

软件工程导论 –

第3章 需求分析

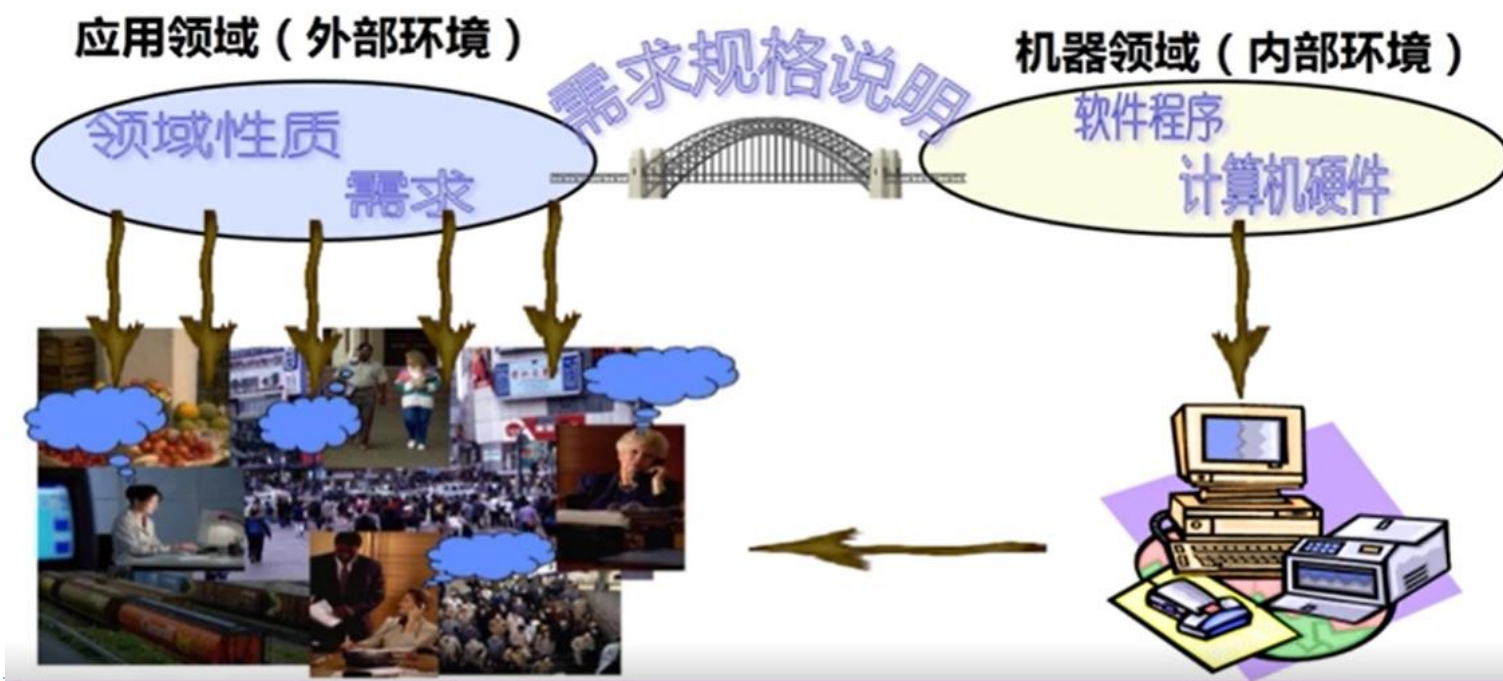
2023.05

第3章 需求分析 (Requirement Analysis)

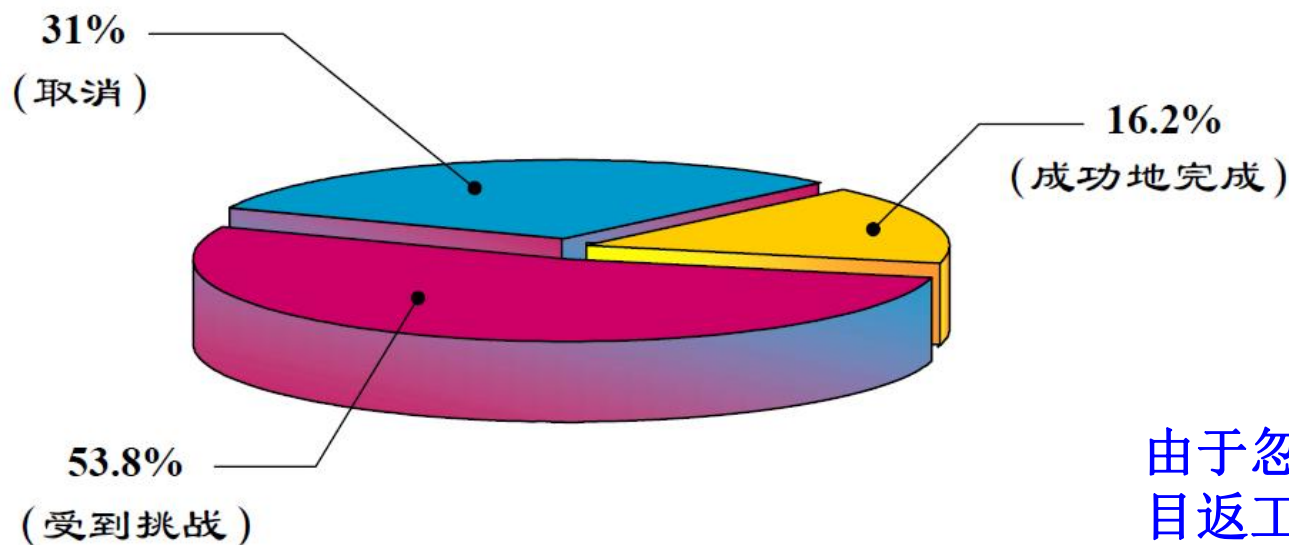
- ▶ **3.1** 需求分析的任务
- ▶ **3.2** 需求分析的过程
- ▶ **3.3** 分析建模和规格说明
- ▶ **3.4** 实体-联系图
- ▶ **3.5** 状态转换图
- ▶ **3.6** 其他图形工具
- ▶ **3.7** 验证软件需求
- ▶ **3.8** 需求管理

什么是需求？

- 需求，是人們要解決的某個問題或達到某種目的的需要。是系統或其組成部門為滿足某種書面約定（合同、標準、規範等）所具有的能力。**需求將作為系統開發、測試、驗收的正式文檔依據。** – IEEE 610.12, 1990



成功来之不易



Source: Standish Group

引述 大多数软件灾难的种子通常都是在软件项目开始的头三个月内种下的。

Caper Jones

由于忽略需求过程造成的项目返工是恶性的，大量的项目在需求阶段就注定了它的失败。

需求分析遇到的坑



客户如此描述需求



项目经理如此理解



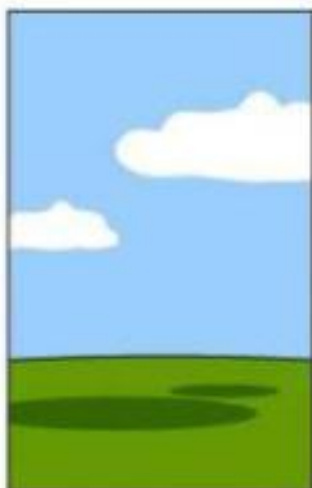
分析员如此设计



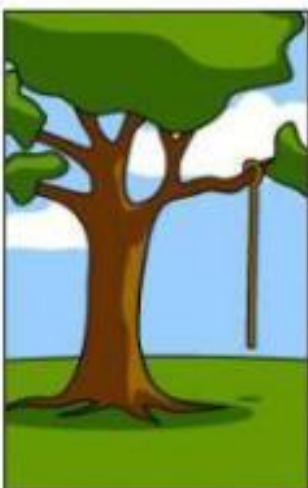
程序员如此编码



商业顾问如此诠释



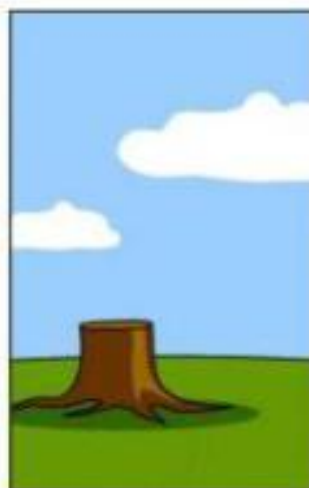
项目文档如此编写



安装程序如此“简洁”



客户投资如此巨大



技术支持如此肤浅



解密：
实际需求—原来如此

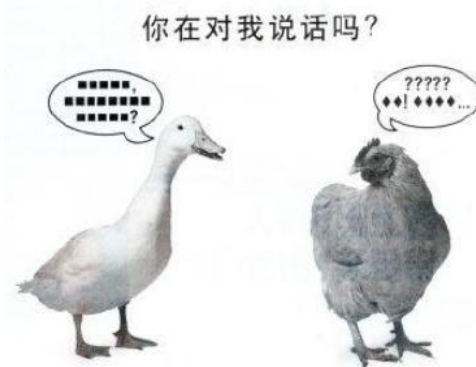
软件需求的固有困难（1）

▶ 以下是需求过程不科学的典型例子：

- 开发人员在用户处呆了两三天就埋头开发；
- 用户告诉开发人员我要开发一个XX系统，但是我很忙，你先开发一个让我看看；

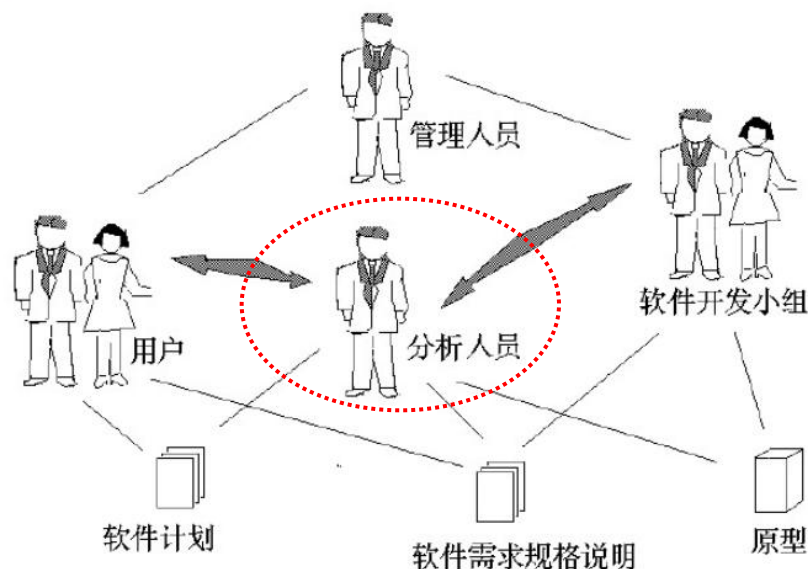
▶ 交流的困难：

我知道你相信你已经理解了你认为我所说的内容，但是我并不能肯定你已认识到你所听到的并不是我所想要的。

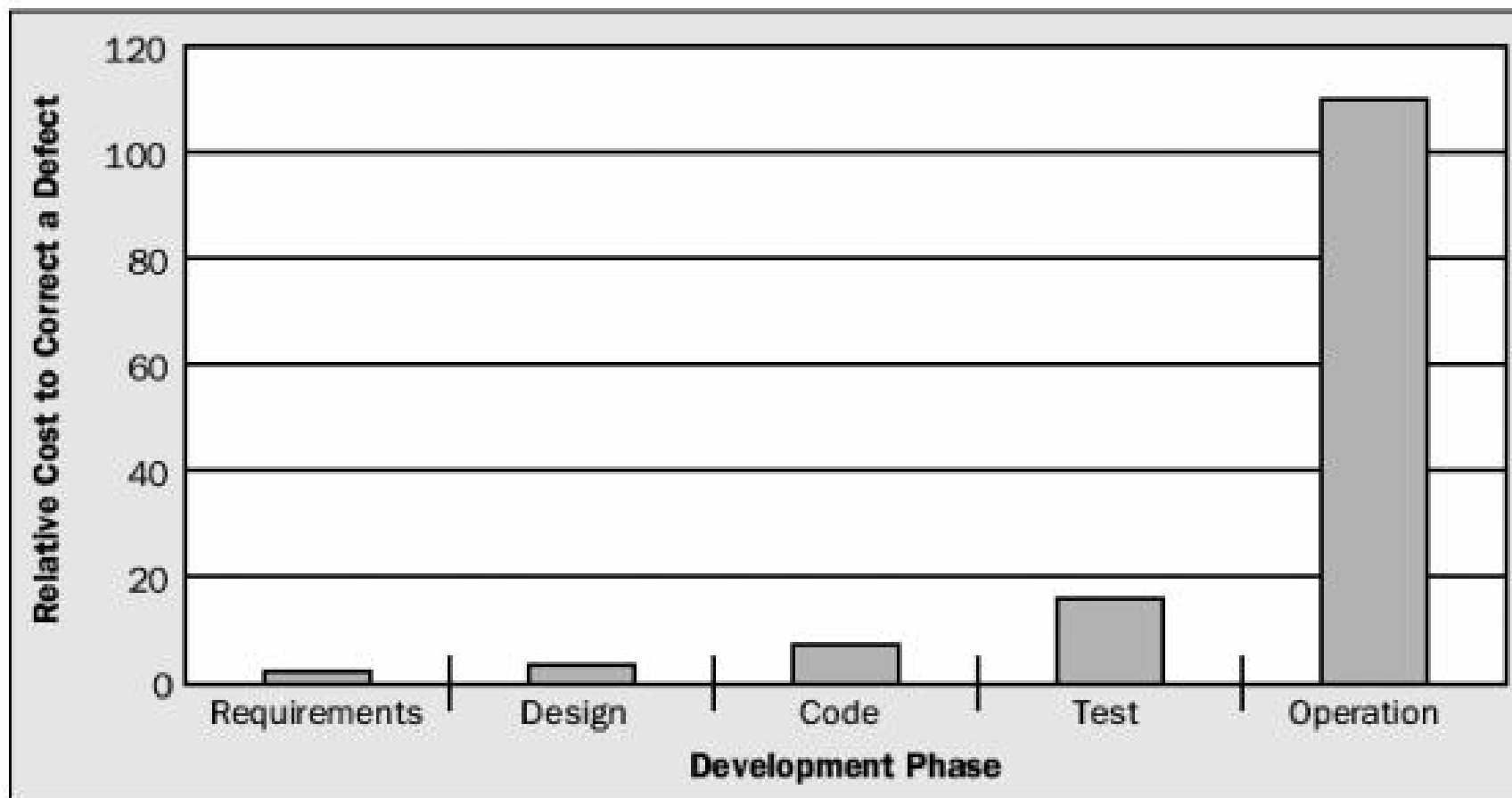


软件需求的固有困难（2）

- 用户不一定清楚的知道他到底想要什么；
- 软件开发人员不了解项目的业务背景知识；
- 日常交流所用的语言文字本身具有很大的模糊性；
- 用户企业不同部门之间需求彼此矛盾；
- 用户的需求经常会发生变化。



需求错误的成本



3.1 需求分析的任务

需求分析和规格说明阶段

目的：澄清用户的需求

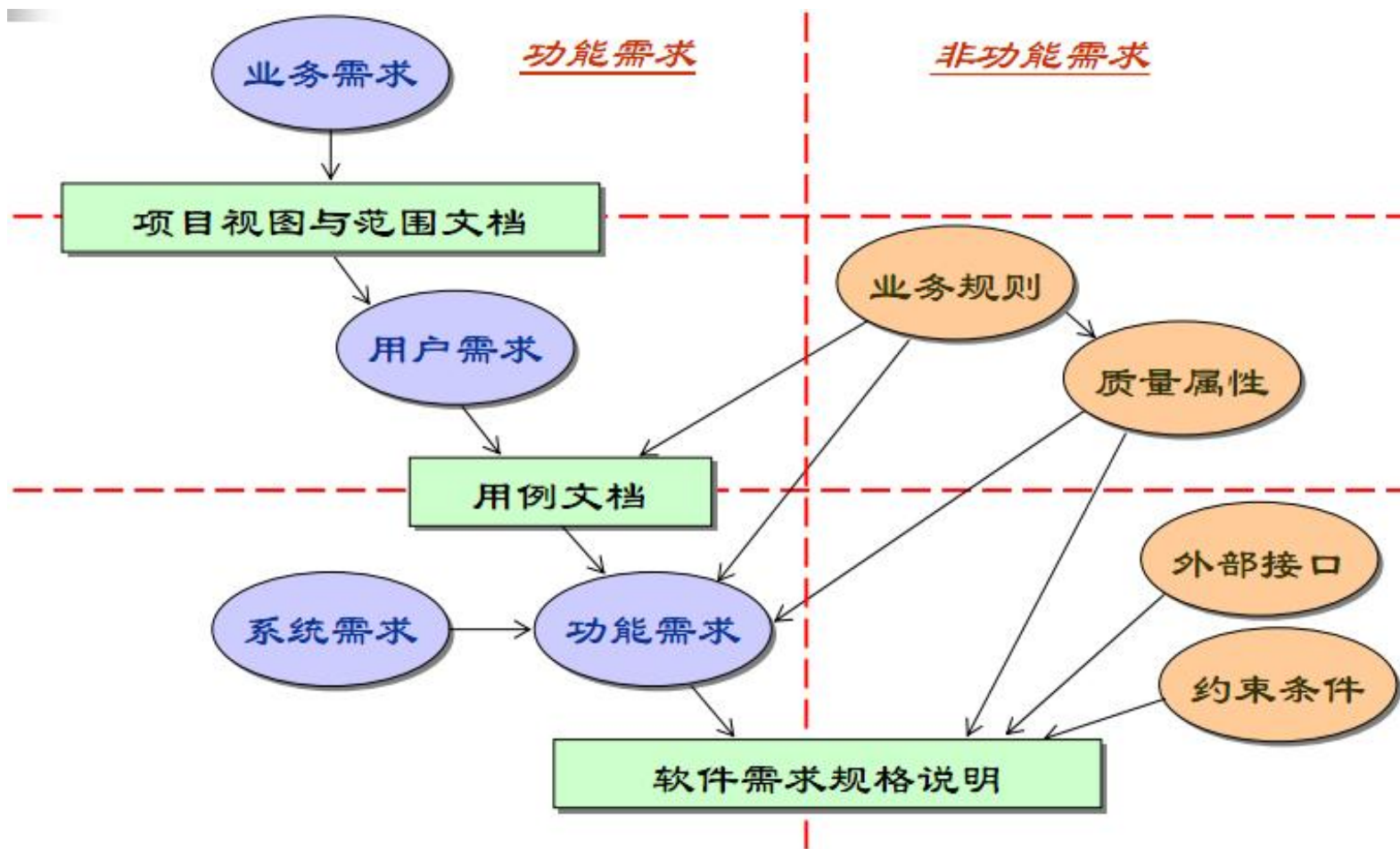
基本任务：用户和软件人员双方一起来充分地理解用户的要求，并把双方共同的理解明确地表达成一份书面文档---需求说明书

引述 需求不是架构。需求既不是设计，也不是用户接口。需求就是指需要什么。

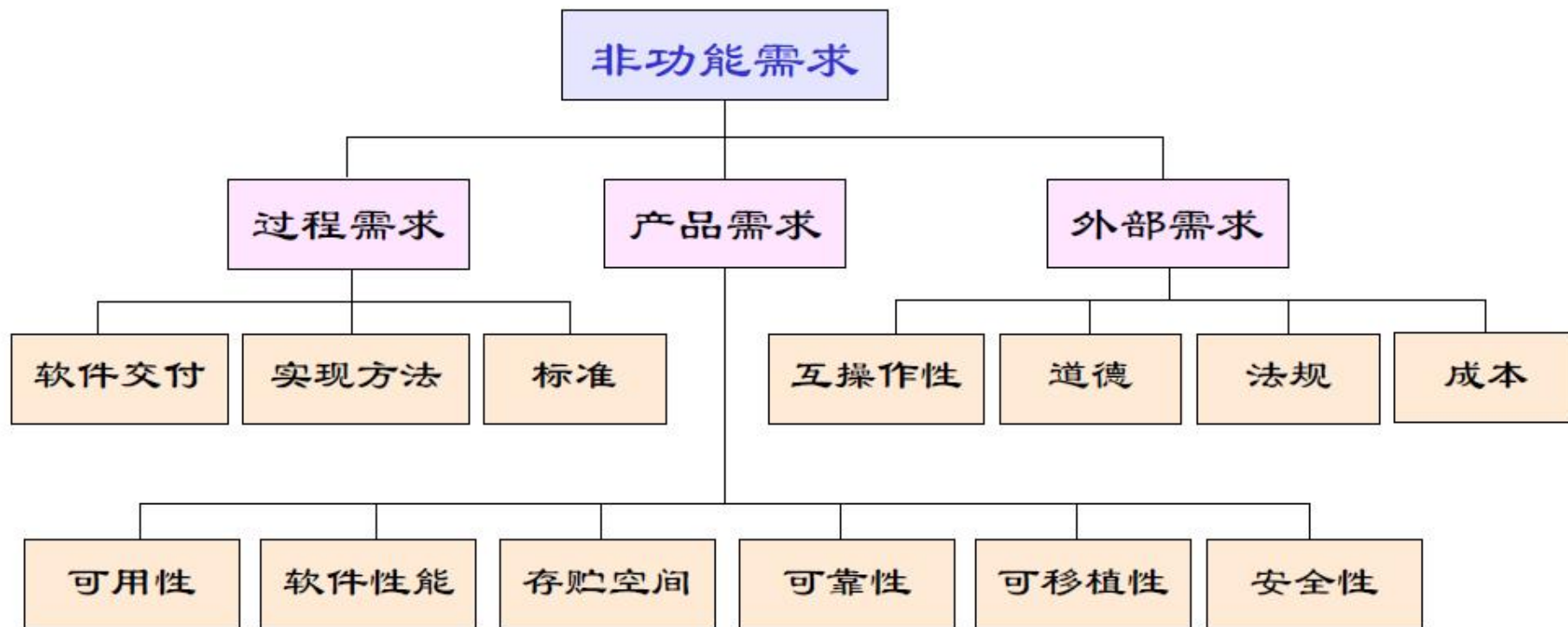
Andrew Hunt,
David Thomas

理解(做什么) → 表达

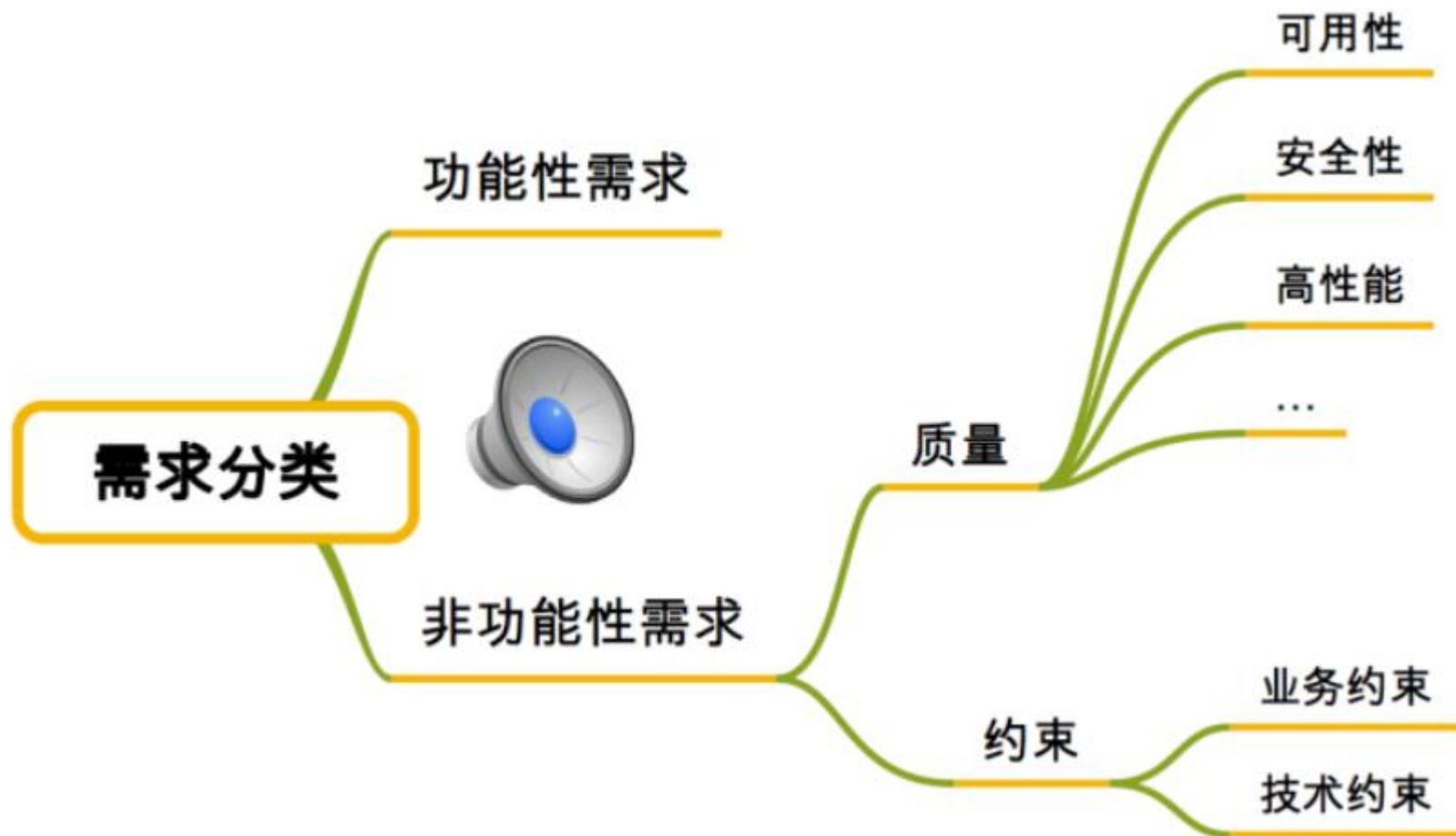
软件需求各部分之间的关系



非功能需求



软件需求分类图



<https://www.cnblogs.com/djessel/p/10588526.html>

思考问题

- ▶ 针对一个图书管理系统，分析以下描述，哪些是功能需求，哪些是非功能需求，哪些不属于需求？

– 普通读者必须注册成合法用户才可以使用该系统。

功能需求

– 用户可以预订目前借不到的图书资料。

功能需求

– 用户不希望自己的借阅记录被他人查阅。

功能需求

– 系统通过 ADO 与图书资料数据库连接，并从图书资料数据表中获得图书资料的基本信息。

一般不是需求，属于实现方法。但如果有特殊实现要求，也可以作为外部需求。

– 系统应该具有良好的可扩展性。

非功能需求

- **思考：**如果所开发的软件系统只是简单地替代现行的手工操作方式，是否可行？为什么？

3.2 需求分析的过程

需求分析阶段可分为四个过程：

- (1) 调查研究、
- (2) 分析与综合、
- (3) 书写需求分析的文档
- (4) 评审。

需求的来源（以图书管理系统为例）

- 客户或用户

干系人(stakeholders)

- 学院的高层管理者、项目投资入
- 系统管理员
- 教师、学生、图书管理员

- 标准

- 图书资料的标准

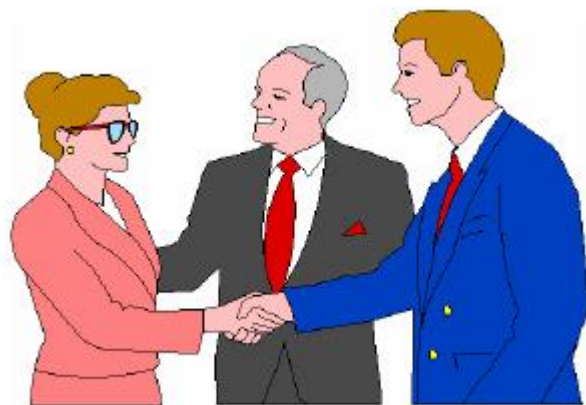
- 政策或法律

- 图书资料管理规程、知识产权和版权保护等

- 系统或过程文档

- 当前手工管理的文件、表格、记录等

- 相关领域的专家



需求分析的过程 - (1) 调查研究

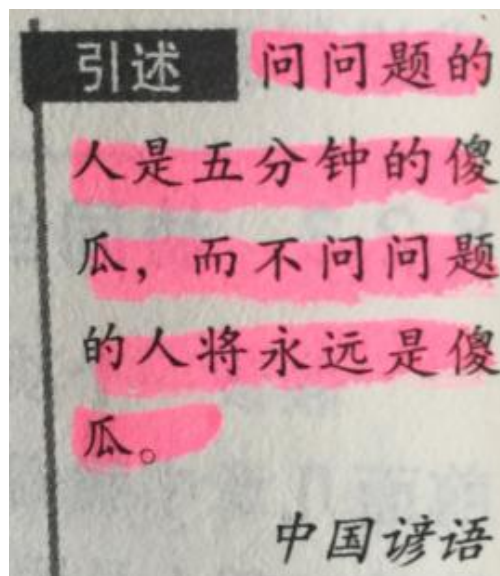
- ▶ 系统分析员协同程序员向用户做需求调查，阅软件计划中的可行性报告和项目开发计划报告，访问系统现场，并由此确定当前系统必须做什么，并获得当前系统的具体模型，用数据流图、UML图或IPO图表示出来。

需求获取技术

- 需求获取的关键在于通过与用户的沟通和交流，收集和理解用户的各项要求。

- 需求获取技术

- 用户面谈
- 需求专题讨论会
- 问卷调查
- 现场考察
- 原型化方法
- 基于用例的方法



需求获取时的注意点

CSDN: 你对需求分析怎么看? 你后来还提出了“客户需求可行性分析”理论, 能大概介绍其关键点吗?

<https://fangang.iteye.com/category/200994>

范钢: 这些年做了那么多软件项目, 有成功的、有失败的, 真的切身感受到, 需求分析的成功与否, 非常关键地决定了一个项目的成功与否。无数令人绝望的项目都是始于不到位的需求分析。然而, 在实际工作中, 我们的需求分析工作既低效又无所适从。通过这一系列的文章, 我点出了需求分析工作问题的核心, 那就是被动式的需求分析, 即客户说什么就是什么, 一切都以客户说的为准。客户往往是非技术的, 因此客户提出的需求往往也是非常理想而难以实现的。用这样的需求去指导开发, 结果可想而知。

正确的做法应当是主动式的需求分析, 即跳出客户需求的本身去主动学习与理解客户的业务领域, 分析客户需求背后的动机, 从而站在更加深刻的角度去理解客户的需求, 然后结合技术实现去分析客户需求隐藏的问题, 这就是所谓的“客户需求可行性分析”, 即分析客户需求是否可行。最后, 站在客户的角度去探讨其中的问题, 同时提出更加可行的方案。

这里的关键问题就在于我们对客户业务领域的理解到底有多深。我们对业务领域理解得越深, 我们对客户需求的理解就越深, 分析的问题也就越深, 与客户探讨的过程中就越能被客户认可, 从而提出为客户认可与接收的技术方案。这样的方案, 既准确表达了客户的需求, 又可以在技术上是可行的。按照这样的方案设计开发软件项目, 才能真正有效地规避需求方面的风险。这种思路在实践中得到了许多的成功, 并且让无数的网友脑洞大开。

3.2 需求分析的过程 – （2） 分析与综合

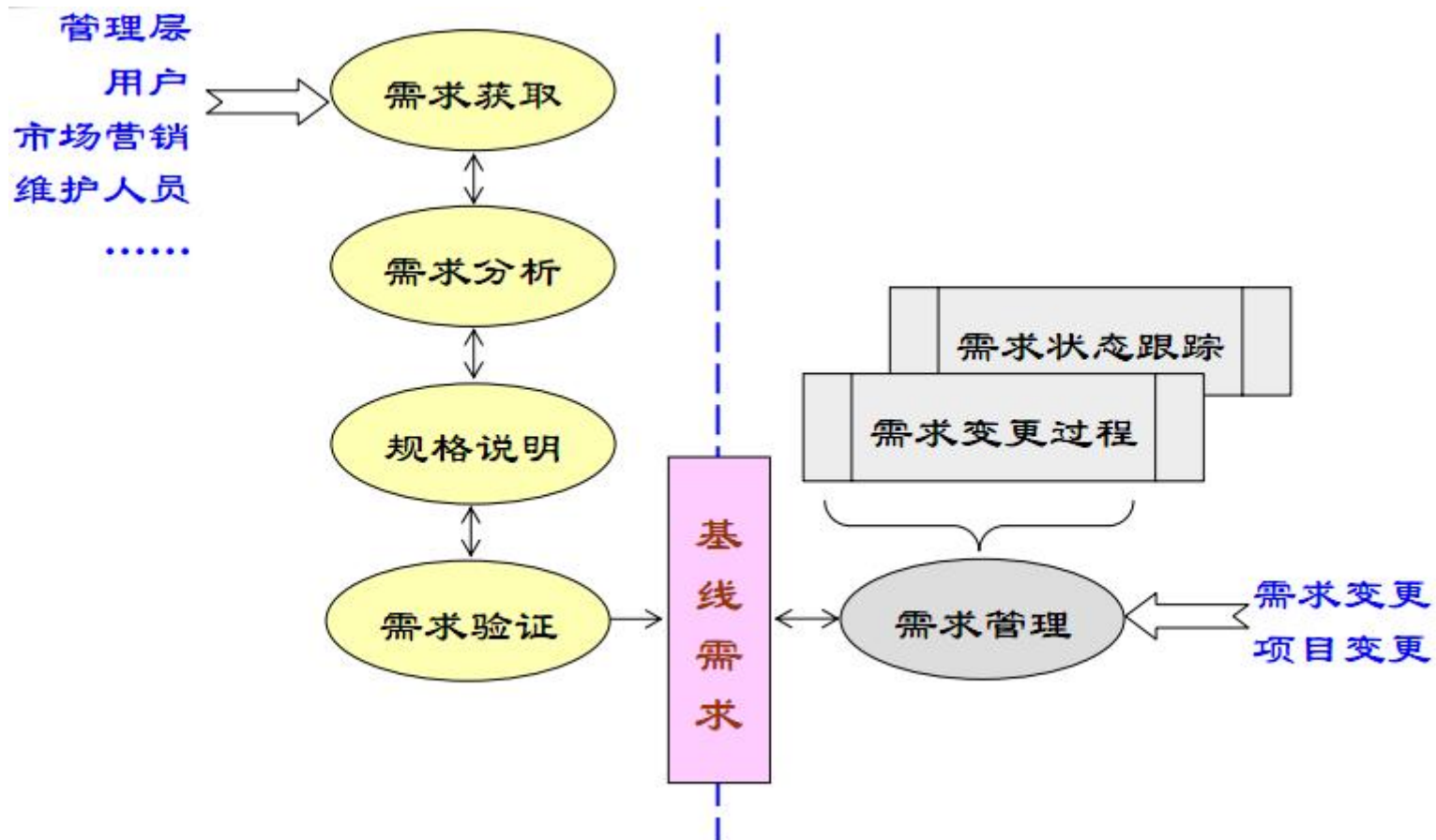
分析员需从**数据流**和**数据结构**出发，逐步细化所有的软件功能，找出系统中各元素之间的联系、接口特征和设计上的限制，分析它们能否满足功能要求，是否合理。

依据**功能需求**、**性能需求**、**运行环境需求**等，剔除其中不合理的部分，增加其需要的部分。最终综合成系统的解决方案后，给出目标系统的详细逻辑模型。

3.2 需求分析的过程 – (3) 书写和 (4) 评审

- ▶ (3) 书写需求分析的文档
- ▶ 把分析的结果用正式的文档记录下来，作为最终软件配置的一个组成成分。应该完成下述四份文档资料：系统规格说明、数据需求、用户系统描述、修正的开发计划。
- ▶ (4) 需求分析评审
- ▶ 作为需求分析阶段的复查手段，在需求分析的最后一步，应该对功能的正确性、完整性和清晰性，以及其他需求给予评价。

需求工程过程详细



需求分析的过程步骤

— 定义系统的边界

- 建立系统与其外部实体间的界限，明确接口处的信息流。

— 分析需求可行性

- 分析每一个需求实现的可行性，确定与实现相关的开发风险。

— 确定需求优先级

- 需求优先级有助于开发组织和版本规划。

— 建立需求分析模型

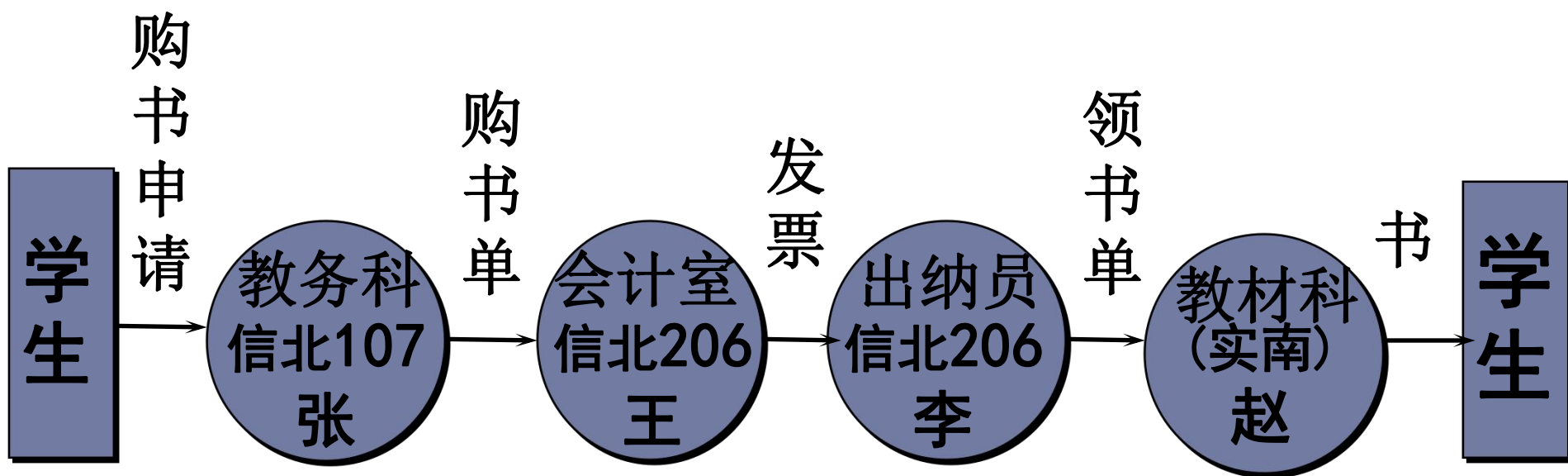
- 通过建立需求的多种视图，揭示出需求的不正确、不一致、遗漏和冗余等更深的问题。

— 创建数据字典

- 确保客户和开发人员使用一致的定义和术语。

需求分析的过程1（以图书购买系统为例）

1 通过对现实环境的调查，获得当前系统的物理模型。

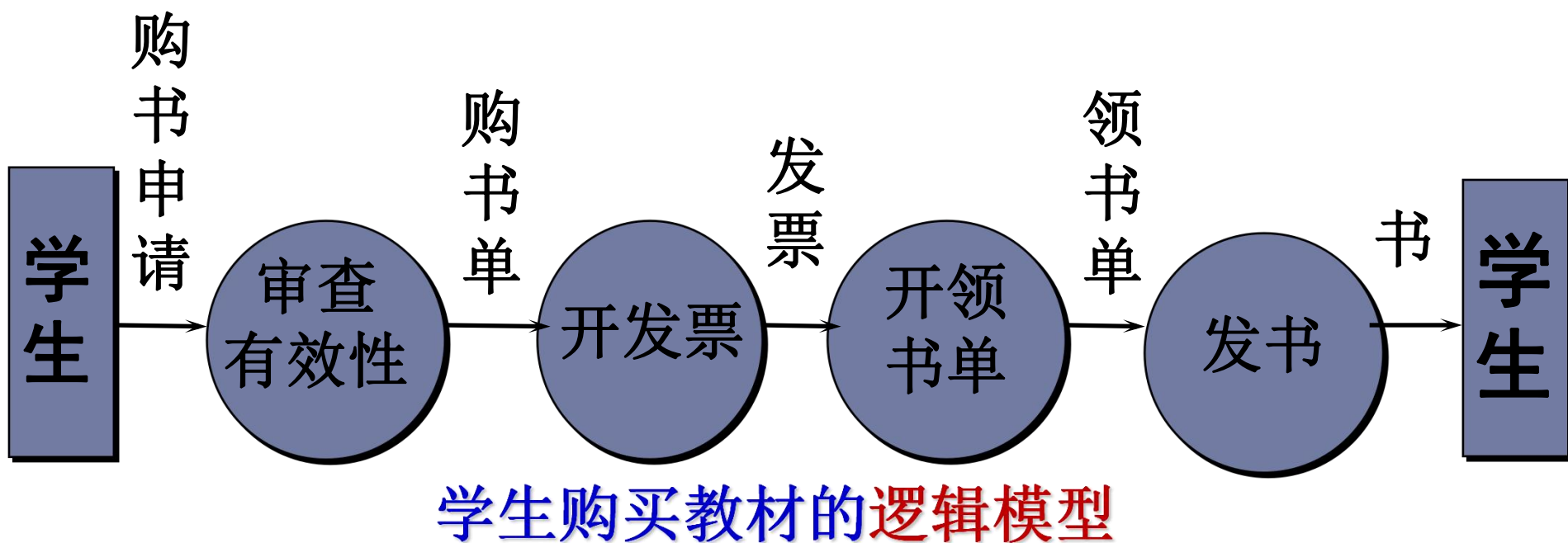


学生购买教材的实际处理流程：当前系统物理模型

需求分析的过程2

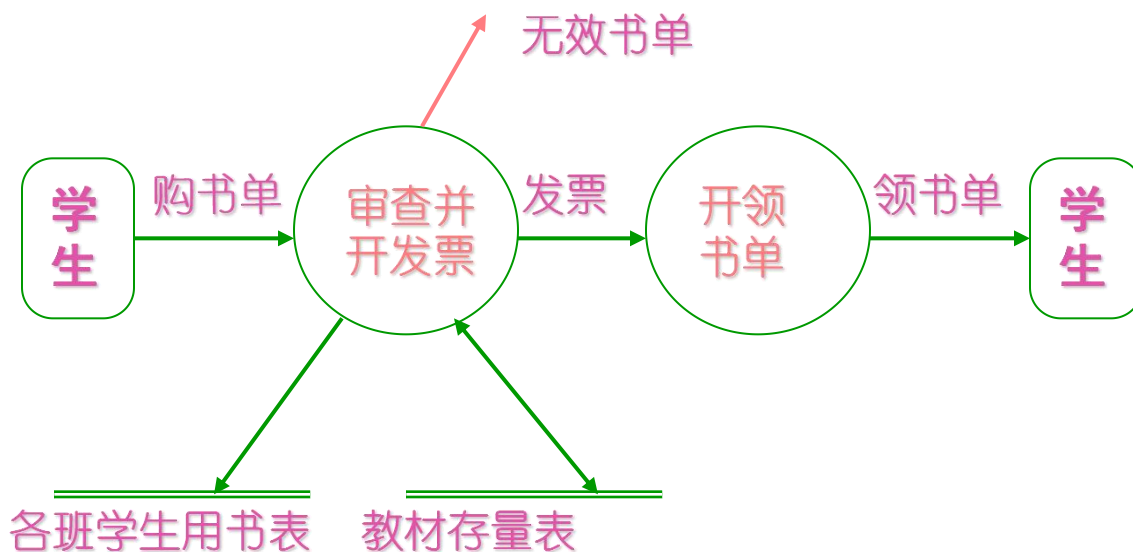
2 去掉具体模型中的非本质因素：

抽取现实系统的实质，抽象出当前系统的逻辑模型。



需求分析的过程3

3 分析当前系统与目标系统的差别，建立目标系统的逻辑模型。



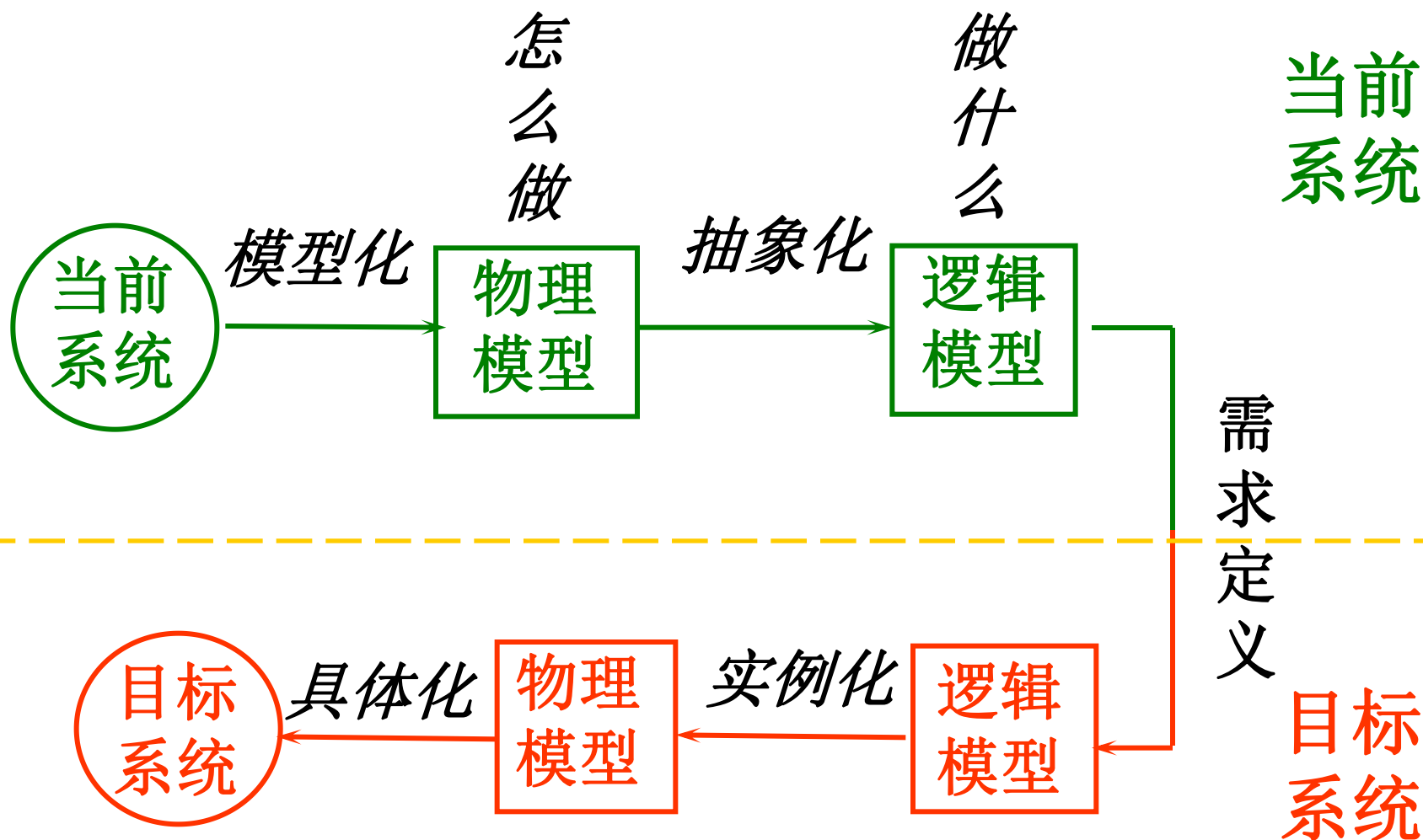
计算机教材管理系统的逻辑模型

需求分析的过程4~5

4 对目标系统的逻辑模型进行细化、改进与优化

5 需求分析的验证（需求评审）

需求分析的过程总括



需求分析的过程总括

- ▶ 模型是对对象系统的形式化的特征抽象，概括性或近似地表示；
- ▶ 构造模型的过程是一个抽象、分析的过程。



模型构造的过程

	逻辑模型 (本质模型、概念模型)	物理模型 (实施模型、技术模型)
现行系统	描述重要的业务功能，无论系统是如何实施的。	描述现实系统是如何在物理上实现的。
目标系统	描述新系统的主要业务功能和用户新的需求，无论系统应如何实施。	描述新系统是如何实施的（包括技术）。

需求分析的具体任务总结：

- (1) 确定软件系统的综合需求
- (2) 分析系统的数据需求
- (3) 导出软件系统的逻辑模型
- (4) 修正系统开发计划
- (5) 开发原型系统
- (6) 验证软件需求分析的正确性
- (7) 编写软件需求规格说明书

3.3 分析建模与规格说明

- ▶ 结构化分析方法（**Structured Analysis**，简称**SA**方法）是**70**年代中期提出的一种面向数据流、自顶向下、逐步求精进行需求分析的方法。
- ▶ 结构化分析方法适用于分析大型的数据处理系统，特别适用于企事业管理系统。
- ▶ 结构化分析方法通常与设计阶段的结构化设计方法（**Structured Designed**，简称**SD**方法）衔接起来使用。

结构化分析方法 – 过程指导

▶ 建立系统的功能模型

利用数据流图（**Data Flow Diagram, DFD**）将大问题分割成若干个小问题，采用分层的数据流图，先建立系统环境图（顶层），再逐步求精。

▶ 建立数据字典

采用结构化英语，小说明，补充材料等列出。

结构化分析方法步骤和工具

▶ 步骤

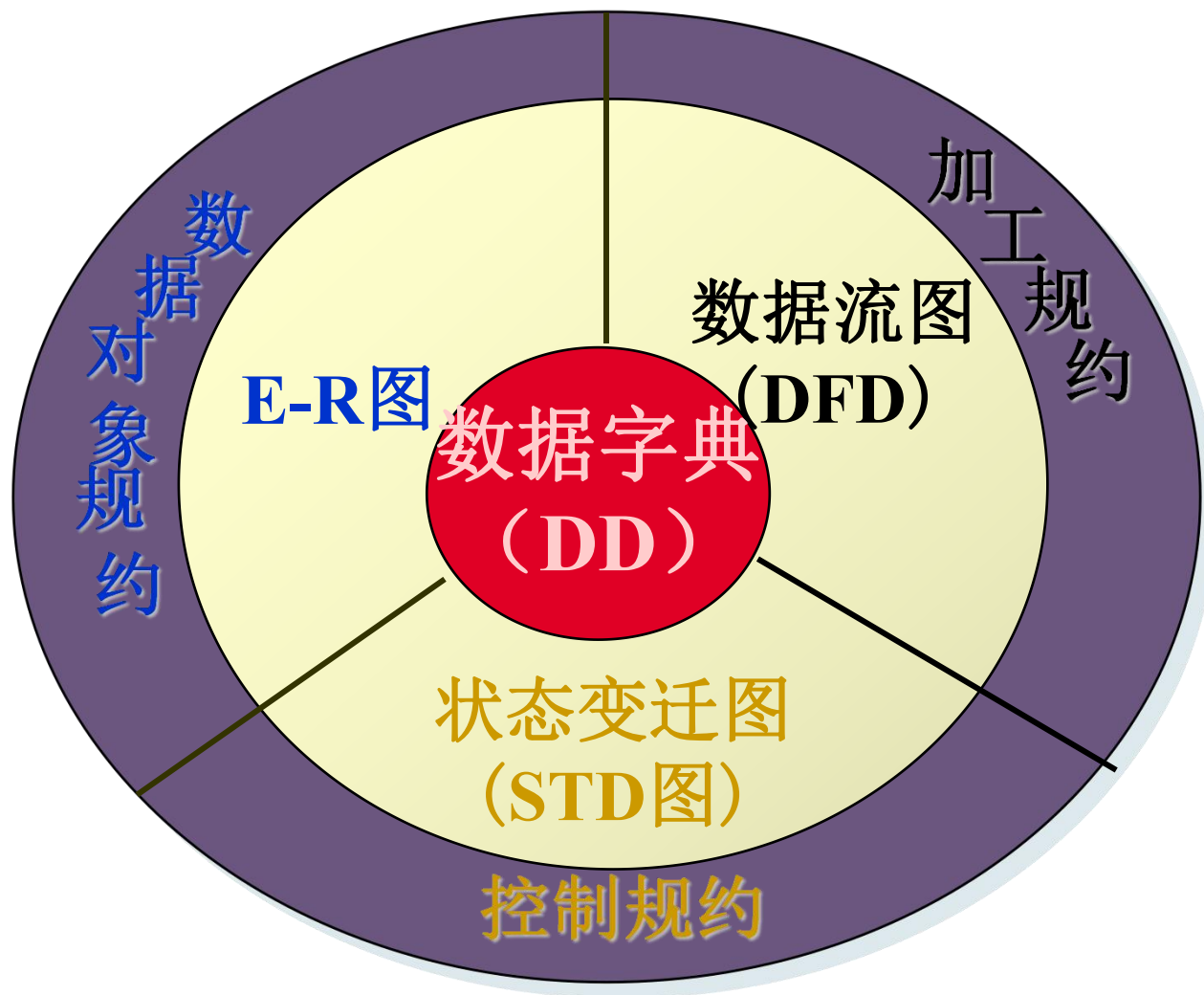
理解→ 获得具体模型→ 抽象出逻辑模型

▶ 工具

- ▶ 一般功能分析工具：DFD、数据字典、结构化英语、判定表、判定树等。
- ▶ 一般行为分析工具：状态迁移图、Petri网等。
- ▶ 一般数据分析工具：ER图或者EER(扩展ER)图。

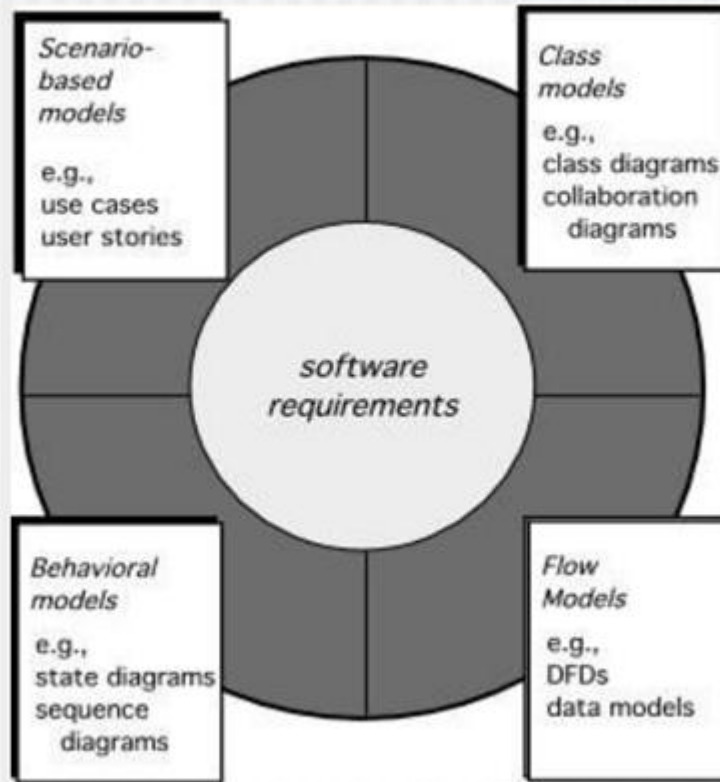
- ▶ **SA**主要针对数据处理领域，因此，系统分析的侧重点在于功能分析和数据分析，而行为分析使用得较少。

结构化分析（SA）方法建模图



面向对象的需求分析建模图

Elements of Requirements Analysis



分析模型的构成元素

▶ 数据字典(DD)

- ▶ 模型核心，包含了所有数据对象的描述的中心库。

▶ E-R图(ERD)

- ▶ 表示数据对象以及相互的关系，用于数据建模。

▶ 数据流图(DFD)

- ▶ 指明数据在系统中移动时如何被变换；
- ▶ 描述对数据流进行变换的功能；
- ▶ DFD中每个功能的描述包含在加工规约(小说明)。
- ▶ 用于功能建模。

▶ 状态变迁图(STD)

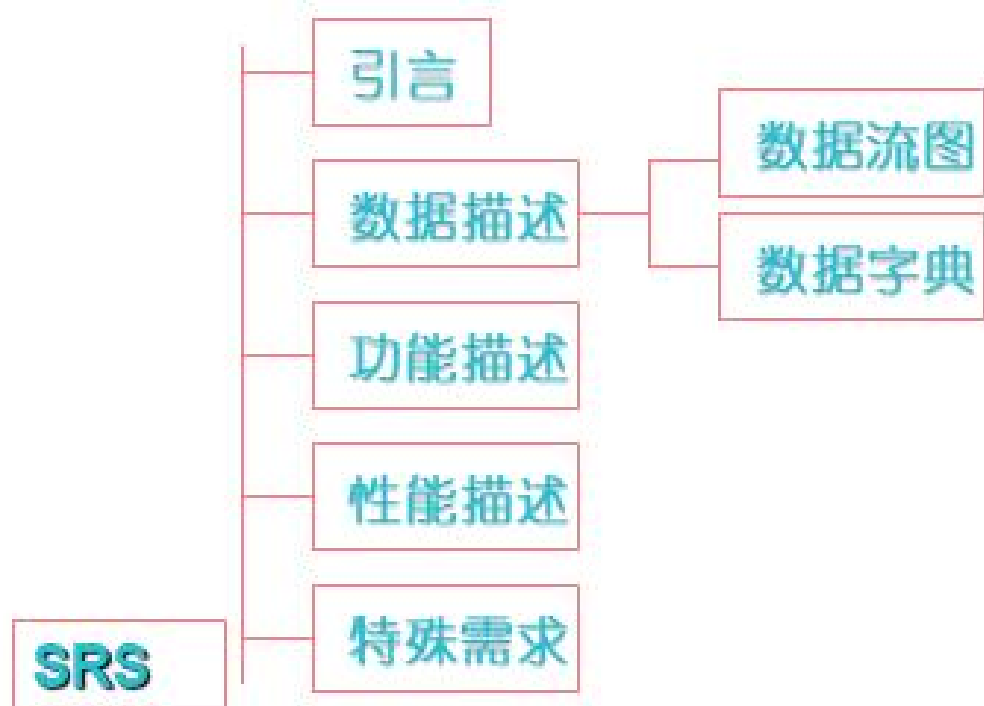
- ▶ 指明作为外部事件的结果, 系统将如何动作。用于行为建模。

SA方法局限性

- ✧ 数据库
- ✧ 人机界面
- ✧ 实时系统

软件需求规格说明书 ---- SRS(Software Requirement Specification)

- 需求规格说明 (*SRS, Software Requirement Specification*) 精确地阐述一个软件系统必须提供的功能和性能以及它所要考虑的限制条件。



需求说明书的三个作用

- ▶ 作为用户和软件人员之间的合同，为双方相互了解提供基础
- ▶ 反映出问题的结构，可以作为软件人员进行设计和编写的基础
- ▶ 作为验收的依据，即作为选取测试用例和进行形式验证的依据

用简单的话来说明需求过程，就是确定系统该做些什么以及该符合什么条件。

软件需求规格书内容

1. 介绍

- 1.1 目的
- 1.2 文档结构
- 1.3 预期读者
- 1.4 产品范围
- 1.5 参考资料

2. 总体描述

- 2.1 产品远景
- 2.2 产品功能
- 2.3 用户分类和特征
- 2.4 操作环境
- 2.5 设计和实现约束
- 2.6 用户文档
- 2.7 假设和依赖

3. 外部接口需求

- 3.1 用户接口
- 3.2 硬件接口
- 3.3 软件接口
- 3.4 通讯接口

4. 用例

- 4.1 用例名称1
 - 4.1.1 用例描述
 - 4.1.2 业务规则
 - 4.1.3 功能性需求
- 4.x 用例名称x

5. 非功能需求

- 5.1 性能需求
- 5.2 安全性需求
- 5.3 权限需求
- 5.4 软件质量属性
- 5.5 业务规则（全局）

6. 其他需求

7. 全局功能定义（可选）

附录1. 术语表

附录2. 分析模型

附录3. 已确定的需求列表

不应该包括在SRS中的内容

— 项目开发计划

- 诸如成本、人员、进度、工具、方法等

— 产品保证计划

- 诸如配置管理、验证与测试、质量保证等

— 软件设计细节

- 需求通常用于表达“做什么”，而不描述“如何做”。



如何描述产品或产品构件需求 - 1

▶ 商业需求

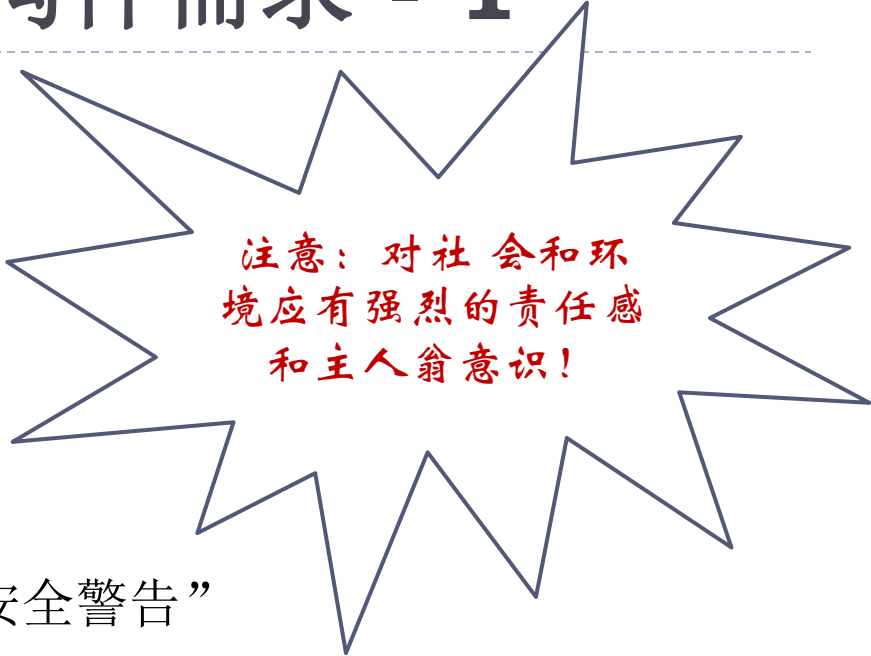
- ▶ 客户或商业利益的描述
 - ▶ “提高市场占有率X%”
 - ▶ “每年节约电费Y¥”
 - ▶ “每年节约维护成本Z¥”

▶ 商业规则

- ▶ 描述一个方针、标准或规律
 - ▶ “只有现场操作人员才能越过安全警告”
 - ▶ “必须符合FDA标准的137-B”

▶ 用例或场景

- ▶ 描述一个用户需要执行的任务
 - ▶ “为一个包裹打印邮件标签”
 - ▶ “管理需要分析的一系列样本”



注意：对社会和环境应有强烈的责任感和主人翁意识！

如何描述产品或产品构件需求 - 2

▶ 外部接口需求

- ▶ 描述系统与外部世界的关联
 - ▶ “必须输出一个CSV格式的文件”
 - ▶ “必须读取从XXX系统传过来的信号”

▶ 质量属性

- ▶ 描述系统如何很好地执行某些功能
 - ▶ “系统必须强壮、直观、可靠”
 - ▶ “复杂数据的响应时间不能超过60秒”

▶ 约束

- ▶ 对设计和实现的限制
 - ▶ “必须在4M内存上运行”
 - ▶ “数据必须向前版本兼容”
 - ▶ “必须使用Oracle9.1或之后的运行引擎”

如何描述产品或产品构件需求 - 3

▶ 数据定义

- ▶ 描述一个数据项或结构
 - ▶ “费用编号的格式是一个3字节的部门前缀，一个连字号，和一个6位项目编码”
 - ▶ “控制温度的缺省值是37度”

▶ 方案构想

- ▶ 建议一个可能解决问题的途径
 - ▶ “用户从下拉框中选择期望的值”
 - ▶ “用JAVA编写代码”

▶ 功能需求

- ▶ 表明系统在确定条件下的行为
 - ▶ “如果压力超过了40，则压力警告灯应该亮”
 - ▶ “用户可以按字母顺序排列项目列表”

写作要点

- ▶ **简洁**：保持语句和段落的简短；
- ▶ **一致**：上下文用语一致，陈述风格应该具有一致的样式。
- ▶ **具体**：例如：系统必须…；或者；用户必须…要紧跟着一个行为动作或可观察的结果。举例：计算过程中出现除零错误时，系统必须给出提示信息（**不好！不具体**）； 计算过程中出现除零错误时，系统必须立即弹出对话框显示该错误，并进行声音提示（**好！**）。

※必须避免模糊的、主观的术语，减少不确定性

避免：用户友好、容易、简单、美观、迅速、有效、支持、许多、最新技术、优越的、可接收的和健壮的（人和猫理解的美女词义一样嘛☺）

※避免使用比较性词汇，例如：提高、最大化、最小化、最佳化。定量地说明所需要提高的程度或者说清一些参数可接受的最大值和最小值（如“提高投影仪的投影高度”，每个人理解的高度一样嘛？）

※不应该把多个需求集中在一个冗长的叙述段落中。务必记住不要在需求说明中使用“和/或”，“等等”之类的连词（这个等，对于测试人员来说就是灾难…）

编写需求文档的几点建议

- ▶ 句子和段落要短，要采用主动语气，使用术语时要保持一致；
- ▶ 从客户的角度看看需求是否被有效的定义；
- ▶ 从测试人员的角度看看是否便于系统测试用例的编写；
- ▶ 正确的把握细化程度，要避免包含多个需求的长的叙述段落；
- ▶ 不要使用“也许”，“大概”，“可能”这样的表达模糊性质的词汇；
- ▶ 通篇文档细节上要保持一致；
- ▶ 多使用图和表；

3.4 实体-联系图

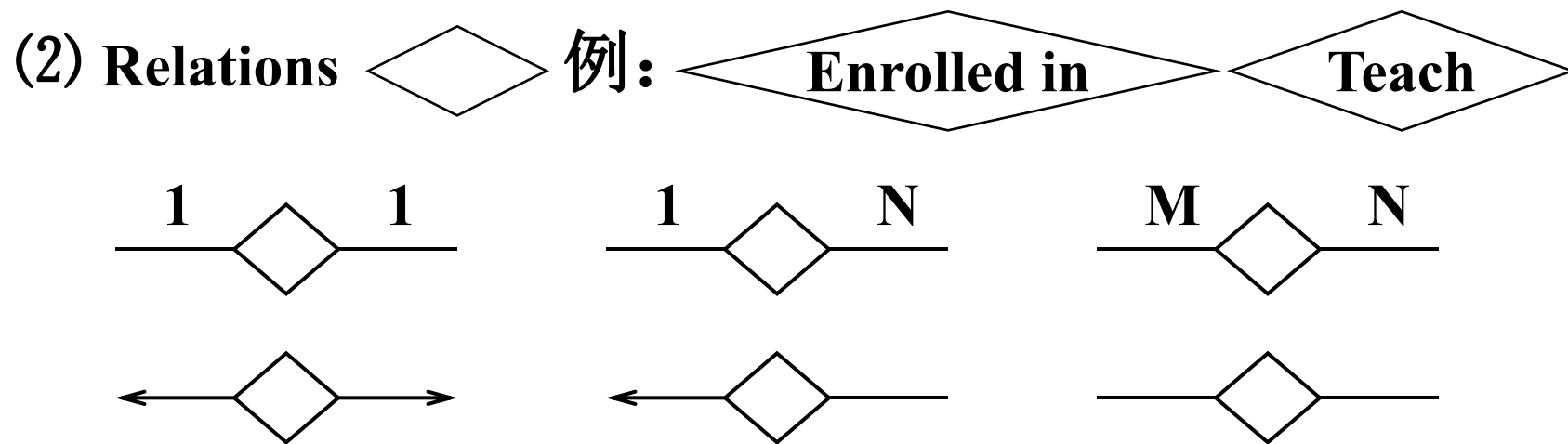
- ▶ **数据建模**最常用的表示概念性数据模型的方法，是实体联系方法（Entity-Relationship Approach）
- ▶ ER图描述现实世界中的实体，而不涉及这些实体在系统中的实现方法。

3.4 实体-联系图

(1) **Entities**  例:  **Student** ,  **Instructor** ,  **Class**

实体是客观世界中存在的且可相互区分的事务。实体可以是人也可以是物，可以是具体的事物也可以是抽象概念。例如，职工、学生、课程、教师等都是实体。

3.4 实体-联系图



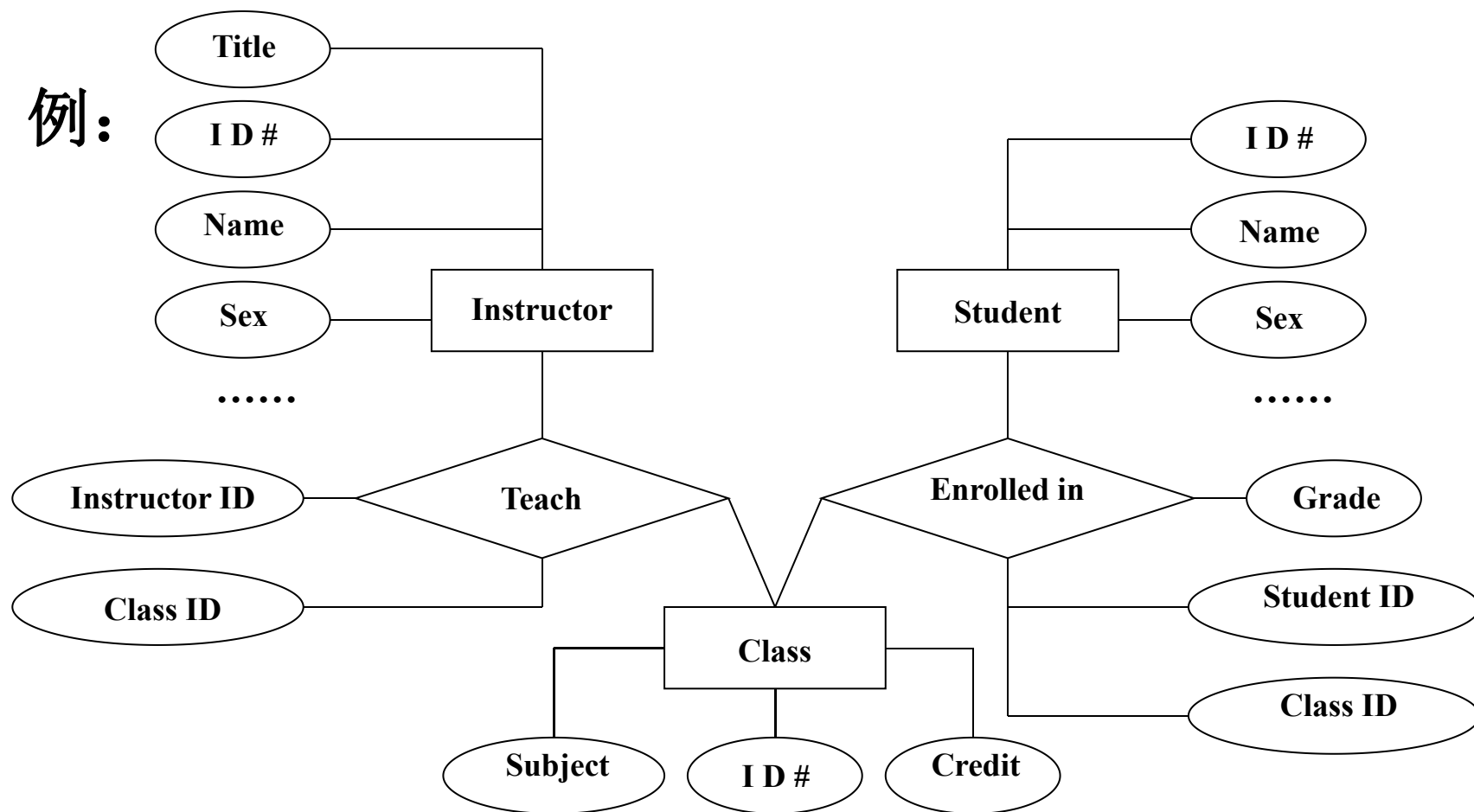
关系：客观世界中的事物彼此间往往是有联系的，例如，教师与课程间存在“教”这种联系。

3.4 实体-联系图

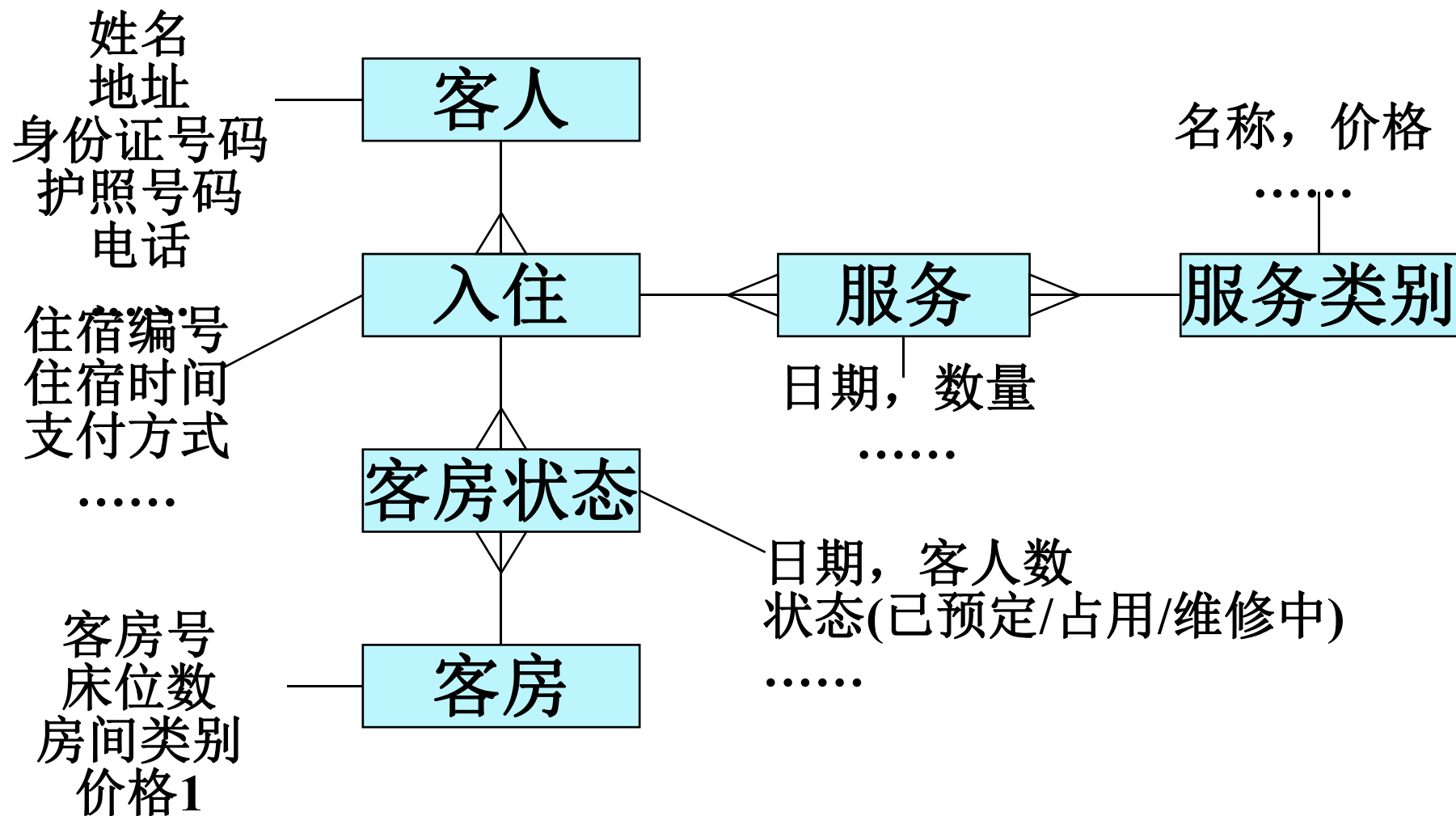
(3) Attributes  例: , 

- ▶ **属性**是实体或联系所具有的性质。通常一个实体由若干个属性来刻画。
- ▶ 例如, “学生”实体有学号、姓名、性别、系、年级

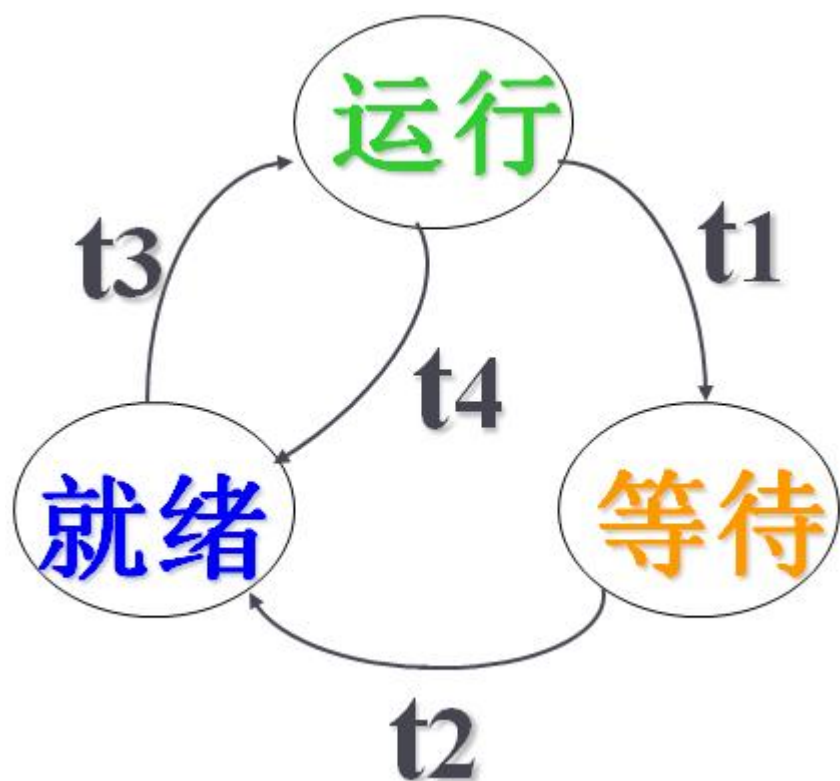
3.4 实体-联系图



实体-联系图实例：简单的酒店系统数据模型



3.5 状态迁移图



状态 事件	运行	就绪	等待
t1	等待		
t2			就绪
t3		运行	
t4	就绪		

进程的状态迁移图和状态迁移表

3.6 其他图形工具 - 判定表

- ▶ 在某些数据处理中，某数据流图的加工需要依赖于多个逻辑条件的取值，就是说完成这一加工的一组动作是由于某一组条件取值的组合而引发的。这时使用判定表来描述比较合适。
- ▶ 一张判定表通常由四部分组成，
左上部列出的是所有的条件，左下部为所有可能的操作，
右上部分表示各种条件组合的一个矩阵，右下部分是对应于每种条件组合应有的操作。

判断表	I 条件类别	II 条件组合
	III 操作	IV 操作执行

判定表例子1

考试总分	>=620	>=620	<620
单科成绩	有满分	有不及格	有满分
发升级通知书	y	y	n
发留级通知书	n	n	y
发重修通知书	n	y	n

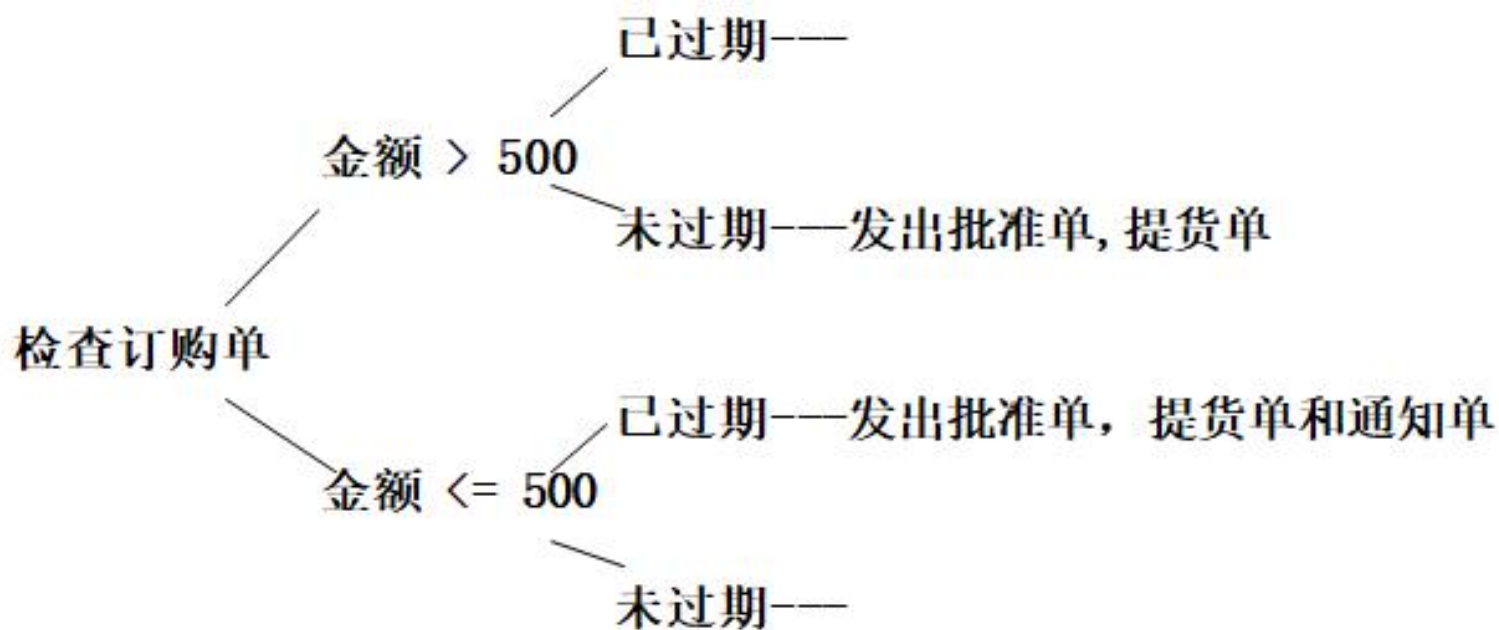
判定表例子2

大分类有无状况	小分类有无状况	对应的数据结构		
		大分类数组	小分类指针数组	输入数据指针数组
×	×	×	×	○
○	×	○	×	○
×	○	×	×	×
○	○	○	○	○

×：无 ○：有

3.6 其他图形工具 - 判定树

判定树是判定表的变种，它也能清晰地表达复杂的条件组合与所对应的操作之间的关系。判定树的优点在于它无须任何说明，一眼就能看出其含义，易于理解和使用。



加工逻辑描述工具的选择：

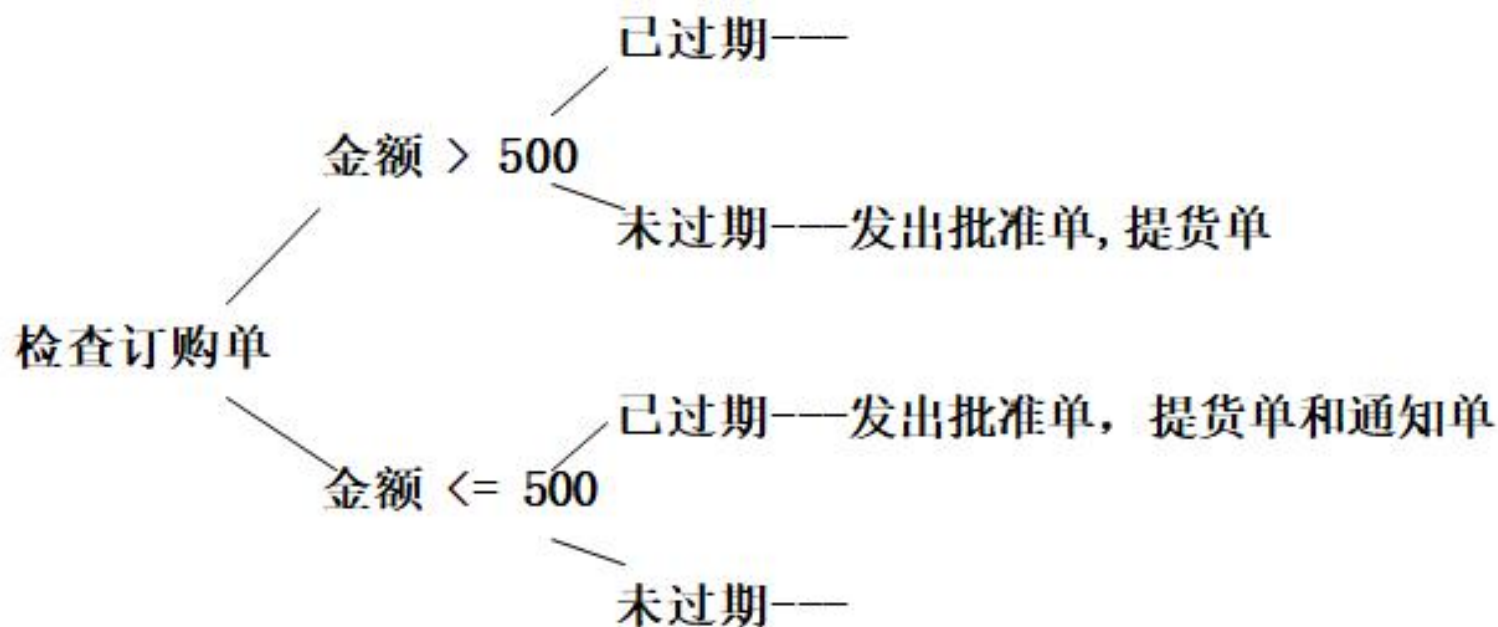
a. 不太复杂的判断逻辑，使用判断树比较好；

b. 复杂的判断逻辑，使用判断表比较好；

c. 若一个处理逻辑既包含了一般的顺序执行动作，又包含了判断或循环逻辑，则使用结构化语言比较好。

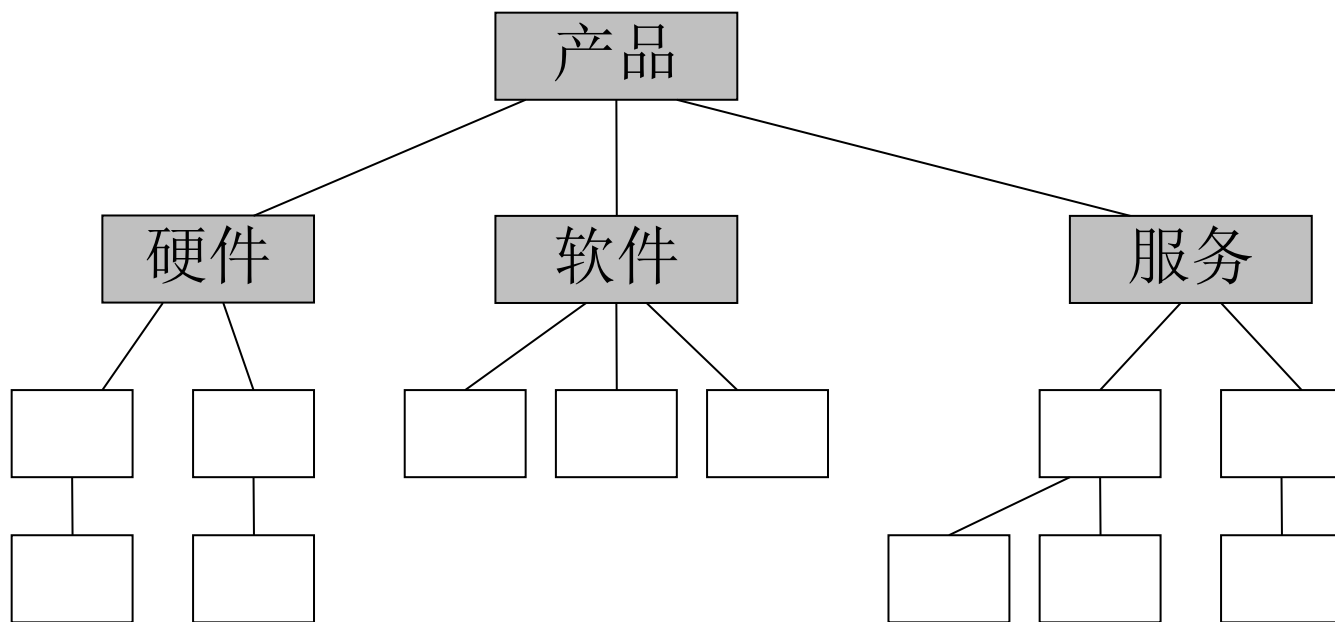
第三次作业

- ▶ 将以下判定树转换为对应的判定表。



3.6 其他图形工具

层次方框图——用数形结构的一系列多层次的矩形框描绘数据的层次结构。



3.6 其他图形工具

■ Warnier图

表明信息的逻辑组织。

■ IPO图

输入/处理/输出图

3.7 验证软件需求

需求分析阶段的其它工作：

- 确定设计限制
- 确定验收准则
- 编写“初步用户手册”
- 复查需求说明书

3.7.1 从哪些方面验证软件需求的正确性

- 👉 完整性（不要遗漏）。
- 👉 正确性（明确用户动机、正确的表达）
- 👉 一致性（不矛盾、表述统一）。
- 👉 现实性（不超出能力、符合环境限制）。
- 👉 可测试性（内容明确）

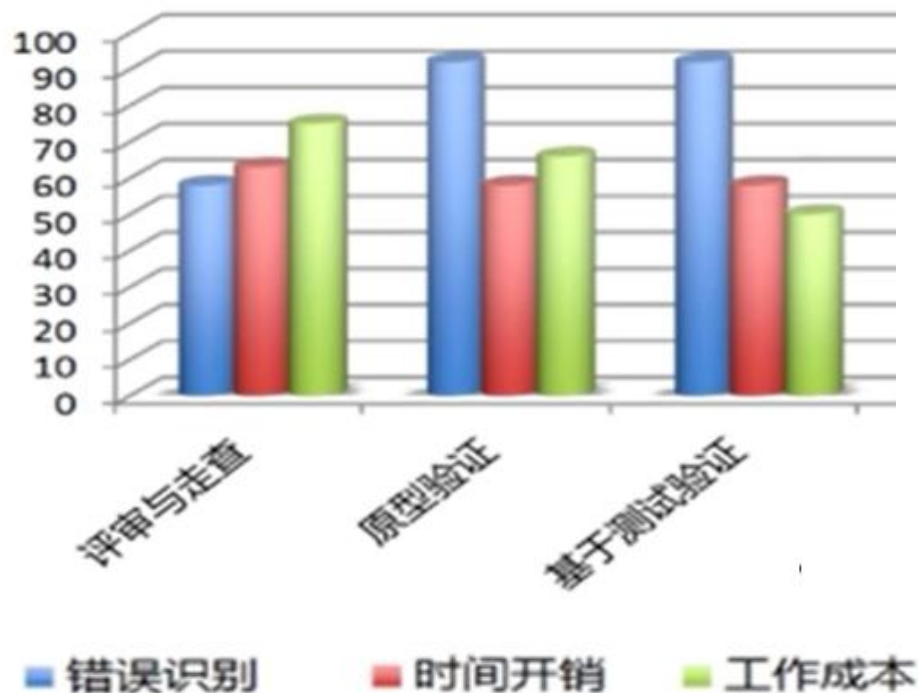
完整 一致 精确 无二义 简明易懂易于维护
错误发生的越早则纠正代价越小

验证软件需求的方法

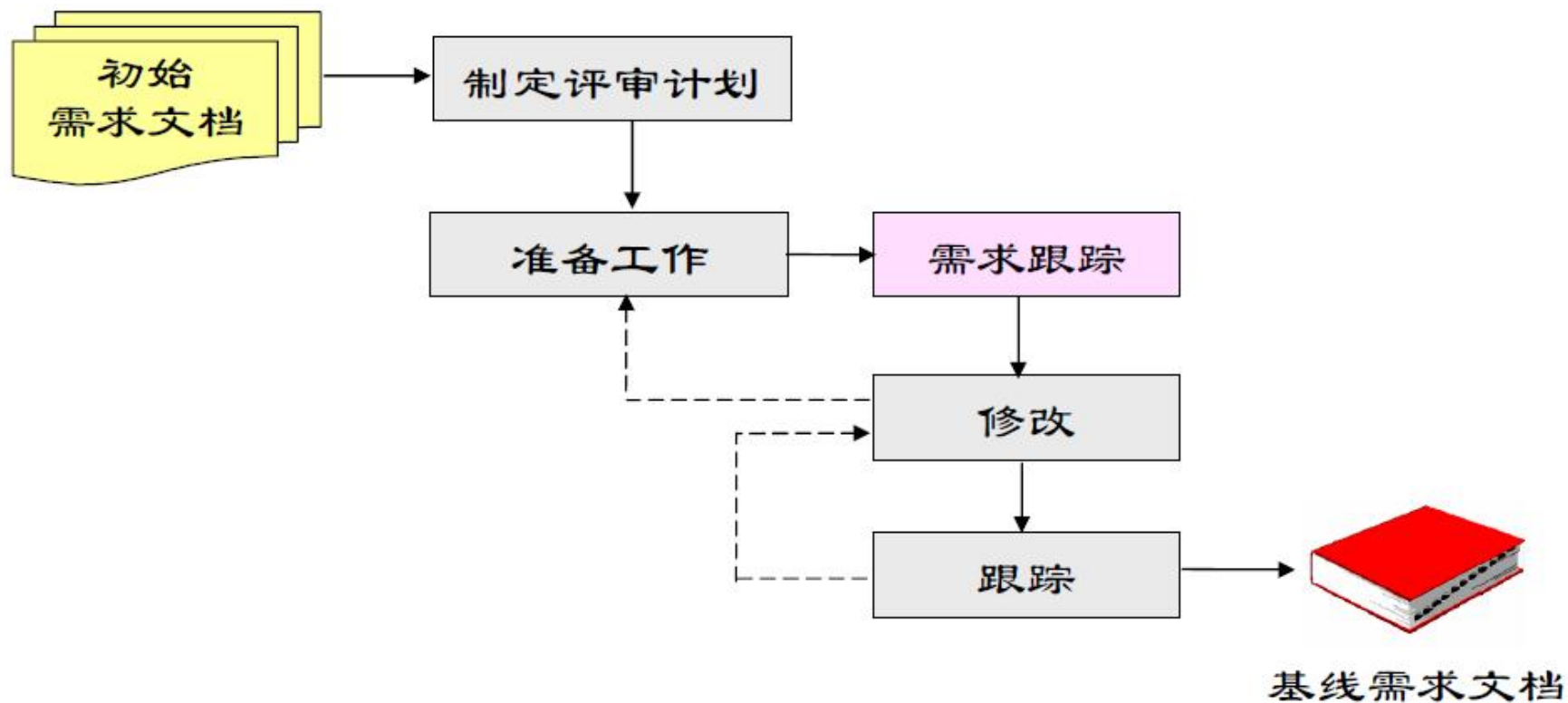
- **需求验证**是检验需求能否满足客户的意愿。
- 需求验证的技术
 - **需求评审**：由不同代表（如分析员、客户、设计人员、测试人员）组成的评审小组以会议形式对需求进行系统性分析。
 - **原型评价**：客户和用户在一个可运行的系统模型上实际检验系统是否符合他们的真正需要。
 - **测试用例生成**：通过设计具体的测试方法，发现需求中的许多问题。
- 需求验证主要围绕需求规格说明的质量特性展开。

需求验证技术的比较

- ▶ 评审/走查、测试是软件质量保证的重要方法。
- ▶ 原型法在必要时采用原型法进行验证，
- ▶ 3种方法同时作为质量保证-验证需求的方法，互为补充。



需求评审过程



软件需求规格说明书的质量特性（1）

- 正确性

- 需求规格说明对系统功能、行为、性能等的描述必须与用户的期望相吻合，代表了用户的真正需求。

- 审查需求的正确性应该考虑的问题

- 用户参与需求过程的程度如何？
- 每一个需求描述是否准确地反映了用户的需要？
- 系统用户是否已经认真考虑了每一项描述？
- 需求可以追溯到来源吗？

- 举例：下面的需求描述正确吗？

- 在用户每次存钱的时候系统将进行信用检查。

软件需求规格说明书的质量特性（2）

- 无二义性

- 需求规格说明中的描述对于所有人都只能有一种明确统一的解释。

- 审查需求的无二义性应该考虑的问题

- 需求规格说明是否有术语词汇表？
- 具有多重含义或未知含义的术语是否已经定义？
- 需求描述是否可量化和可验证？
- 每一项需求都有测试准则吗？

- 举例：下面的需求描述是无歧义的吗？

- 如果用户试图透支，系统将采取适当的行动。

软件需求规格说明书的质量特性（3）

- 完整性

- 需求规格说明应该包括软件要完成的全部任务，不能遗漏任何必要的需求信息。

- 审查需求的完整性应该考虑的问题

- 是否存在遗漏的功能或业务过程？
- 在每个定义的功能之间是否有接口？
- 是否有信息或消息在所定义的功能之间传递？
- 是否定义了功能的使用者？
- 是否已经清楚地定义了用户与功能之间的交互？
- 是否定义了与外部过程和系统之间的接口？

软件需求规格说明书的质量特性（4）

- 审查需求的完整性应该考虑的问题（续）
 - 所描述的功能是否可以映射到业务过程中？
 - 文档中是否存在待确定的需求引用？
 - 文档中是否存在未定义的术语和引用？
 - 文档的各个部分都完整吗？
 - 需求包括非功能属性的说明吗？
 - 是否考虑了软件性能？
 - 是否考虑了安全性要求？
 - 是否考虑了可靠性？
 - 是否考虑了系统容量问题？

软件需求规格说明书的质量特性（5）

- 可验证性

- 需求规格说明中描述的需求都可以运用一些可行的手段对其进行验证和确认。

- 审查需求的可验证性应该考虑的问题

- 在需求文档中是否存在不可验证的陈述，诸如“用户界面友好”、“容易”、“简单”、“快速”、“健壮”、“最新技术”等？
- 所有描述都是具体的和可测量的吗？

- 举例：下面的两个需求描述中哪一个难以验证？

- 系统将在 20 秒内响应所有有效的请求。
- 如果用户试图透支，系统将采取适当的行动。

软件需求规格说明书的质量特性（6）

- 一致性

- 需求规格说明对各种需求的描述不能存在矛盾，如术语使用冲突、功能和行为特性方面的矛盾以及时序上的不一致等。

- 审查需求的一致性应该考虑的问题

- 文档的组织形式是否易于一致？
- 不同功能的描述之间是否存在矛盾？
- 是否存在有矛盾的需求描述或术语？
- 文档中是否存在时序上的不一致？

- 举例：下面的两个需求描述是否有矛盾？

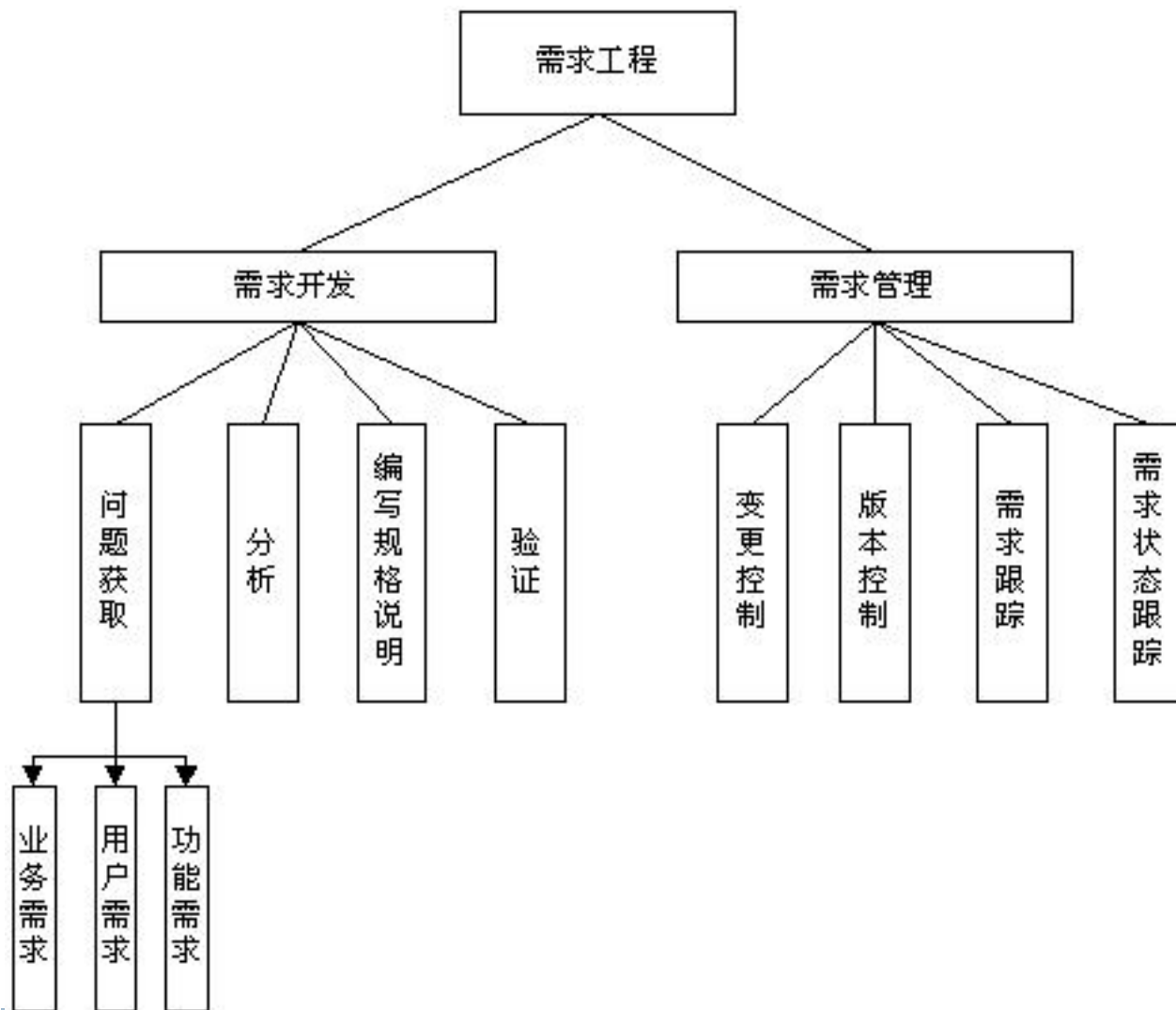
- 系统允许立即使用所存的资金。
- 只有在手工验证所存资金后，系统才能允许使用。

3.8 需求管理

- 需求管理是分析变更影响并控制变更的过程，主要包括变更控制、版本控制和需求跟踪等活动。



需求过程 的 workflow



需求跟踪管理表实例1

- ▶ **需求跟踪矩阵**（ Requirement tracking matrix ， RTM ）
是一种主要管理需求变更和验证需求是否得到了实现的有效工具，借助RTM，可以跟踪每个需求的状态。
- ▶ 是否设计了， 是否实现了， 是否测试了。

用例	功能需求	设计元素	代码	测试实例
UC1	Catalog.query.sort	Class catalog	Catalog.sort ()	test2
				test3
UC2	Catalog.update	Class catalog	Catalog.update ()	test10
				test11
				test12

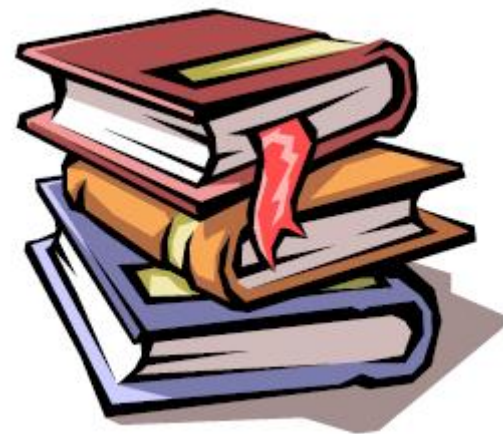
需求跟踪管理表实例2：流媒体服务计费系统

需求ID	需求名	优先级	需求关联	更改次数	状态	来源	SRS的参	概要设计	详细设计	设计变更次数	源程序文件	源程序文件变更次数	测试用例	测试用例变更次数	备注
1	从媒体日志文件采集数据	高		0	代码实现	功能性需求	3.1	4.1	SD4.1 UID2.2.11	变更1次 (SMBSC19)	Begin.jsp Began.jsp Collection.java SmbsTimerTask.java smbs.sql Closed.jsp	Build1.2 增加 Closed.jsp	SMBS-TC-FT-DC-001~ SMBS-TC-FT-DC-005		
2	构造组合查询，查看结果	高		0	代码实现	功能性需求	3.2	4.2	SD4.2 UID 2.2.2 UID2.2.10	变更1次 (SMBSC19)	CollectField.java ComboQuery.java DisplayQuery.jsp InputQuery.jsp NoDataFound.jsp ProcessQuery.jsp		SMBS-TC-FT-CQ-001 ~ SMBS-TC-FT-CQ-008		
3	构造交叉统计报表	高		0	代码实现	功能性需求	3.3	4.3	SD4.3 UID2.2.3 UID2.2.9	变更2次 SMBSC20 SMBSC28	CrossTabInput.jsp CrossTable.java DisplayCrsTab.jsp DisplayCrsXML.jsp	Build2.0 增加 DisplayCrsXML S.jsp	SMBS-TC-FT-CS-001~ SMBS-TC-FT-CS-008 SMBS-TC-FT-CS-009	SMBS-TC- FT-CS-009 (变更1次 —新增)	
4	计算媒体访问费用	高		0	代码实现	功能性需求	3.4	4.4	SD4.4 UID2.2.16	变更1次 (SMBSC21) 增加 UID2.2.16	computeCost.java Filelength.java Filesize.java Starttime.java countCost.java result.jsp		SMBS-TC-FT-VF-001~ SMBS-TC-FT-VF-003		
5	定义计费策略	高		0	代码实现	功能性需求	3.5	4.5	SD4.5 UID2.2.4 UID2.2.5	变更1次 (SMBSC22)	changeRule1.jsp CountRule.java CountRule1.java define.jsp RuleControl.jsp		SMBS-TC-FT-DF-001~ SMBS-TC-FT-DF-007		
6	用户月结算	高		0	代码实现	功能性需求	3.6	4.6	SD4.6 UID2.2.12	变更1次 (SMBSC23)	DisplayMonthPayment. sp MonthPayInput.jsp Payment.java		SMBS-TC-FT-MC-001~ SMBS-TC-FT-MC-004		
7	用户临时结算	高		0	代码实现	功能性需求	3.7	4.7	SD4.7 UID2.2.13		ProcessTemp.jsp TempPayInput.jsp		SMBS-TC-FT-TC-001~ SMBS-TC-FT-TC-003		

小结

根本任务：

确定满足用户需要的系统必须做什么。

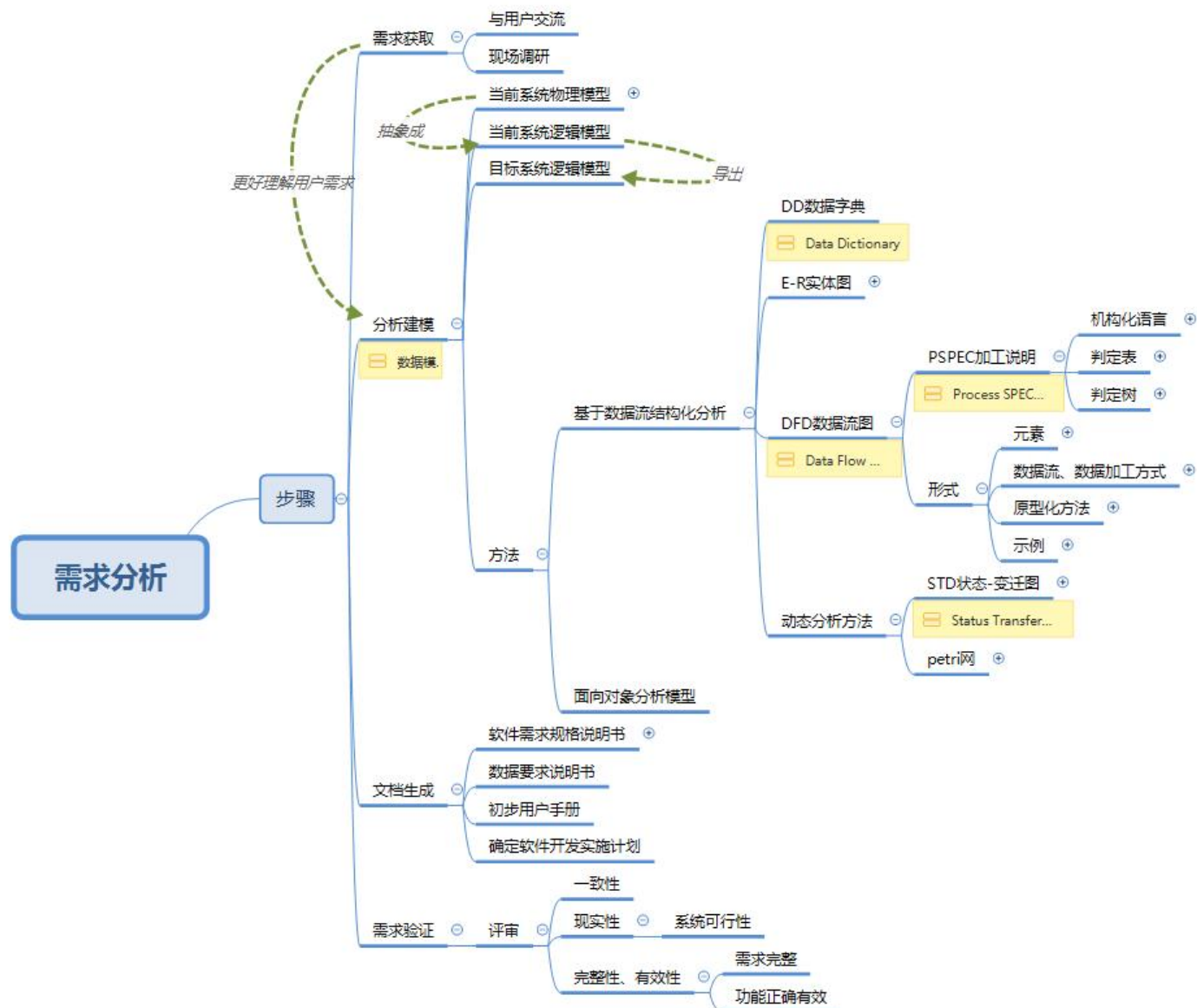


需求获取的方法、如何做需求分析；

系统逻辑图：

数据流图，ER图，数据字典，判定树、判定表

需求分析思维导图



<https://blog.csdn.net/Elsa15>

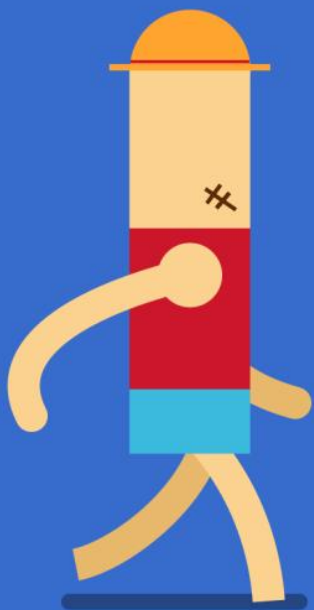
附录：需求工程师的能力要求

- ▶ 分析问题&解决问题的能力
- ▶ 人际沟通交流能力
- ▶ 软件工程知识和技能（开发经验）
- ▶ 应用领域有关知识（业务专业能力）
- ▶ 书面语言组织和表达能力
- ▶ 其他能力。。。

应避免的问题

- 用户需求=产品需求，不懂挖掘
- 看不起小需求，不注重细节
- 存在即合理，不敢质疑
- 只看重商业价值，不重视用户体验

参考链接：<https://www.aiyingli.com/43085.html>



补充练习

下列哪项需求描述属于业务需求描述？

- ☐ A 我们的任务是无缝集成有竞争力的软件信息服务来解决商业问题
- ☐ B 我们的目标是让客户将我们的品牌和质量联系在一起
- ☐ C 我们公司的主营业务是销售飞机票
- ☐ D 公司网站上销售的产品必须满足所有食品药品监管需求

提交

需求管理只适用于需求分析阶段。

- ☐ A 正确
- ☐ B 错误
- ☐ C 说不清楚
- ☐ D 都可以

提交

下面属于需求收集方法的有

- ☐ A 问卷
- ☐ B 现场观察
- ☐ C 客户面谈
- ☐ D 需求专题讨论会

提交

判断对错。

- ✗ ☒ A 非功能性需求就是质量属性需求。
- ✓ ☒ B 需求的确认必须由客户或客户方的代表认可。
- ✓ ☒ C 需求管理是对变化需求的导出、编档、组织和跟踪。
- ✗ ☒ D 客户需求只需要收集功能性需求，其他需求可以不考虑。

提交

END

谢谢！