



软件工程

第四章 需求工程

乔立民

qlm@hit.edu.cn

2011年4月16日

主要内容

- **4.1 系统工程**
 - **4.1.1 基于计算机的系统**
 - **4.1.2 系统建模**
- **4.2 软件需求**
- **4.3 需求工程**

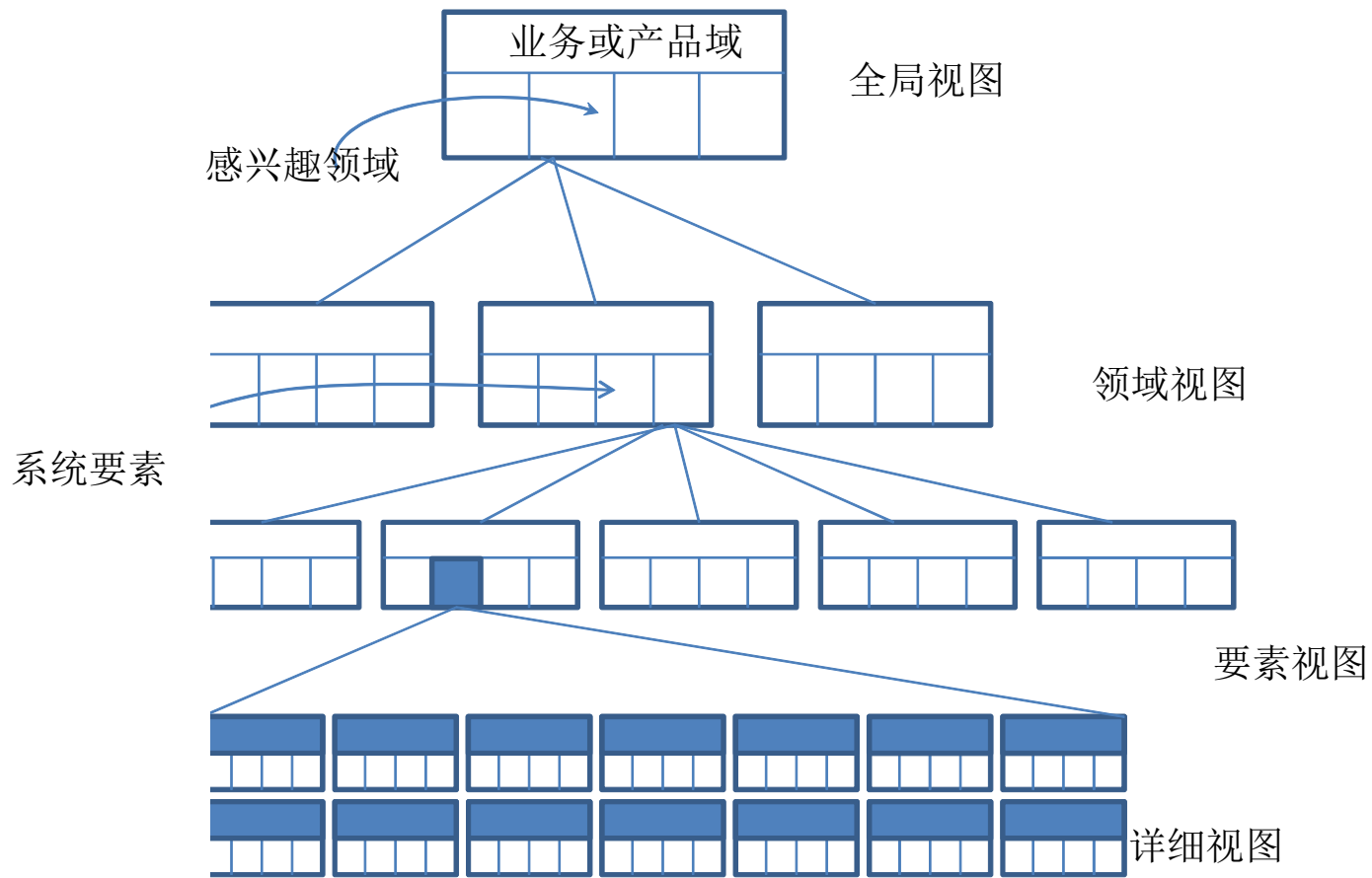
系统工程

- 系统工程是组织管理的技术。把极其复杂的研制对象称为**系统**，即由相互作用和相互依赖的若干组成部分结合成具有特定功能的有机整体，而且这个系统本身又是它所从属的一个更大系统的组成部分。……系统工程则是组织管理这种系统的规划、研究、设计、制造、试验和使用的科学方法，是一种对所有系统都具有普遍意义的科学方法。
- 业务过程系统工程：关注企业业务过程
- 产品系统工程：关注产品

基于计算机的系统

- 基于计算机的系统：组织在一起通过处理信息来实现预定目标的要素集合或排列
- 要素
 - 软件
 - 硬件
 - 人员
 - 数据库
 - 文档
 - 规程

系统工程层次结构



业务过程工程

■ 数据架构

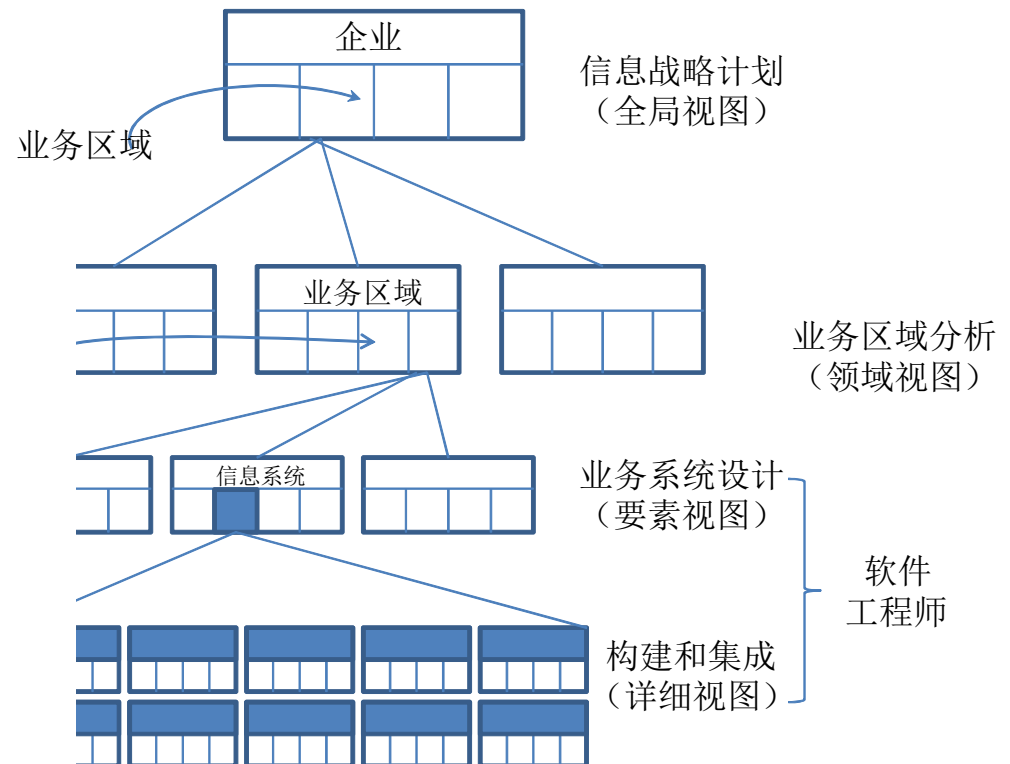
- 在业务活动中流动的数据对象通过数据库组织起来，为业务需要提供所需的信息

■ 应用架构

- 执行转换的程序（软件）系统

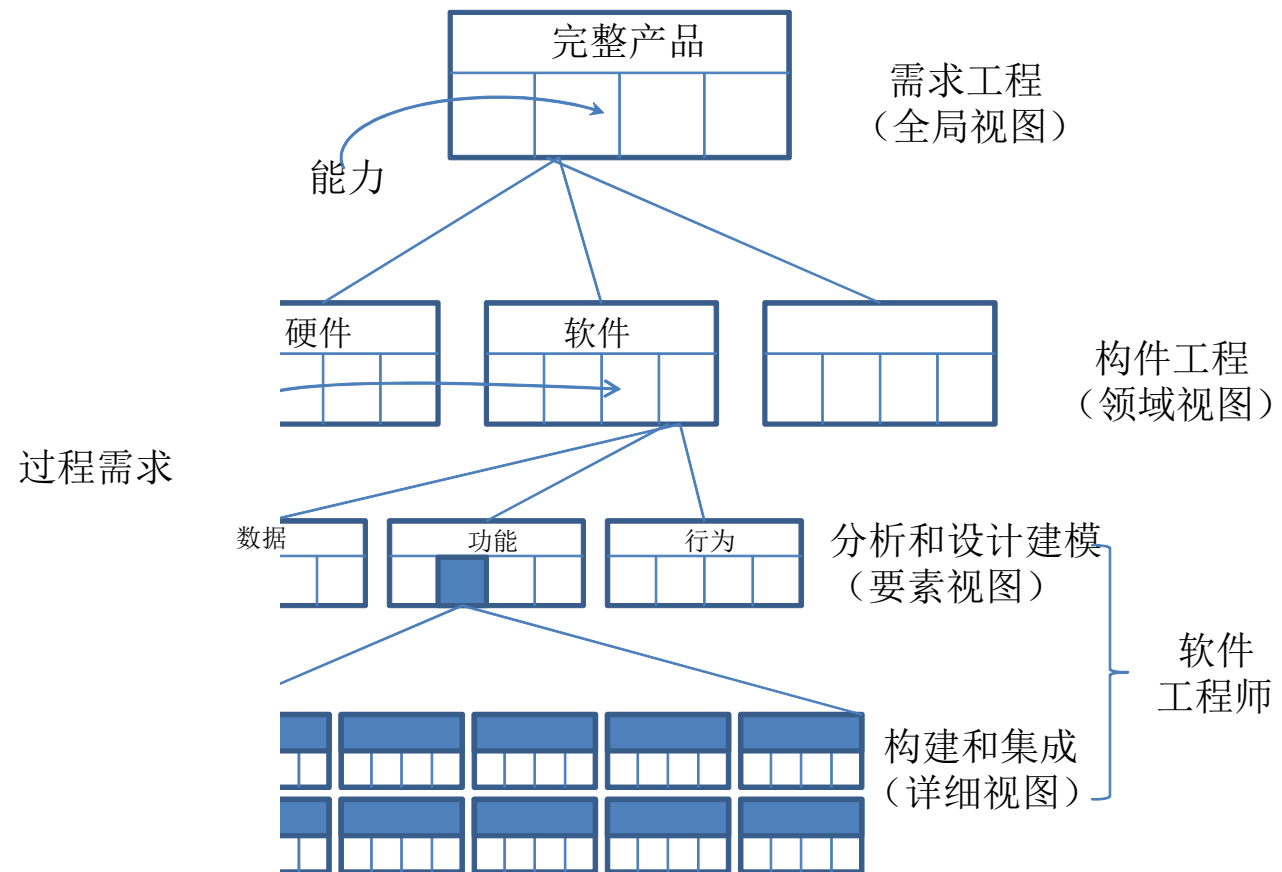
■ 技术基础设施

- 为数据架构和应用架构提供基础



产品工程

- 软件
- 硬件
- 数据（数据库）
- 人员



本章内容

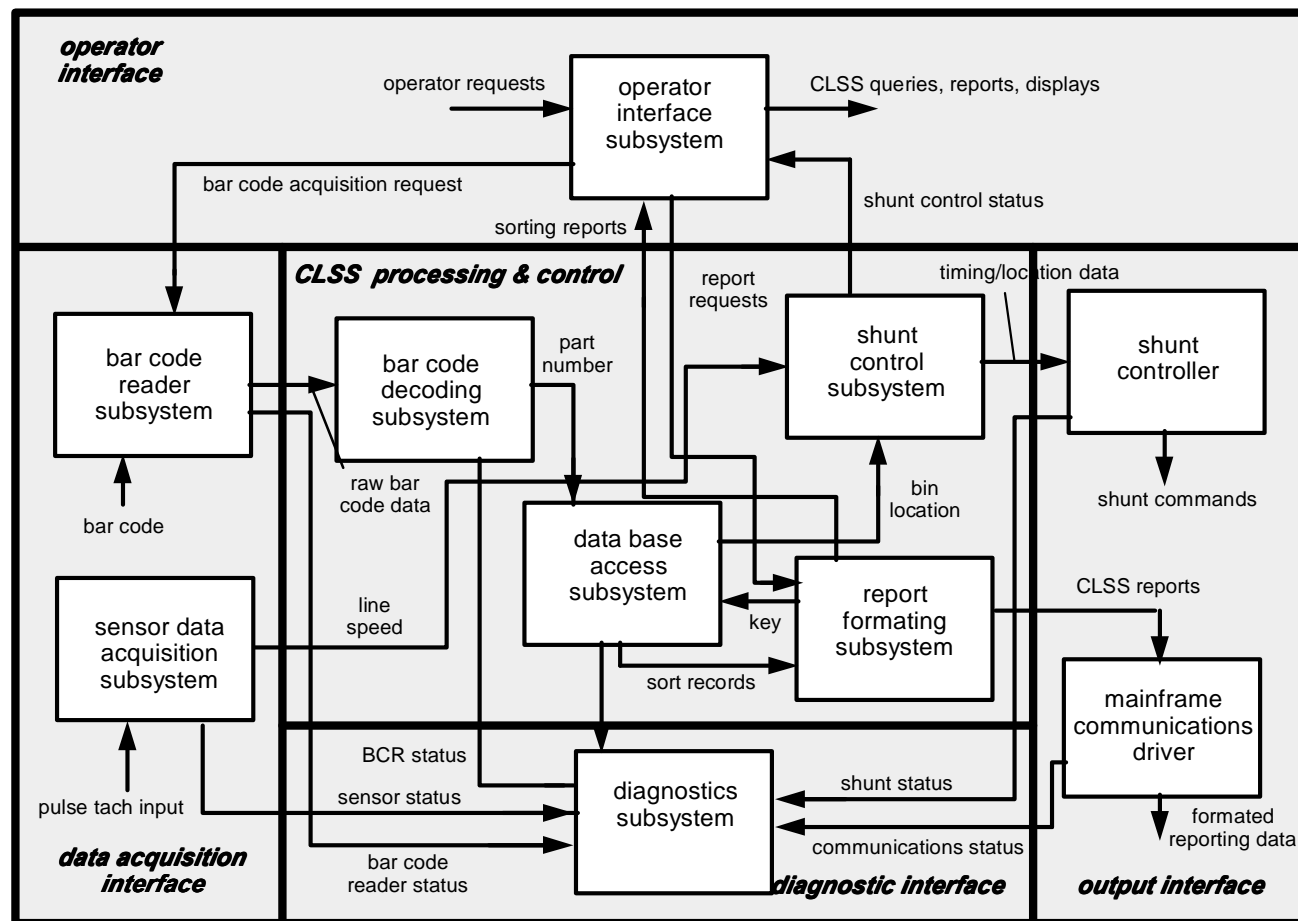
- **4.1 系统工程**
 - **4.1.1 基于计算机的系统**
 - **4.1.2 系统建模**
- **4.2 软件需求**
- **4.3 需求工程**

模型

- 模型是某个事物的抽象，其目的是在构建这个事物之前先来理解它。
- 用途：
 - 在构建物理实体之前先测试
 - 与客户交流
 - 可视化
 - 降低复杂度
- 分级或分层方式，不同视角

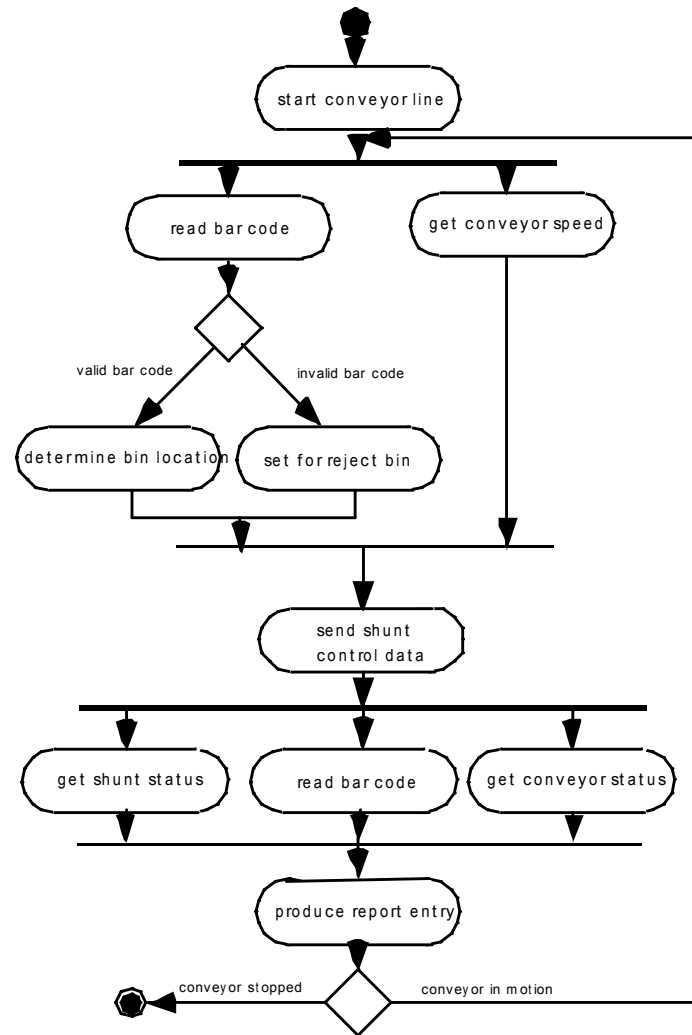
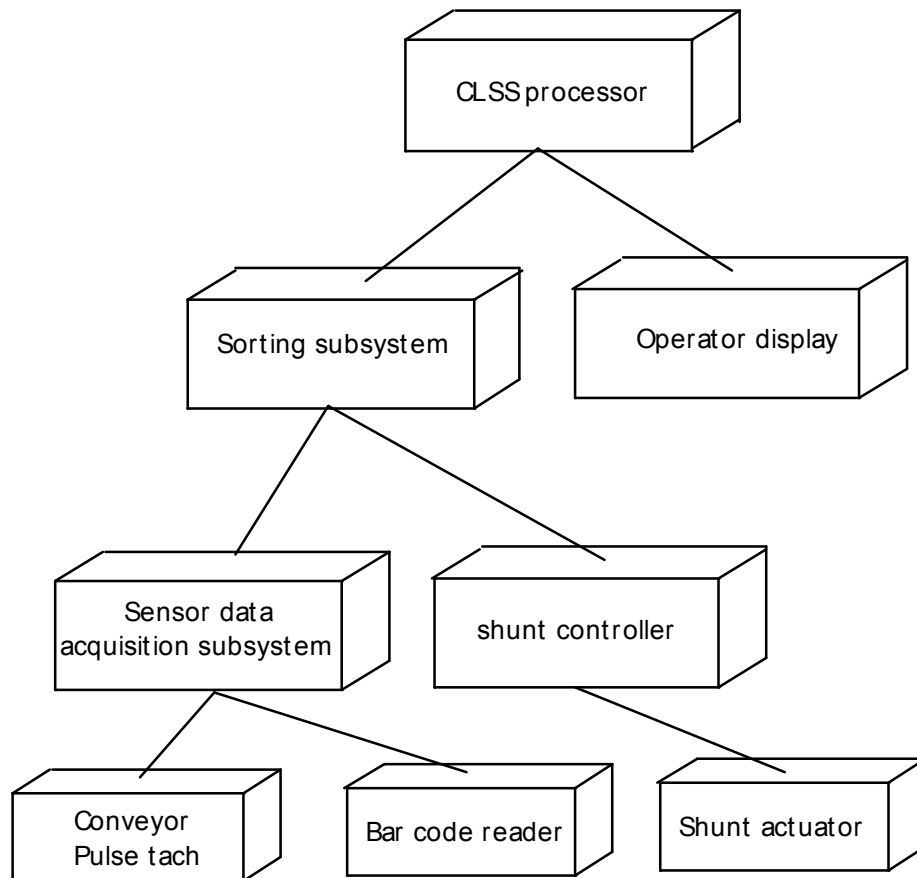
Hatley-Pirbhai建模

- 输入—处理—输出：基于流的建模

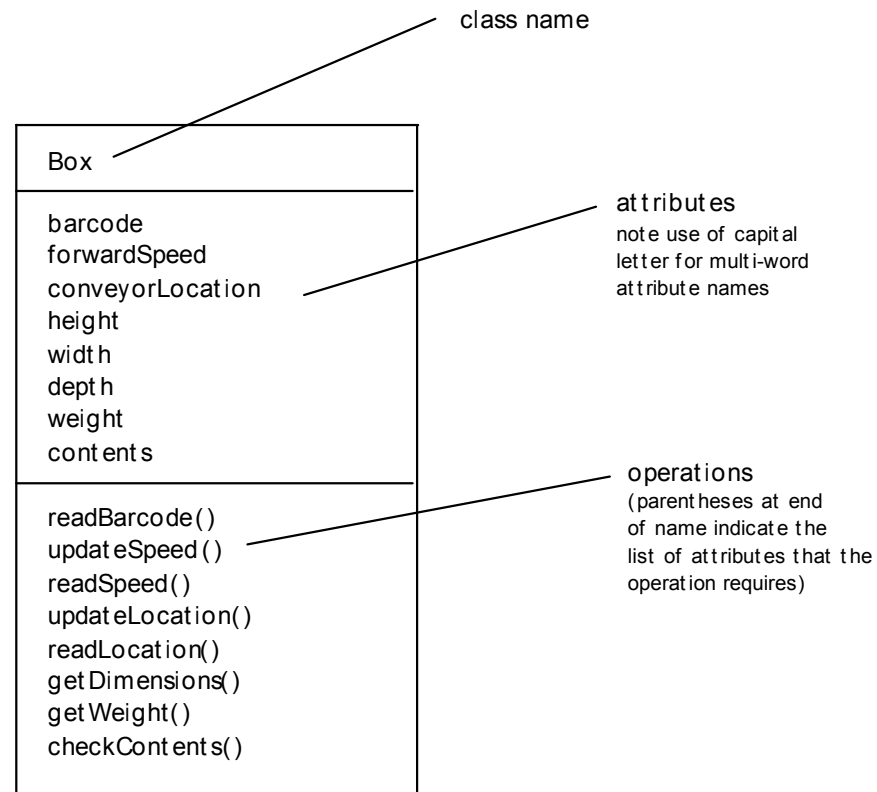


UML建模

基于对象的建模



UML建模



本章内容

- **4.1** 系统工程
- **4.2** 软件需求
 - **4.2.1** 需求定义
 - **4.2.2** 需求分类
- **4.3** 需求工程

软件开发的目标是什么？

- 质量要求：符合客户业务工作；
- 进度要求：在预定的时间完成
- 资金要求：在预算约束下完成；
-
- 软件开发要能够满足各方面的需求

[案例分析1] “身份证号码可以修改吗？”

- 小张开发了一套人力资源软件，有一天他接到了人力资源部李部长打来的电话...

李部长 赵卫兵想修改身份证号码，但系统不允许，能帮忙吗？

小张 为什么？

李部长 他的身份证号码和别人的重了，公安局给他发了新身份证。

小张 赵卫兵的以前数据需要保留吗？身份证号码是不能修改的，是每个人的系统标识，除非删除掉这个人，再以新员工增加进去，但是以前数据都一起删除了。

李部长 他的数据当然要保留，公安局已经把他的身份证号码更换了，尽快修改系统。

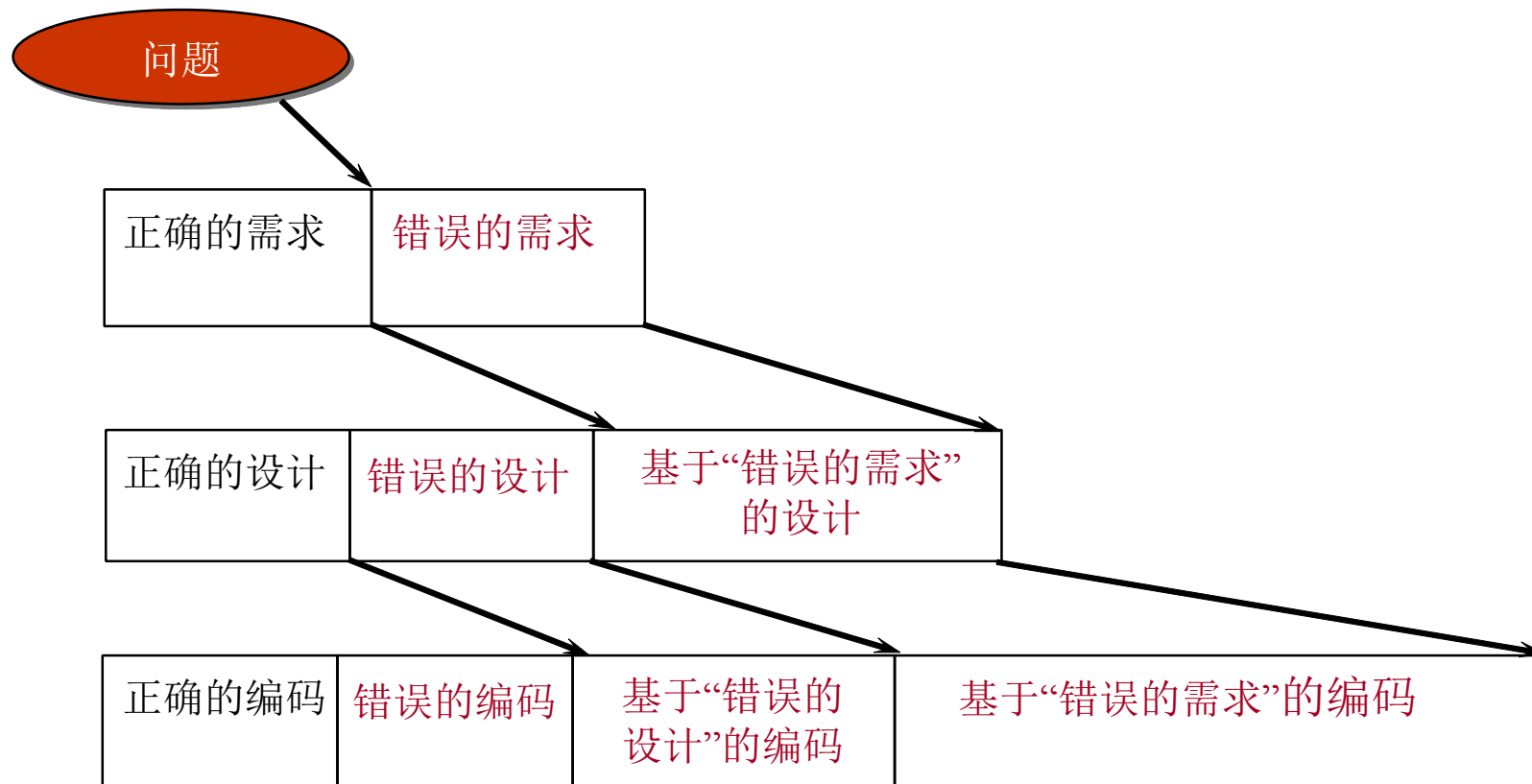
小张 这并不是我的错！在开发系统之前，你从来没有向我提起过有这种需求！

李部长 不管怎样，请尽快把这个功能修改完毕，否则赵卫兵将无法领取工资了。

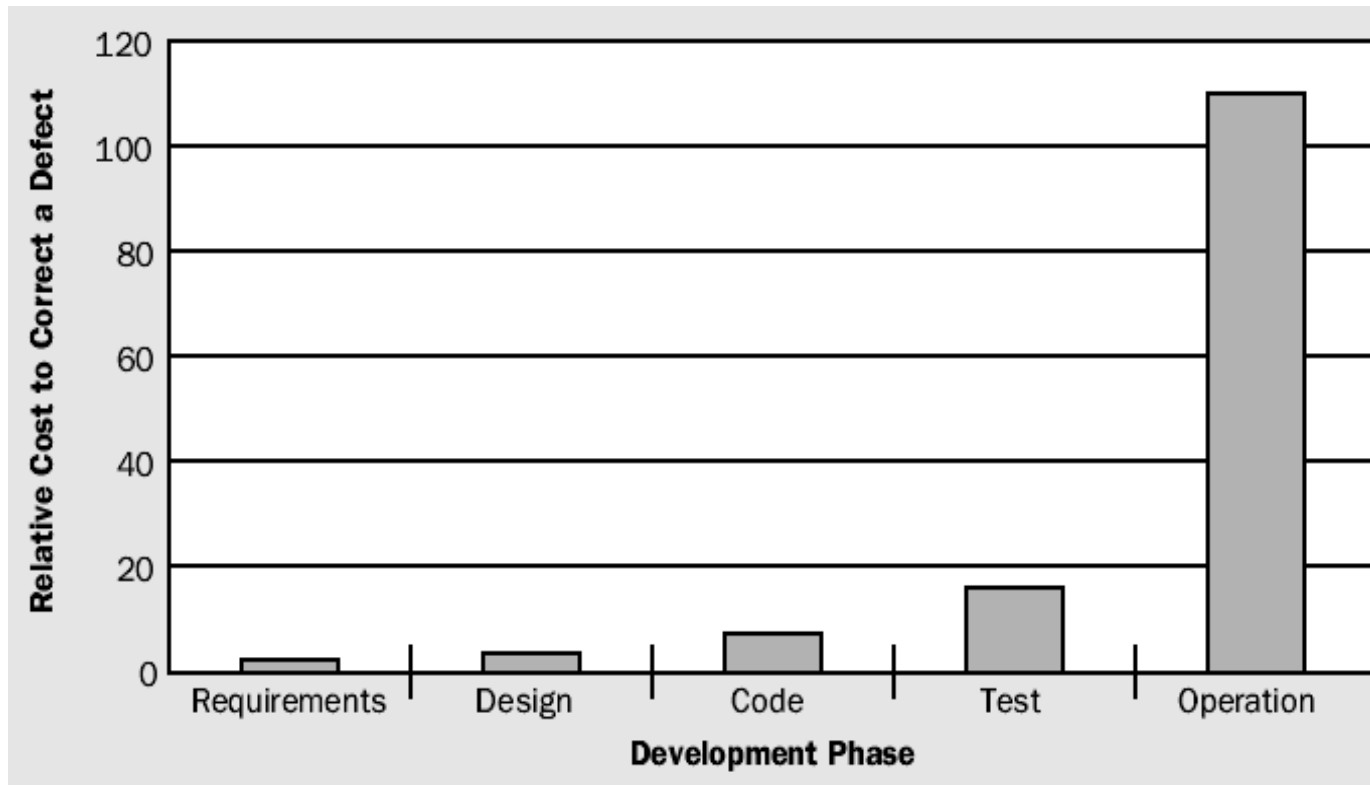
小张 如果你一开始就告诉我你想随时改变某人的身份证，那这些就都不会发生！

.....

“错误的需求”的扩散效应



“错误的需求”的修复代价



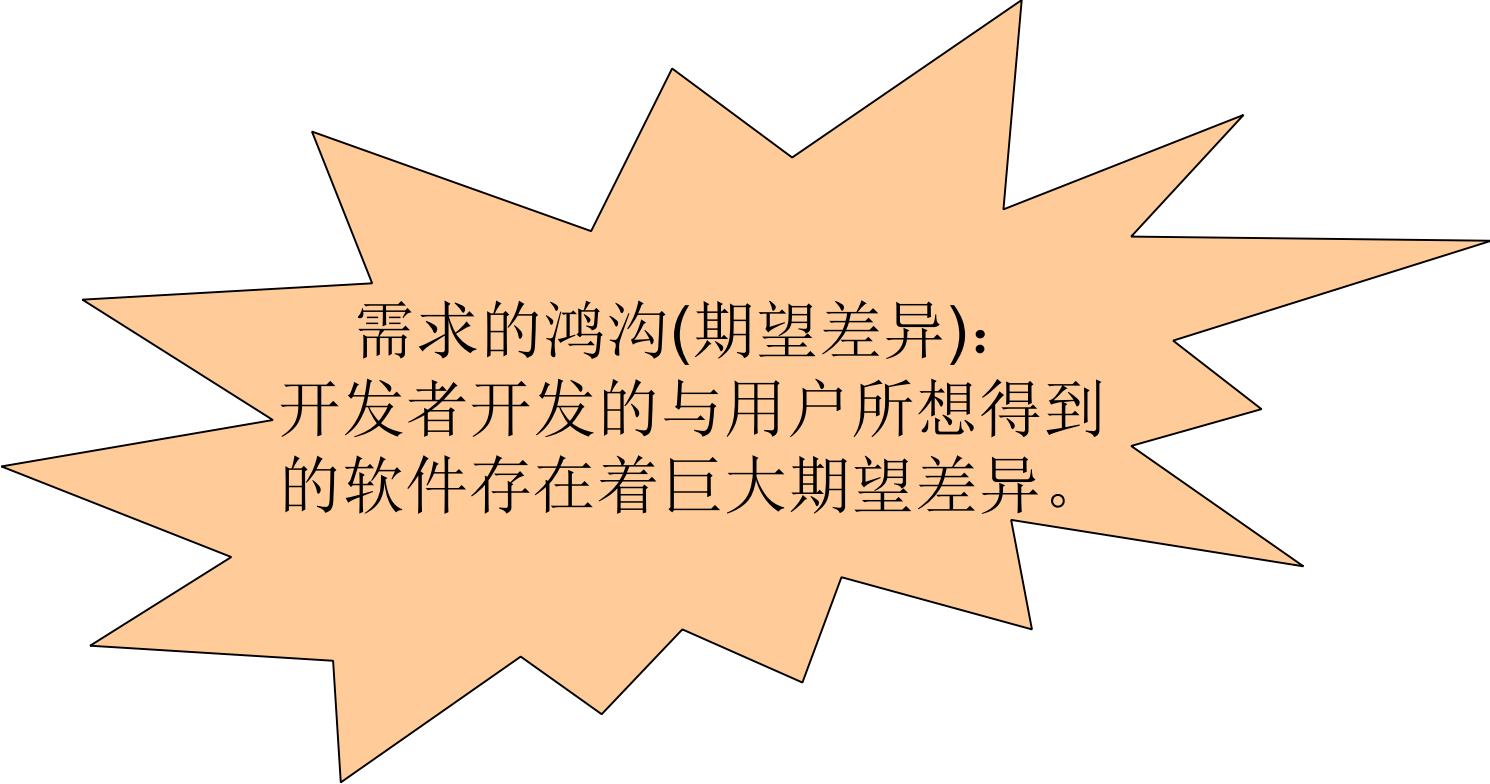
“构建一个软件系统最困难的部分是确定构建什么...在出错之后会严重影响随后实现的系统，并且在以后的修补是如此的困难...”

“错误的需求”所带来的后果

■ 早期的需求错误可能造成

- 重新进行规格说明、设计、编码和测试
- 改变订单：告诉用户和操作员用一个修正后的版本来代替有缺陷的版本
- 纠正活动：消除由于不正确的系统错误造成的一切危害，可能涉及到赔偿客户损失以及重新运行系统等
- 报废：即使设计、代码和测试完成得很好，由于它们是根据不正确的需求产生的，所以不得被丢弃
- 收回有缺陷的软件产品以及相关的用户手册
- 技术人员为客户重新安装新软件所必须支付的服务成本

根本原因是什么？



需求的鸿沟(期望差异):
开发者开发的与用户所想得到的
软件存在着巨大期望差异。

什么是“软件需求”

- 软件需求(**Software Requirements**):
 - 用户解决问题以达到特定目标所需的能力;
 - 系统或系统构件要满足的合同、标准、规范或其他正式文档所需具备的能力; ——IEEE, 1997
- 软件需求: 以一种清晰、简洁、一致且无二义性的方式, 描述用户对目标软件系统在功能、行为、性能、设计约束等方面的期望, 是在开发过程中对系统的约束。
- 需求通常用于表达“做什么”, 而不描述“如何做”。



“软件需求”的作用

■ “Requirement is the Basics of Quality”

- 充分理解现实中的业务问题，并作为软件设计的基础；
- 为软件项目的成本、时间、风险估计提供准确的依据；
- 减少开发工作量，避免将时间与资源浪费在设计与实现错误的需求上；
- 通过提供需求文档和需求基线，来有效的管理系统演化与变更；
- 作为顾客与开发团队之间正式合同的一部分；
- 为最终的验收测试提供标准和依据；

视角不同，需求各有不同

开发人员

开发经理

销售人员

顾客

终端用户

维护人员

系统管理员

易用性

易调试性

可修改性

可测试性

结构清晰性

功能性

价格

开发成本

按时交付

性能

稳定性与可维护性

易安装性

易集成性

不同的角色，他们所关心的非功能需求都有哪些？

不同角色的关注点之间，是否有重叠的情况？

不同角色的关注点之间，是否有冲突的情况？

关于“需求”的例子

■ **Course Registration System(学生选课系统)**

- 某大学希望采用计算机管理学生的选课
- 学生可以在一个学期开始之前选择该学期开设的某些课程
- 老师可以使用选课系统获得选课学生的名单，并登记学生的课程学习成绩
- 学生不希望自己的学习成绩被他人查阅
-(你可以补充吗？)

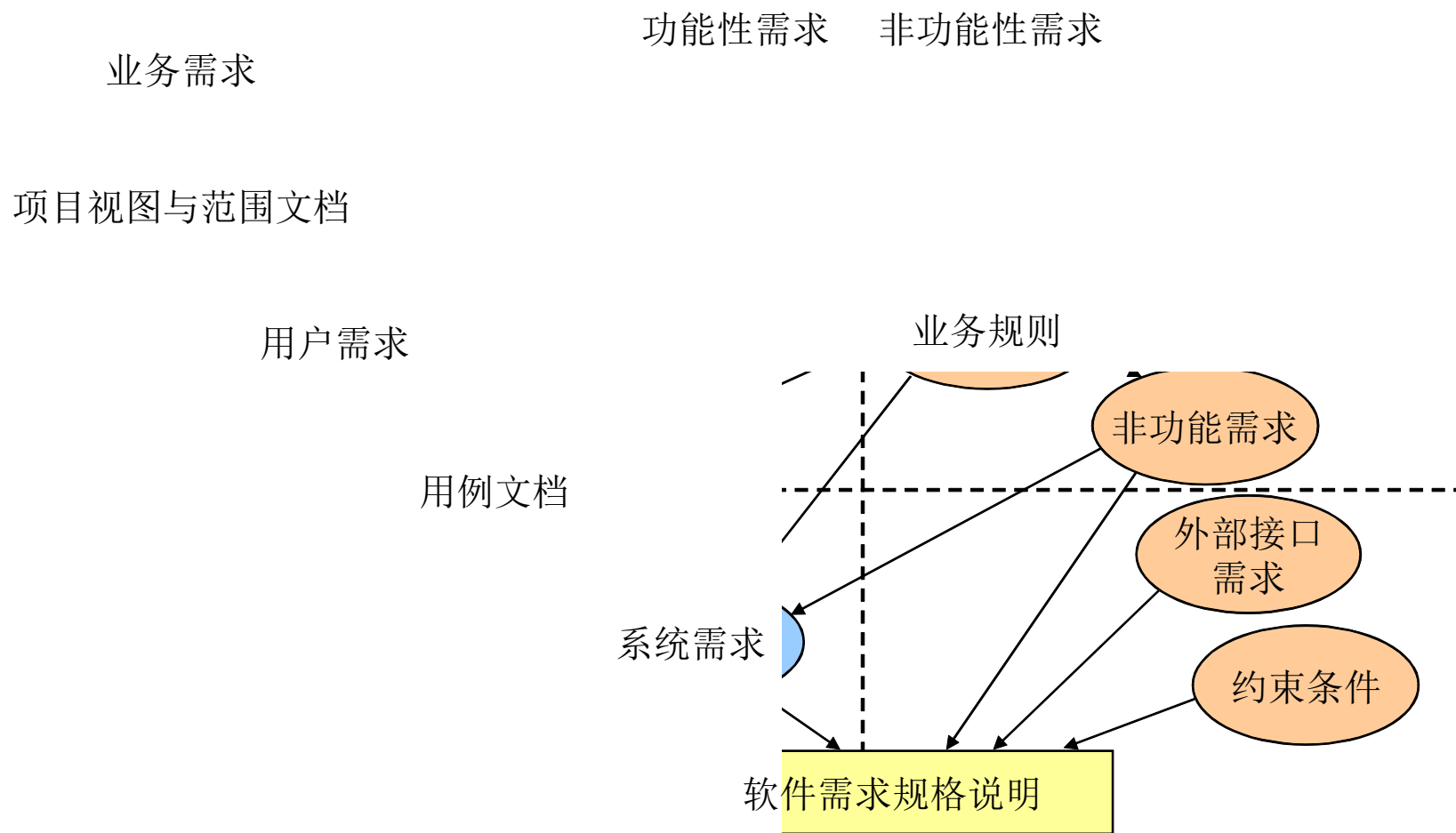
■ 以下描述是否属于需求？为什么？

- 系统通过JDBC与Oracle数据库CourseDB建立连接，并使用T-SQL语句从CourseOffering数据表中获得课程的开设信息。

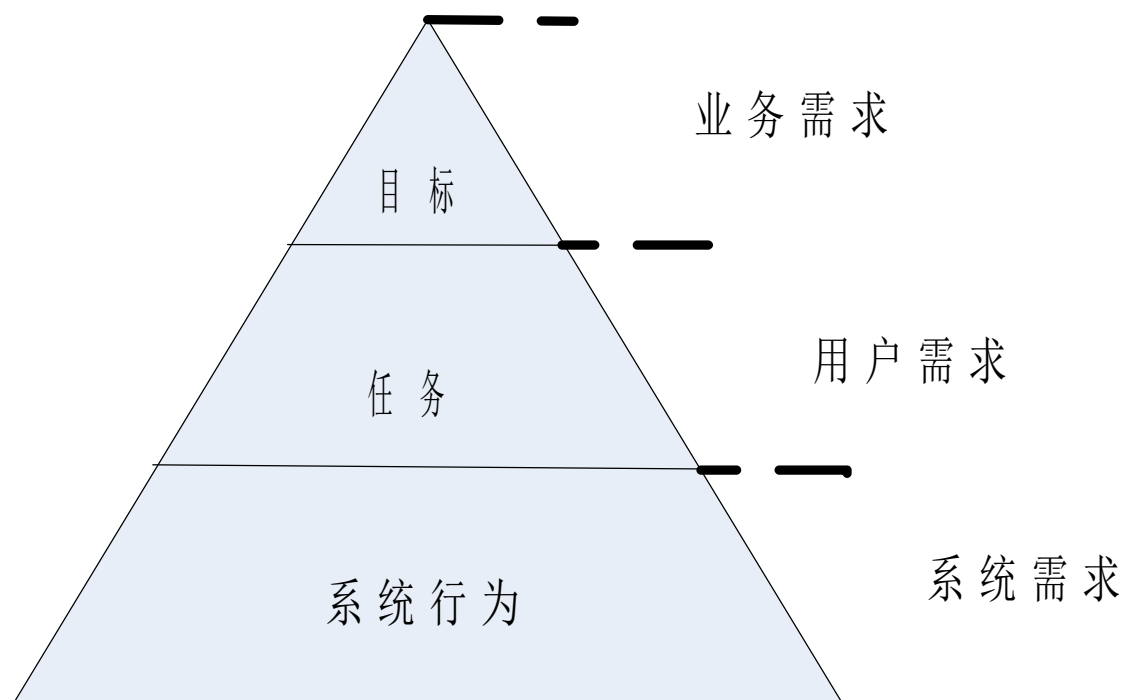
本章内容

- **4.1 系统工程**
- **4.2 软件需求**
 - **4.2.1 需求定义**
 - **4.2.2 需求分类**
- **4.3 需求工程**

不同层次的软件需求



功能需求的层次性(F)



1 业务需求

- **业务需求(Business Requirements):** 客户对于系统的高层次目标要求(**high-level objectives**)，定义了项目的远景和范畴(**vision and scope**)

- 业务：属于哪类业务范畴？应完成什么功能？为何目的？
- 客户：软件为谁服务？目标客户是谁？
- 特性：区别于其他竞争产品的特性是什么？
- 价值：价值体现在哪些方面？
- 优先级：功能特性的优先级次序是什么？

- **[例]“图书资料管理系统”的业务需求**

- 该系统使用计算机实现图书资料的日常管理，提高工作效率和服务质量；
- 该系统可让用户在网络上查询与浏览电子资料，改变原有的借阅模式；
- 由于版权的限制，某些电子资料只能浏览/打印，但不能下载。

2 用户需求

- **用户需求(User Requirements):** 从用户角度描述的系统功能需求与非功能需求，通常只涉及系统的外部行为而不涉及内部特性。
- **[例]**用户可以通过Internet随时查询图书信息和个人借阅情况，并可以快速查找和浏览需要的电子资料；
 - [功能需求]用户通过Internet查询图书信息；
 - [功能需求]用户通过Internet浏览个人借阅情况；
 - [功能需求]用户通过Internet查找和浏览电子资料；
 - [非功能需求]随时、快速

业务需求与用户需求的对比

■ 针对**Course Registration System**

■ 业务需求

- 由于实行学分制管理，学校领导希望用计算机管理学生选课。
- 课程信息维护、选课管理、课程成绩登记和查询等业务全部由手工方式改为计算机应用。

■ 用户需求

- 教务管理员希望能够增加、修改和删除学校的课程目录，并且设置各学期课程的开设信息。
- 学生希望能够在学期开始之前查询所有开设课程的详细信息，并能够通过校园网进行选课。
- 学生希望在选课期间系统能够**24**小时使用，系统使用方便快捷。

3 系统需求

- **系统需求(System Requirements, SR):** 系统应该提供的功能或服务，通常涉及用户或外部系统与该系统之间的交互，不考虑系统内部的实现细节；

- **【例】**
 - 用户可从图书资料库中查询或者选择其中一个子集；
 - 系统可提供适当的浏览器供用户阅读馆藏文献；
 - 用户每次借阅图书应对应一个唯一的标识号，它被记录到用户的账户上；

功能需求的层次性(F)

■ 将用户需求转化为系统需求的过程是一个复杂的过程

- 首先需要分析问题领域及其特性，从中发现问题域和计算机系统的共享知识，建立系统的知识模型；
- 然后将用户需求部署到系统模型当中，即定义系列的系统行为，让它们联合起来实现用户需求，每一个系统行为即为一个系统需求。
- 该过程就是需求工程当中最为重要的需求分析活动，又称建模与分析活动。

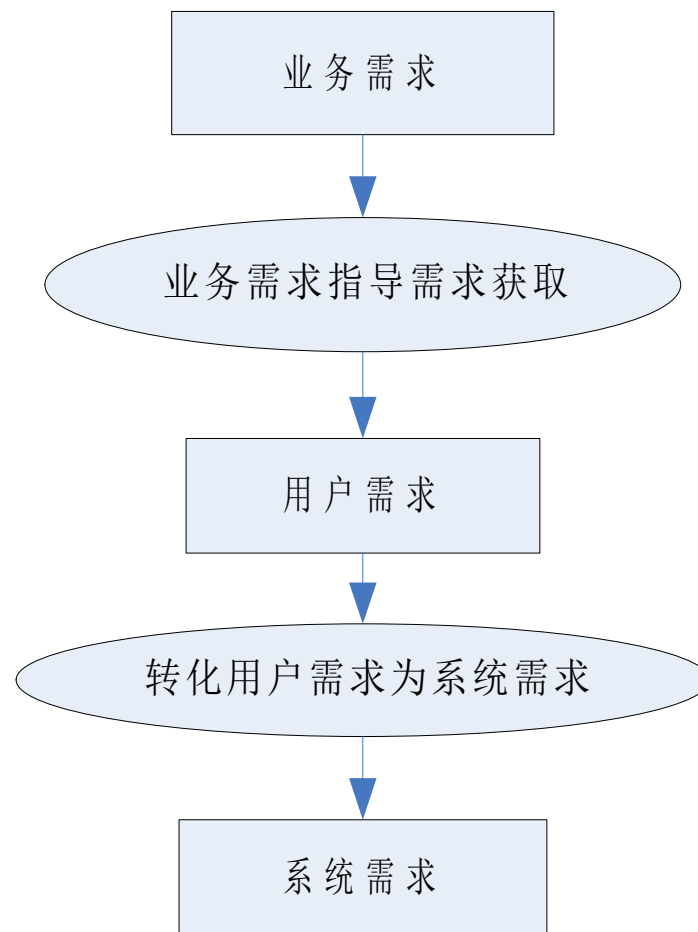


图 不同抽象层次需求之间的联系

系统需求的类型（FURPS+）

- 功能性需求(**Functional Requirements**): 描述了系统与其独立于系统实现环境之间的交互。
- 环境包括用户和任何其他与该系统进行交互的外部系统。

需求的类型（FURPS+）

- 非功能性需求(**URPS**): 描述了不直接关联到系统功能行为的系统的方方面面。
 - 可用性(**Usability**): 是一种用户可以学会的操作、输入准备、解释一个系统或者构件输出的状况。
 - 可靠性(**Reliability**): 是系统或构件在给定时间内、指定条件下, 完成其要求功能的能力。
 - 性能(**Performance**): 需求要考虑系统的定量属性, 比如响应时间, 吞吐量、有效性和准确性。
 - 可支持性(**Supportability**): 需求关注于在进行部署后系统的变化状况, 比如包括可适配性、可维护性、可移植性等。

非功能需求(URPS+)

- **非功能需求(Non-Functional Requirements, NFR):** 从各个角度对系统的约束和限制，反映了客户对软件系统**质量和性能(quality and performance)**的额外要求，如响应时间、数据精度、可靠性等。
- **[例]**
 - 系统在**20秒**内响应所有的请求；
 - 系统应该每周**7天**、每天**24小时**都可使用；
 - 对一个没有经验的用户而言，经过**2小时**培训即可使用系统的所有功能。
- **注意：**非功能需求隐含了对可选设计方案的一些关键影响
 - 体系结构设计(e.g., 体系结构风格选择)
 - 算法设计(e.g., 排序策略的选择)

其他需求(+)

- 实现需求：资源限制、语言和工具、硬件等
- 界面需求：人机交互
- 操作需求：对其操作设置的系统管理
- 打包需求：发布形式
- 合法需求：应用其他软件的许可等

非功能需求的度量

- **NFR:** 检验起来非常困难，一般采用一些可度量的特性进行描述。
- 例如：
 - 即使对一个没有经验的用户，系统也应该很容易使用，且是用户错误降到最少；
- 修改为：
 - 对一个没有经验的用户来说，经过**2**个小时的培训就应该使用系统的全部功能。在这样的培训之后，一个有经验的用户每天的出错平均数不应超过**2**次。

非功能需求的度量

- **NFR:** 检验起来非常困难，一般采用一些可度量的特性进行描述。

非功能特性	度量指标
速度	每秒处理的事务 用户的响应时间 屏幕的刷新速度
存储空间	分配内存空间、外存空间
可用性	培训时间 帮助页面数
可靠性	平均失败时间 系统无效的概率 失败发生率
容错性	失败后的重启次数 事件引起失败的比例 失败时数据崩溃的可能性

一个例子：拼写检查器

- 业务需求：“用户能有效地纠正文档中的拼写错误”；
- 用户需求：“找出文档中的拼写错误并通过一个提供的替换项列表来供选择替换拼错的词”；
- 功能性需求：
 - 找到拼写错误的单词并以高亮度提示
 - 显示提供替换词的对话框
 - 实现整个文档范围的替换
- 非功能性需求：
 - 正确的找到至少95%以上的错词并100%的加以正确替换
 - 拼写检查的速度应至少达到5000词/秒。

业务规则

- **业务规则(Business Rule):** 对某些功能的可执行性或内部执行逻辑的一些限定条件。
 - 通常表达为“如果...，那么...”的形式
 - 通常是一些容易发生变化的功能；
- 例如：
 - 如果借书卡类型为“教师”，那么一次借阅的最大数量为8本；
 - 如果订单金额大于10000元，那么该订单的折扣为10%；
 - 如果采购单金额在10万到50万之间，那么需要总经理审批；

外部接口需求

- **外部接口需求(External Interface Requirement):** 描述系统与其所处的外部环境之间如何进行交互，包括：
 - 硬件接口需求
 - 软件接口需求
 - 通信接口需求
- 例如：
 - “从<某些设备>读取信号”
 - “给<一些其它系统>发送消息”
 - “以<某种格式>读取文件”
 - “能控制<一些硬件>”
 - “采用<某种类型的>用户界面”

关于需求的一些例子

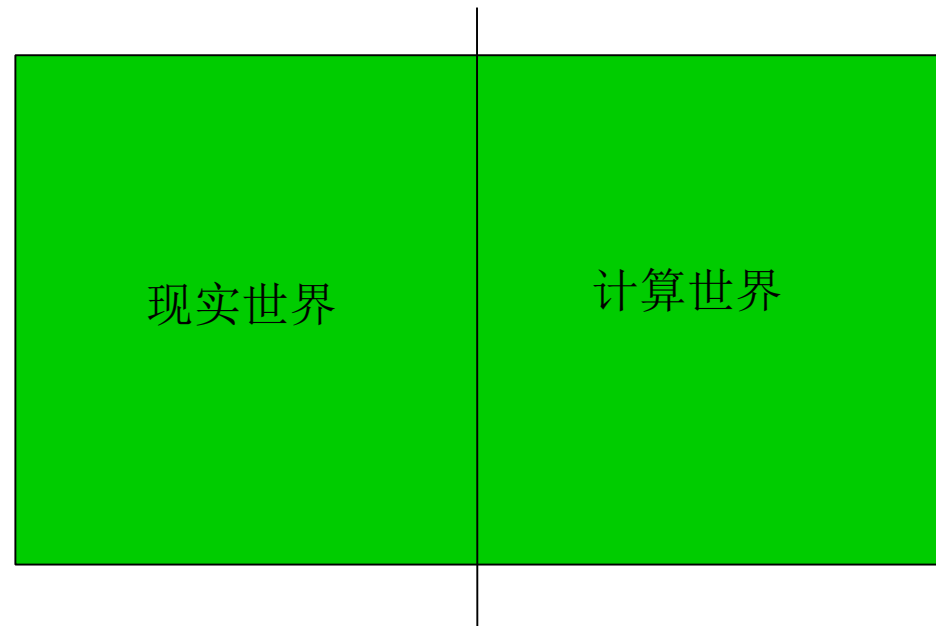
- 系统必须有能力和支持**100**个以上的并发用户，每个用户可以处理操作任务的任选组合，平均响应时间应该小于**1**秒，最大响应时间应小于**5**秒。
- 必须在对话框的中间显示错误警告，使用红色的、**14**点加粗**Arial**字体。
- 系统必须有能力和存储平均操作连续**100**天所产生的事务。
- 系统应该在**5**分钟内计算出给定季度的总销售税。
- 系统应该在**1**分钟内从**1000000**条记录中检索出一个销售订单。
- 系统必须支持**100**个**Windows**工作站的并行访问。
- 系统可从各型号的**modem**上读取信号作为系统输入。

本章内容

- **4.1 系统工程**
- **4.2 软件需求**
- **4.3 需求工程**
 - 4.3.1 为什么需要需求工程
 - 4.3.2 什么是需求工程
 - 4.3.3 需求工程的任务

模拟性

- 应该如何在计算世界中模拟现实世界？
 - 怎样建立共享知识？



现实世界

- 复杂

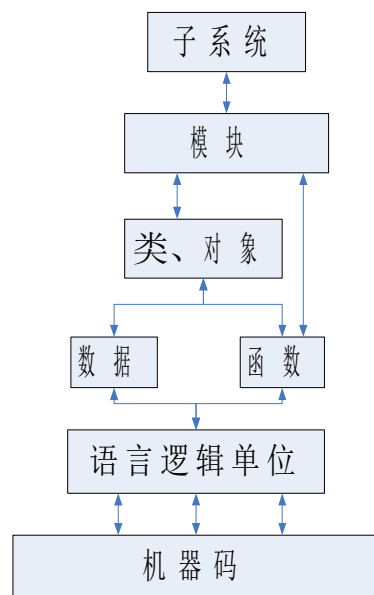
- 复杂性1：包含大量关联事物，具有巨量的分解和组合
- 复杂性2：对任一事物都不能完全描述，因为从不同的角度出发有不同的观察结论

- 非形式化的事物描述

- 实体
- 概念
- ...

计算世界

- 形式化描述
- 有限的实体单位



软件构建单位

- 合作
- 依赖
- 包含、继承
- 调用、使用
- 控制逻辑
- 序列

单位组织方法

需求分析

■ 需求分析

- 建立分析模型
- 以中立的（接近用户语言）语言、半形式化的方式定义“软件应该做什么”
 - 限定了模拟的范围，软件设计按照定义来安排计算实体及其组合方式
 - 缓和了用户和软件设计者之间（形式化 VS 非形式化）的交流差距

现实世界

用户模型

计算世界

分析模型

计算模型

机器世界

程序编码

机器指令

分析

设计

编码

编译

软件开发的混乱

Standish Group – 2004

调查对象：来自美国不同行业的
365个公司的**CIO**；**8,380**个不同的
项目；

调查的结果

- 预算： **189%**
- 时间： **221%**
- 功能： **61%**

average
time
overrun =
222%.

61% of originally specified
features included

average cost
overrun = 189%

成功项目的影响因素

成功项目的影响要素	影响指数
用户参与	15.9%
高层管理支持	13.9%
清晰的需求说明	13.0%
正确的项目计划	9.6%
切合实际的期望	8.2%
细化的项目里程碑	7.7%
员工能力	7.2%
主人翁精神	5.3%
清晰的目标和前景	2.9%
努力工作	2.4%
其他	13.9%

失败项目的影响因素

失败项目的影响要素	影响指数
不完整的需求说明	13.1%
缺少用户输入	12.4%
缺乏资源	10.6%
不切实际的期望	9.9%
缺乏高层管理支持	9.3%
需求变化	8.7%
缺乏计划	8.1%
额外的无用功能	7.5%
缺乏IT管理	6.2%
技术能力不足	4.3%
其他	9.9%

需求的重要性

■ 需求因素

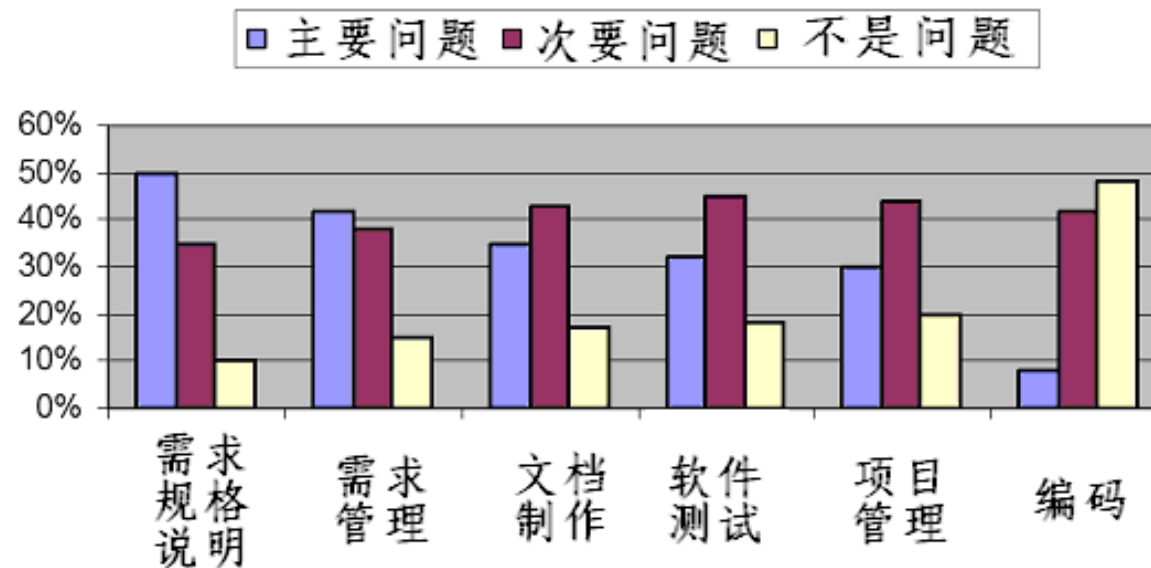
- 用户参与（用户输入）
- 高层管理支持
- 清晰的需求说明
- 切合实际的期望
- 清晰的目标和前景
- 需求变化
- 额外的无用功能

■ 综合来看，需求因素

- 对成功项目的影响指数为**53.9%**
- 对失败项目的影响指数为**60.9%**

另一组数字, ESPITI, 1996

- 欧洲软件协会**ESI**
- 欧洲软件过程改进培训计划项目**ESPITI**
- **17**个国家的超过**3800**个组织



本章内容

- **4.1 系统工程**
- **4.2 软件需求**
- **4.3 需求工程**
 - 4.3.1 为什么需要需求工程
 - 4.3.2 什么是需求工程
 - 4.3.3 需求工程的任务

需求工程

- 是软件工程的一个分支

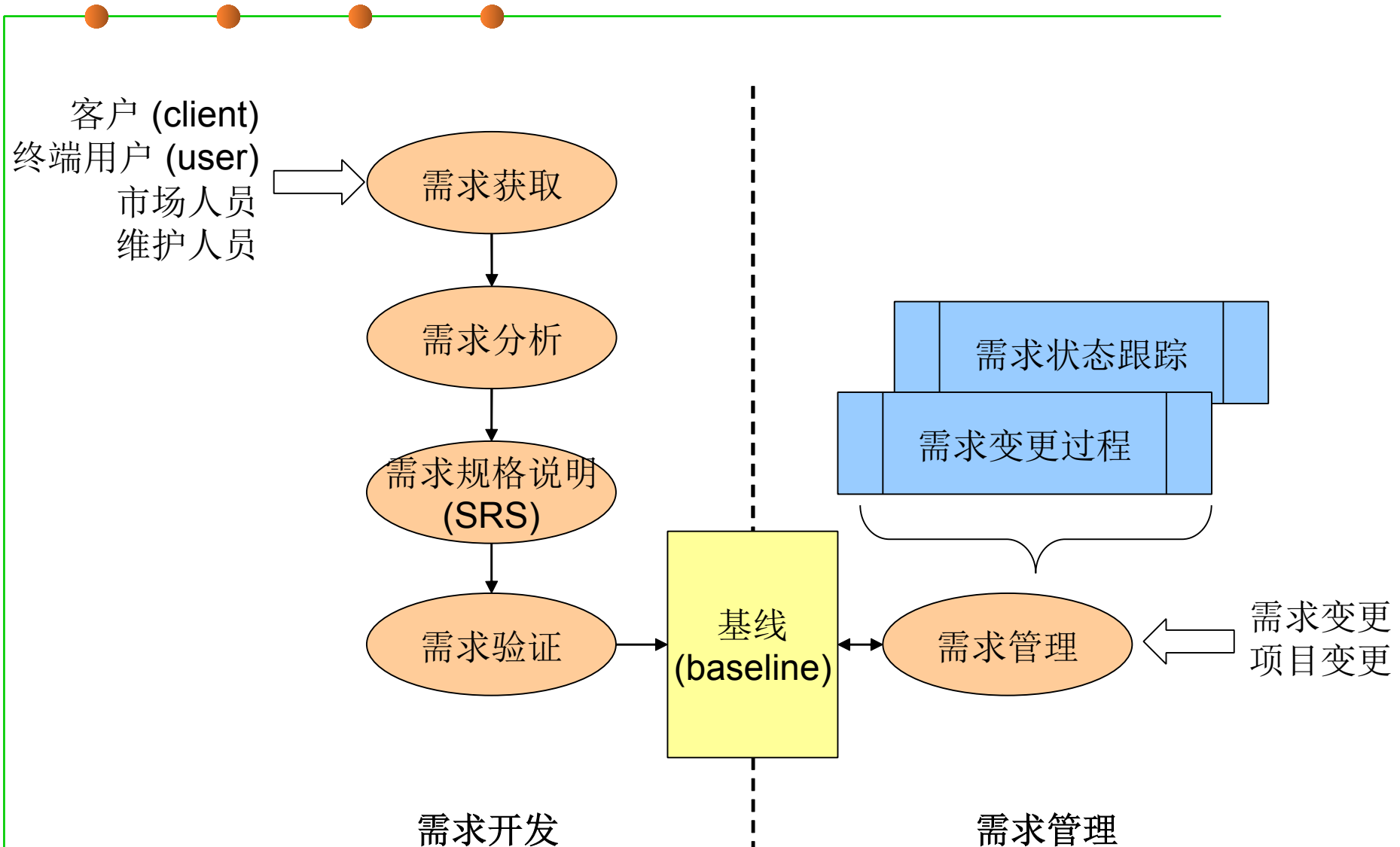
- 它关注于软件系统所应予实现的现实世界目标、软件系统的功能和软件系统应当遵守的约束
- 同时它也关注以上因素和准确的软件行为规格说明之间的联系
- 关注以上因素与其随时间或跨产品族而演化之后的相关因素之间的联系

需求工程

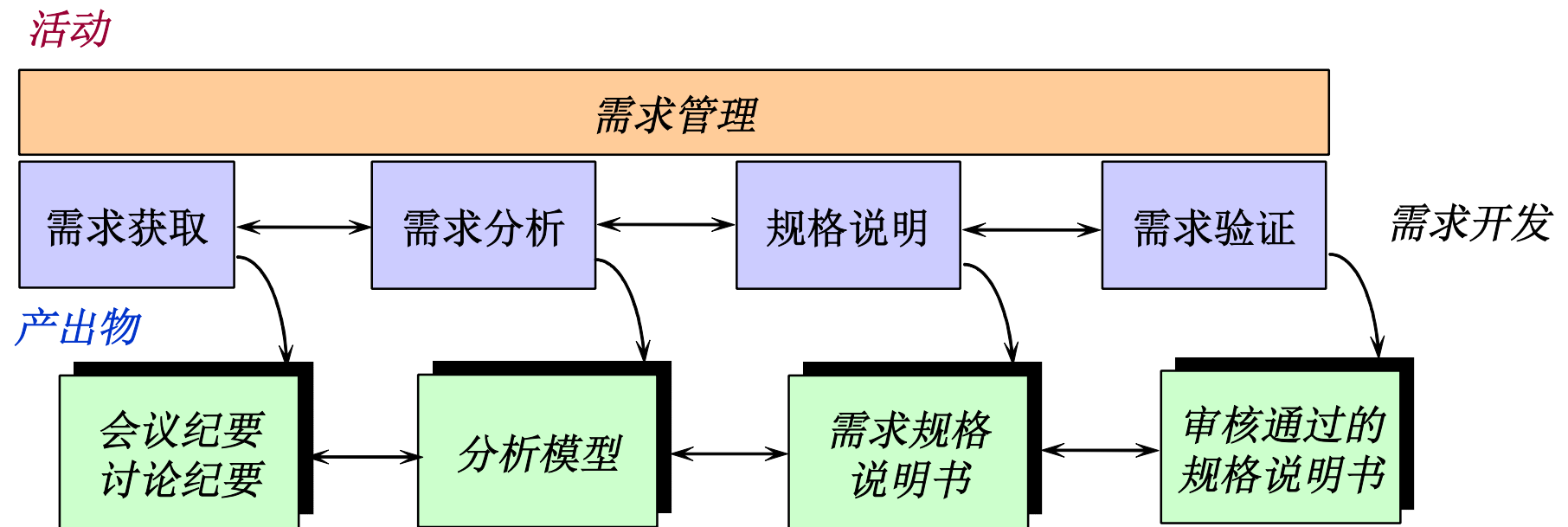
- 需求工程(**Requirement Engineering, RE**)
 - 是一个软件工程动作，开始于沟通并持续到建模
 - 需求工程在设计和构造之间建立起联系的桥梁



需求工程的总体流程



需求工程的总体流程



需求活动分类

- 起始
 - 导出
 - 精化
 - 协商
 - 规格说明
 - 确认
 - 管理
- } 需求获取
- } 需求分析

本章内容

- **4.1 系统工程**
- **4.2 软件需求**
- **4.3 需求工程**
 - 4.3.1 为什么需要需求工程
 - 4.3.2 什么是需求工程
 - 4.3.3 需求工程的任务

(1) 需求获取

- **需求获取(Requirement Elicitation):** 通过与用户的交流，对现有系统的观察及对任务进行分析，从而开发、捕获和修订用户的需求
 - 确定系统范围
 - 明确系统目标
 - 确定系统边界
 - 与客户沟通
 - 客户不完全确定需要什么
 - 对计算机能力及限制不了解
 - 对用户进行分类，聆听每一类用户的需求
 - 形成文档化的描述
 - 分析和整理所获取的需求
 - 形成需求规格说明书（FURPS+）

(2) 需求分析

- **需求分析(Requirement Analysis)**: 对收集到的需求进行提炼、分析和审查, 为最终用户所看到的系统建立概念化的分析模型
 - 分析需求可行性
 - 精化需求, 定义问题的信息域、功能域和行为域
 - 建立需求分析模型
 - 建立软件原型

(3) 形成需求规格说明

- **需求规格说明(Software Requirement Specification, SRS):**
 - 需求分析的结果
 - 精确的、形式化的阐述一个软件系统必须提供的功能、非功能、所要考虑的限制条件等
 - 作为用户和开发者之间的一个契约
 - 是用户、分析人员和设计人员之间进行理解和交流的手段

(4) 需求验证

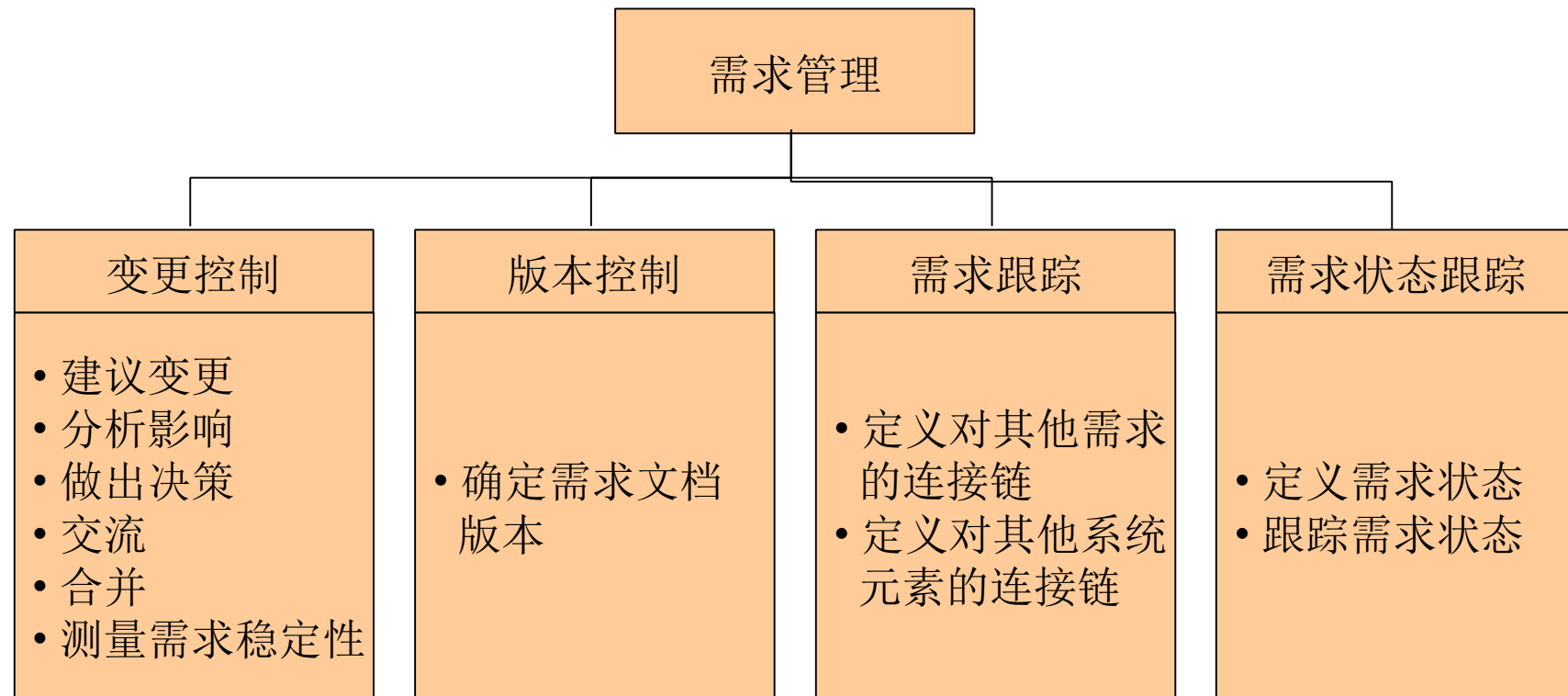
- **需求验证(Requirement Verification)**: 以需求规格说明为输入, 通过评审、模拟或快速原型等途径, 分析需求规格的正确性和可行性, 发现存在的错误或缺陷并及时更改和补充。

(5) 需求管理

■ 需求管理(Requirement Management)

- 定义需求基线(迅速制定需求文档的主体)
- 评审提出的需求变更、评估每项变更的可能影响从而决定是否实施它
- 以一种可控制的方式将需求变更融入到项目中
- 使当前的项目计划与需求一致
- 估计变更需求所产生影响并在此基础上协商新的承诺(约定)
- 让每项需求都能与其对应的设计、源代码和测试用例联系起来以实现跟踪
- 在整个项目过程中跟踪需求状态及其变更情况

需求管理





结束

2011年4月16日