unique, intuitive, and explanatory names so that they cannot be mixed up at a later stage?**
- **Do customers and users alike understand the names and descriptions of the use cases?**



# Checkpoints: Requirements: Use-Case Specifications

- **Is it clear who wants to perform a use case?**
- **Is the purpose of the use case also clear?**
- **Does the brief description give a true picture of the use case?**
- **Is it clear how and when the use case's flow of events starts and ends?**
- **Does the communication sequence between actor and use case conform to the user's expectations?**
- **Are the actor interactions and exchanged information clear?**
- **Are any use cases overly complex?**





# Checkpoints: Requirements: Glossary

- **Does each term have a clear and concise definition?**
- **Is each glossary term included somewhere in the use-case descriptions?**
- **Are terms used consistently in the brief descriptions of actors and use cases?**



# 第六章 软件需求

## 主要内容

RUP中的需求流程

用例模型

术语表

补充规约

检查点

案例实践

# Review: Requirements Overview

- **What are the main artifacts of Requirements?**
- **What are the Requirements artifacts used for?**
- **What is a Use-Case Model?**
- **What is an actor?**
- **What is a use case? List examples of use case properties.**
- **What is the difference between a use case and a scenario?**
- **What is a Supplementary Specification and what does it include?**
- **What is a Glossary**

# Exercise: Requirements Overview

- **Given Problem Statement**
- **Produce following artifacts:**
  - **Use-Case Model, especially the use-case flows of events**
  - **Key abstractions/classes**
  - **The Supplementary Specification**

# 作业（本周三）

1. 软件需求分析需完成哪些任务？
2. 软件需求分析的一般性过程是什么？
3. 基于面向对象的软件需求分析的步骤及其制品是什么？
4. USE CASE Diag. 包含哪些图形符号？分别代表什么？
5. Activity Diag. 包含哪些图形符号？分别代表什么？
6. Problem Statement（问题陈述）应怎样撰写（阐述）？
7. Use case Specification（用例规约）应怎样撰写？
8. Glossary（术语表）应该怎样撰写（阐述）？
9. Supplementary（补充规约）应该怎样撰写（阐述）？

# 实验作业

- 以组为单位，针对你们组的项目，完成如下制品
  - 问题陈述
  - 用例模型（包括用例图和用例规约），至少写**5**个用例规约，写事件流时用文字和活动图
  - 补充规约
  - 术语表

# Lab 作业

- Considering your project, finishing following artifacts
  - Problem Statement
  - Usecase model containing at least 5 case specification
  - Glossary and supplementary

# 总结

- SE：系统化、规范化、可量化的开发软件；研究做到前面的**理论**、技术和工具
- 软件（软硬件）系统实在是太重要，但是还做不好，原因：规模大、复杂、缺少工具，在20世纪60th，软件危机，1968 NATO，**SE的概念**
- 软件的生命：**可行性分析-需求分析-架构设计-数据设计-子系统及其接口设计-部件设计-编码-测试-安装与部署-运维-退出**
- 理解力，面向对象方法，中英文阅读、与写作能力，**CS前序课程**，团队合作，刻苦态度，**UML**



## □ 文档

### - How to model

- 写改写的，不该写的应该不写；详略得当；真是反应；多种表达形式；

### - 在文档中，仅仅文字不够，至少图文并茂

## □ UML (8个)

### - 需求建模

✓Usecase diagram: 用例图

✓Activity diagram: 活动图

## □ 基于面向对象的需求分析步骤和制品

# 基于面向对象的软件需求分析的步骤和制品

## 步骤1：写问题陈述（**problem statement, PS**)(functional+nunFunctional req.)

- 本项目的背景和意义概述，主要功能、角色、外部环境（有交互的其他软硬件），以及主要的非功能需求

## 步骤2：析取用例，画出用例图（**usecase diagram**)(功能需求)

## 步骤3：写用例规约(**usecase specifications**)

- ✓ 写出每个用例的具体事件流(文字)
- ✓ 画出活动图

## 步骤4：写补充规约（**supplymentary doc**)

- ✓ 整个系统的非功能需求

## 步骤5：写术语表(**glossary**)

- ✓ 为今后数据类的析取奠定基础，找出系统需要存储的信息
- ✓ 解释术语

- 制品：在生产软件过程中产生的所有代码和文档， **artifact**
- 需求分析是干什么的？
  - What to do
  - 主要的任务是写出系统的功能需求和非功能需求