

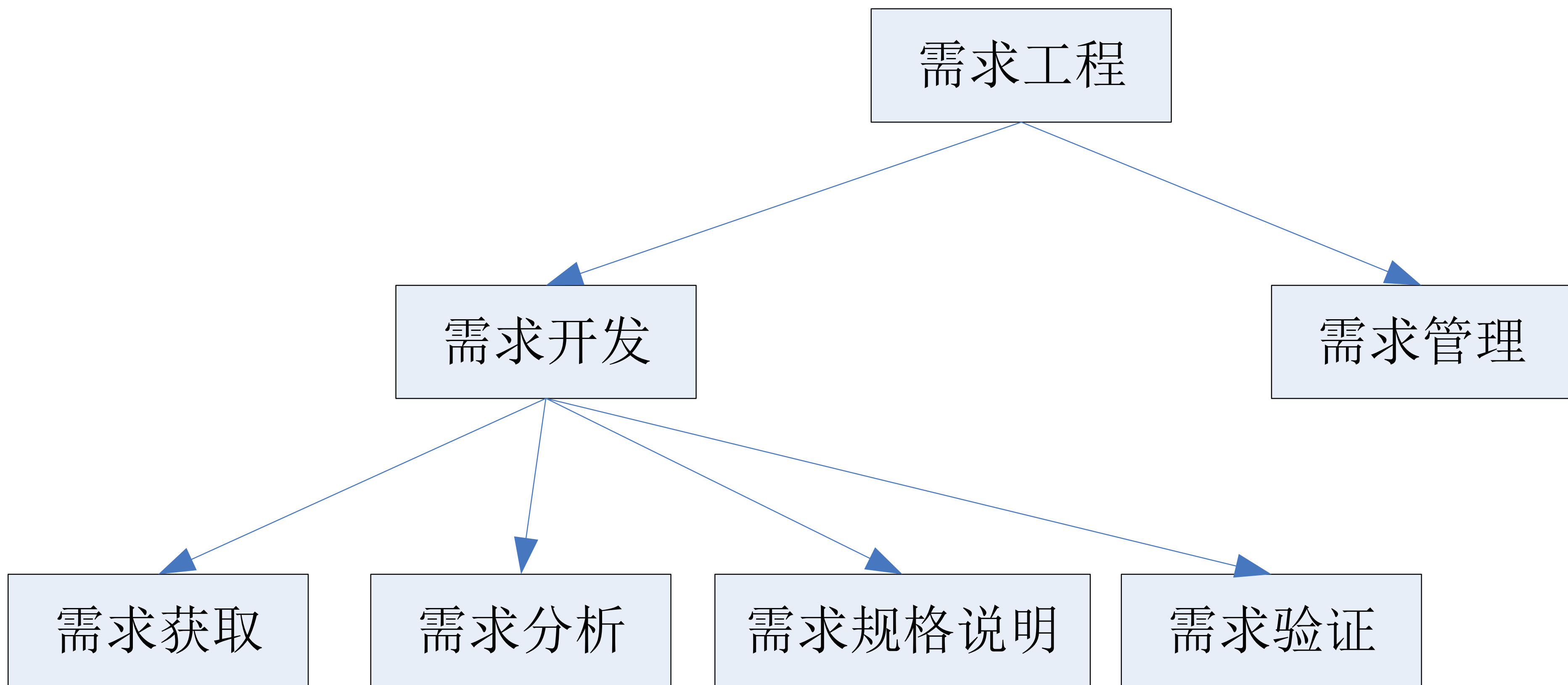
需求总结与补充

刘钦

2017 秋

第五章重点

- 需求工程
- 需求
- 需求层次性
- 需求的分类

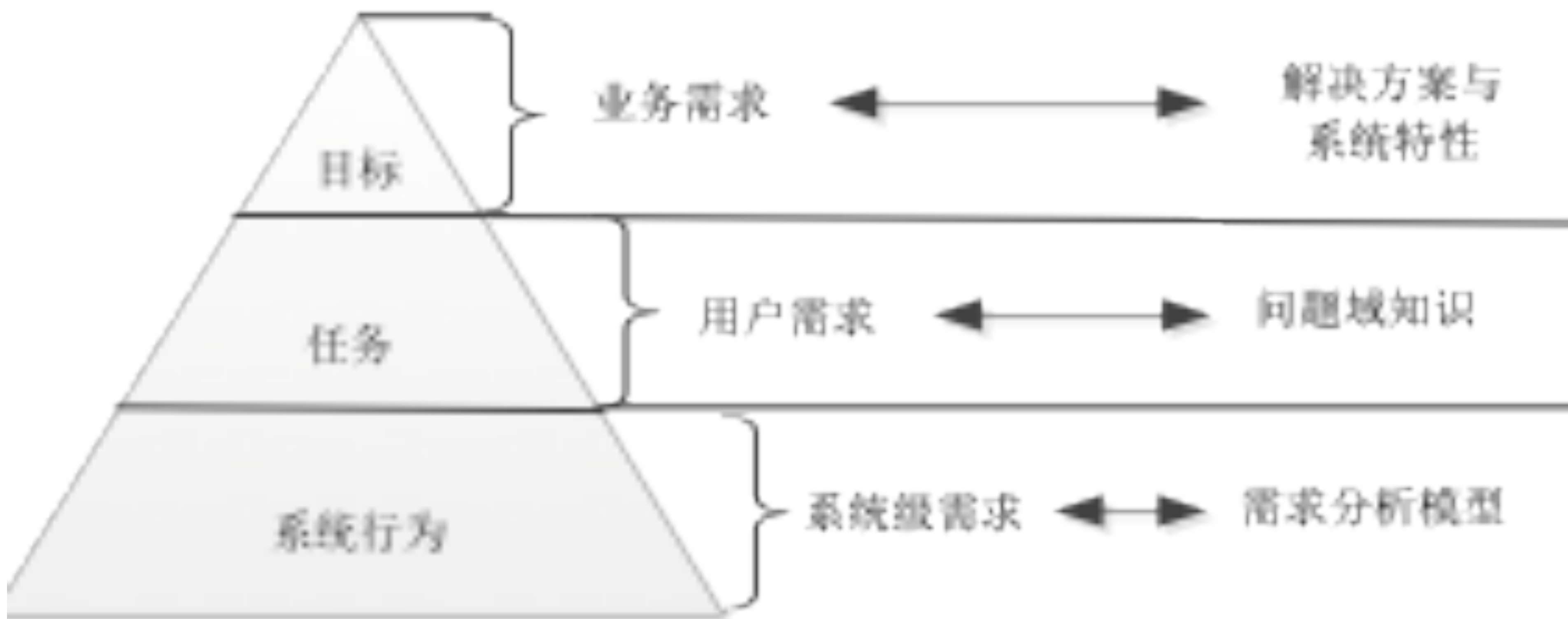


需求工程

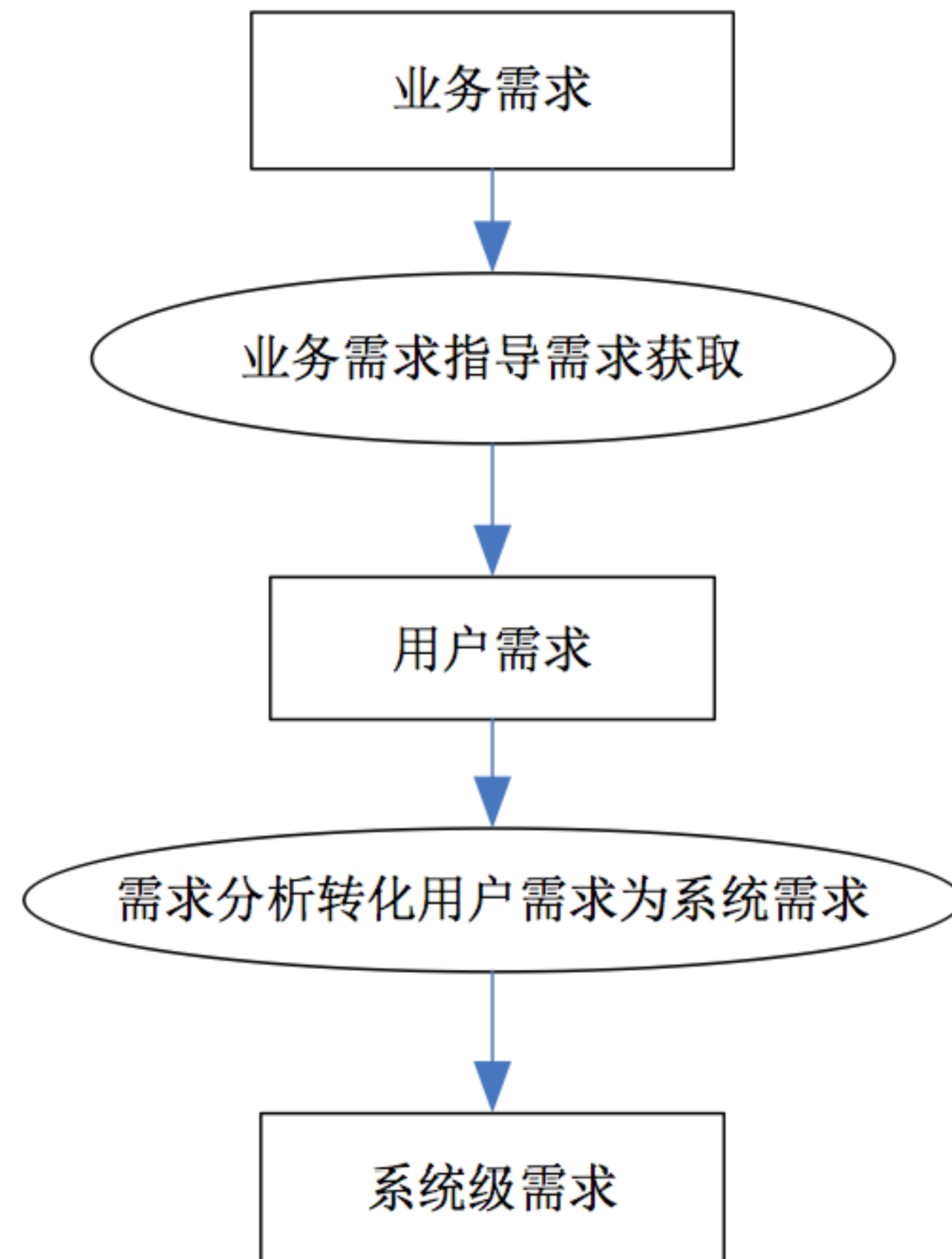
需求

- IEEE对需求的定义为[IEEE610.12-1990]:
 - (1)用户为了解决问题或达到某些目标所需要的条件或能力;
 - (2)系统或系统部件为了满足合同、标准、规范或其它正式文档所规定的要求而需要具备的条件或能力;
 - (3)对(1)或(2)中的一个条件或一种能力的一种文档化表述。

期望



需求的层次性



层次性与过程的关系

功能需求（Functional Requirement）：

- 和系统主要工作相关的需求，即在不考虑物理约束的情况下，用户希望系统所能够执行的活动，这些活动可以帮助用户完成任务。功能需求主要表现为系统和环境之间的行为交互。

性能需求（Performance Requirement）：

- 系统整体或系统组成部分应该拥有的性能特征，例如CPU使用率、内存使用率等。

质量属性（Quality Attribute）：

- 系统完成工作的质量，即系统需要在一个“好的程度”上实现功能需求，例如可靠性程度、可维护性程度等。

对外接口（External Interface）：

- 系统和环境中其他系统之间需要建立的接口，包括硬件接口、软件接口、数据库接口等等。

约束

- 进行系统构造时需要遵守的约束，例如编程语言、硬件设施等

非功能性需求

需求的分类

需求获取的困境

- 用户和开发人员的背景不同，立场不同
 - “床边B超,肝胆胰脾” —— 消除默认知识
- 普通用户缺乏概括性、综合性的表述能力
 - 不聪明的记者 —— 专业的需求人员
- 用户存在认知困境
 - 平板电脑 —— 原型
- 用户越俎代庖
 - 双机热备 —— 需求是开发人员开发出来的，不是用户提出来的
 - 我们就是要求系统能够。。。 ，至于怎么实现是你开发者的事 —— 协商
- 缺乏用户参与
 - 不愿参与的医生 —— 为用户参与提供方便

第六章重点

- 用例
- 用例图
- 结构化分析
 - 上下文图
 - 数据流图
 - 结构图
- 面向对象分析
 - 概念类图
 - 顺序图
 - 状态图
- 使用需求分析方法细化和明确需求

用例

- 用例最初由[[Jacobson1992](#)] 在 Objectory 方法中提出的,它将用例定义为“在系统(或者子系统或者类)和外部对象的交互当中所执行的行为序列的描述,包括各种不同的序列和错误的序列,它们能够联合提供一种有价值的服务”[[Rumbaugh2004](#)]。
- [[Cockburn2001](#)]认为用例描述了在不同条件下系统对 某一用户的请求的响应。根据用户的请求和请求时的系统条件,系统将执行不同的行为序列,每一个行为序列被称为一个场景。一个用例是多个场景的集合。

用例图的建立

- 目标分析与解决方向的确定
- 寻找参与者
- 寻找用例
- 细化用例

细化用例

- 如果用例的粒度不合适就需要进行细化和调整。
- 判断标准是：用例描述了为应对一个业务事件，由一个用户发起，并在一个连续时间段内完成，可以增加业务价值的任务。

细化

- 特价策略制定、赠送策略制定两个用例的业务目的、发起源和过程基本相同，仅仅是业务数据不同，所以可以合并为一个用例销售策略制定。
- 会员管理用例有两个明显不同的业务事件，可以被细化为发展会员和礼品赠送2个更细粒度的用例。
- 客户经理的库存管理用例也有三个不同的业务目标：出库、入库和库存分析，所以也应该细化为三个用例商品出库、商品入库和库存分析，其中库存分析用例与总经理的库存分析用例相同。

常见错误

- 不要将用例细化为单个操作
 - 例如,不要将用户管理细化 为增加、修改和删除三个更小的用例,因为它们要联合起来才能体现出业务价值。
- 不要将同一个业务目标细化为不同用例
 - 例如特价策略制定和赠送策略制定。
- 不要将没有业务价值的内容作为用例
 - 常见的错误有“登录”(应该描述为安全性 质量需求)、“数据验证”(应该描述为数据需求)、“连接数据库”(属性软件内部实 现而不是需求)等。

关于案例的用例的一些问题

- 顾客为什么不是参与者
- 上传下载为什么不是用例
- 系统可不可以分为服务器和客户端两个系统

概念类图的建立

- 对每个用例文本描述，尤其是场景描述，建立局部的概念类图
 - 根据用例的文本描述，识别候选类
 - 筛选候选类，确定概念类
 - 识别关联
 - 识别重要属性
- 将所有用例产生的局部概念类图进行合并，建立软件系统的整体概念类图

使用需求分析方法细化和明确需求

- 细化和明确需求内容
- 建立系统需求

细化和明确需求

- 为什么要细化
 - 用户需求的描述的模糊性和系统设计所需要的严谨性之间的矛盾
- 如何细化
 - 需求分析建模
 - 发现其中的遗漏、冲突、冗余和错误
 - 迭代（获取、分析、获取、分析。。。）

系统顺序图有助于发现交互性的缺失

1. 收银员输入会员编号;
2. 收银员输入商品;
3. 系统显示购买信息;
收银员重复 2-3 步, 直至完成所有输入
4. 系统显示总价和赠品信息;
5. 顾客付款;
6. 系统找零;
7. 系统更新数据;
8. 系统打印收据;
9. 顾客离开

概念类图有助于发现

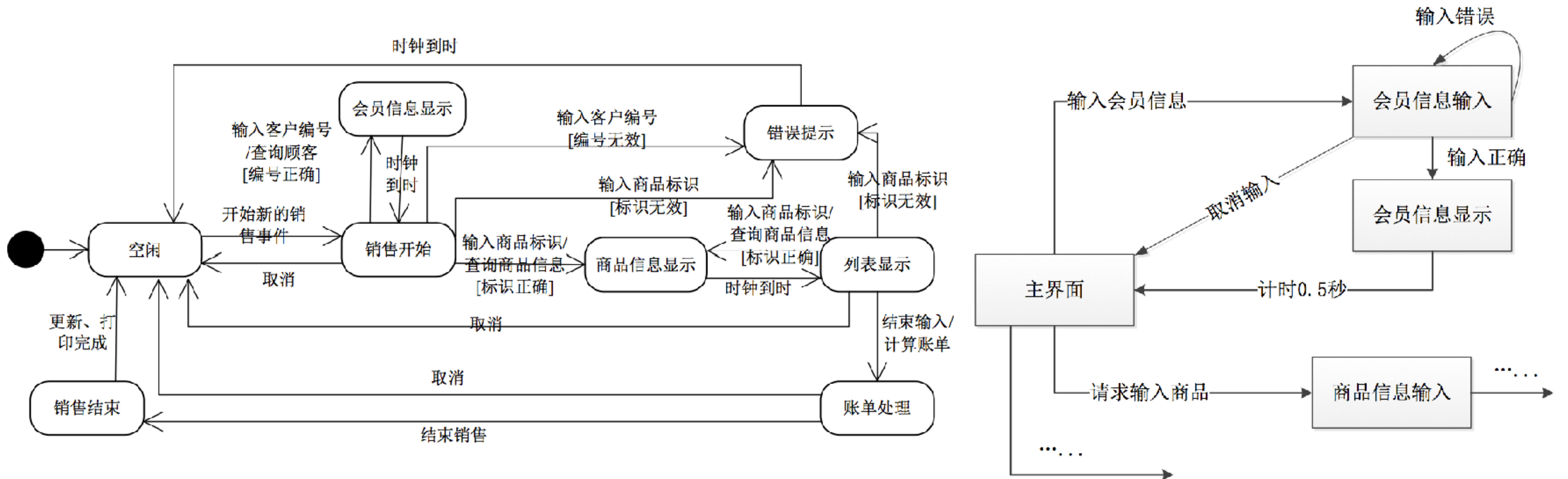
- 部分信息的使用**不准确**
 - 例如步骤 2 中输入的是商品标识,而不是商品
- 部分信息**不明确**
 - 例如会员信息、商品信息、商品列表信息、内容并没有描述。
- **遗漏**了重要内容
 - 例如总价的计算需要使用商品特价策略和总额特价策略,赠品的计算需要使用商品赠送策略和总额赠送策略。

1. 收银员输入会员编号;
2. 系统显示会员信息;
3. 收银员输入商品;
4. 系统显示输入商品的信息;
5. 系统显示所有已输入商品的信息;
收银员重复 3-5 步, 直至完成所有输入
6. 收银员结束商品输入;
7. 系统显示总价和赠品信息;
8. 收银员请求顾客付款;
9. 顾客支付, 收银员输入支付数额;
10. 系统显示应找零数额, 收银员找零;
11. 收银员结束销售;
12. 系统更新数据, 并打印收据。

总价。

的详细

状态图有助于发现界面的跳转



建立系统需求

- 8种规格说明
 - by mode I II
 - by user class
 - by object
 - by feature
 - by stimulus
 - by functional hierarchy
 - multiple organization
- 不同的分析方法适合不同的规格说明

第七章重点

- 用例文档
- 需求规格说明书

Requirements Review

- “After the payment process is complete, the relevant information should be appended to a log file.”
 - a. This requirement should be rewritten; it is incorrect.
 - b. This requirement should be rewritten; it is ambiguous or inconsistent.
 - c. This requirement should be rewritten; it is unrealistic.
 - d. This requirement should be rewritten; it is unverifiable.
 - e. This requirement is fine.

Requirements Review

- “The system should be constructed so that it will be easy to add new functionality in the future.”
 - a. This requirement should be rewritten; it is incorrect.
 - b. This requirement should be rewritten; it is ambiguous or inconsistent.
 - c. This requirement should be rewritten; it is unrealistic.
 - d. This requirement should be rewritten; it is unverifiable.
 - e. This requirement is fine.

Requirements Review

- “The price of a gasoline purchase is computed as the price per gallon for the type of gas purchased, multiplied by the number of gallons purchased (use two decimal points for representing fractions of gallons).”
 - a. This requirement should be rewritten; it is incorrect.
 - b. This requirement should be rewritten; it is ambiguous or inconsistent.
 - c. This requirement should be rewritten; it is unrealistic.
 - d. This requirement should be rewritten; it is unverifiable.
 - e. This requirement is fine.

Requirements Review

- “The system should be available 24 hours a day, 7 days a week.”
 - a. This requirement should be rewritten; it is incorrect.
 - b. This requirement should be rewritten; it is ambiguous or inconsistent.
 - c. This requirement should be rewritten; it is unrealistic.
 - d. This requirement should be rewritten; it is unverifiable.
 - e. This requirement is fine.