



IBM Software Group

## 明确软件需求，提高需求质量

**Rational.** software

IBM 软件部 傅纯一

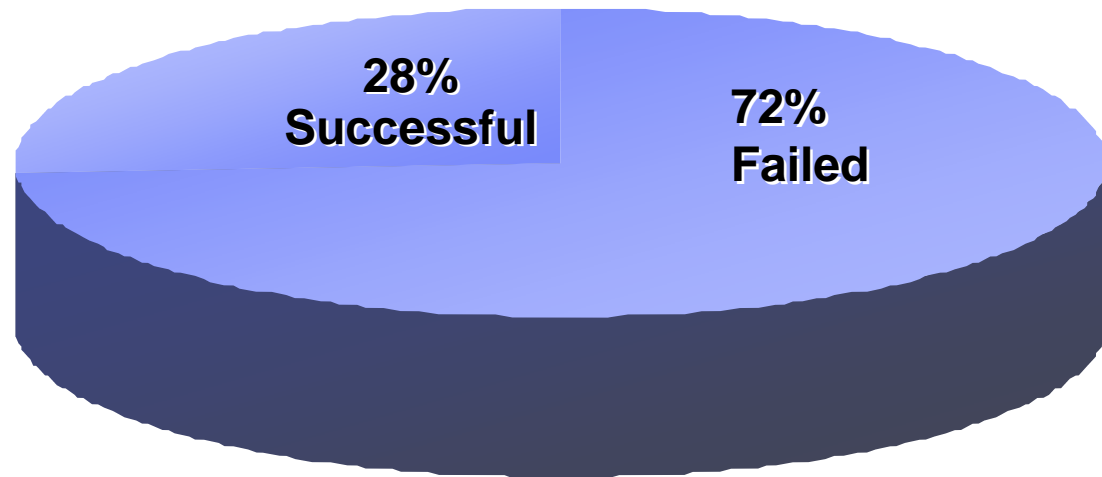


**ON DEMAND BUSINESS™**

© 2005 IBM Corporation

## 软件业的现状

- 超预算
  - ▶ 平均超出费用: 189%
- 推迟发布
  - ▶ 平均超出时间: 222%
- 不能满足期望
  - ▶ 平均覆盖率: 61%



## 项目失败的原因

- ✓ 缺少用户输入
- ✓ 目标不明确
- ✓ 不完整的需求和规范
- ✓ 不断变化的需求和规范
- ✓ 缺乏计划

## CMMI 中关于需求的两个过程域

- 需求开发 RD (Requirement Development)
  - ▶ 开发并分析客户、产品和构件的需求
- 需求管理 REQM (Requirement Management)
  - ▶ 管理项目中产品和构件的需求，从而能够及时发现需求、项目计划和工作产品之间的不一致性



## 需求工作的两个方面

- 需求开发 RD (Requirement Development)
  - ▶ 业务建模 - 更好地理解客户的业务流程
  - ▶ 用例建模 - 清晰地描述软件需求
  - ▶ RUP – 统一需求工件模板
- 需求管理 REQM (Requirement Management)
  - ▶ 管理需求的追踪性
  - ▶ 管理需求的变更

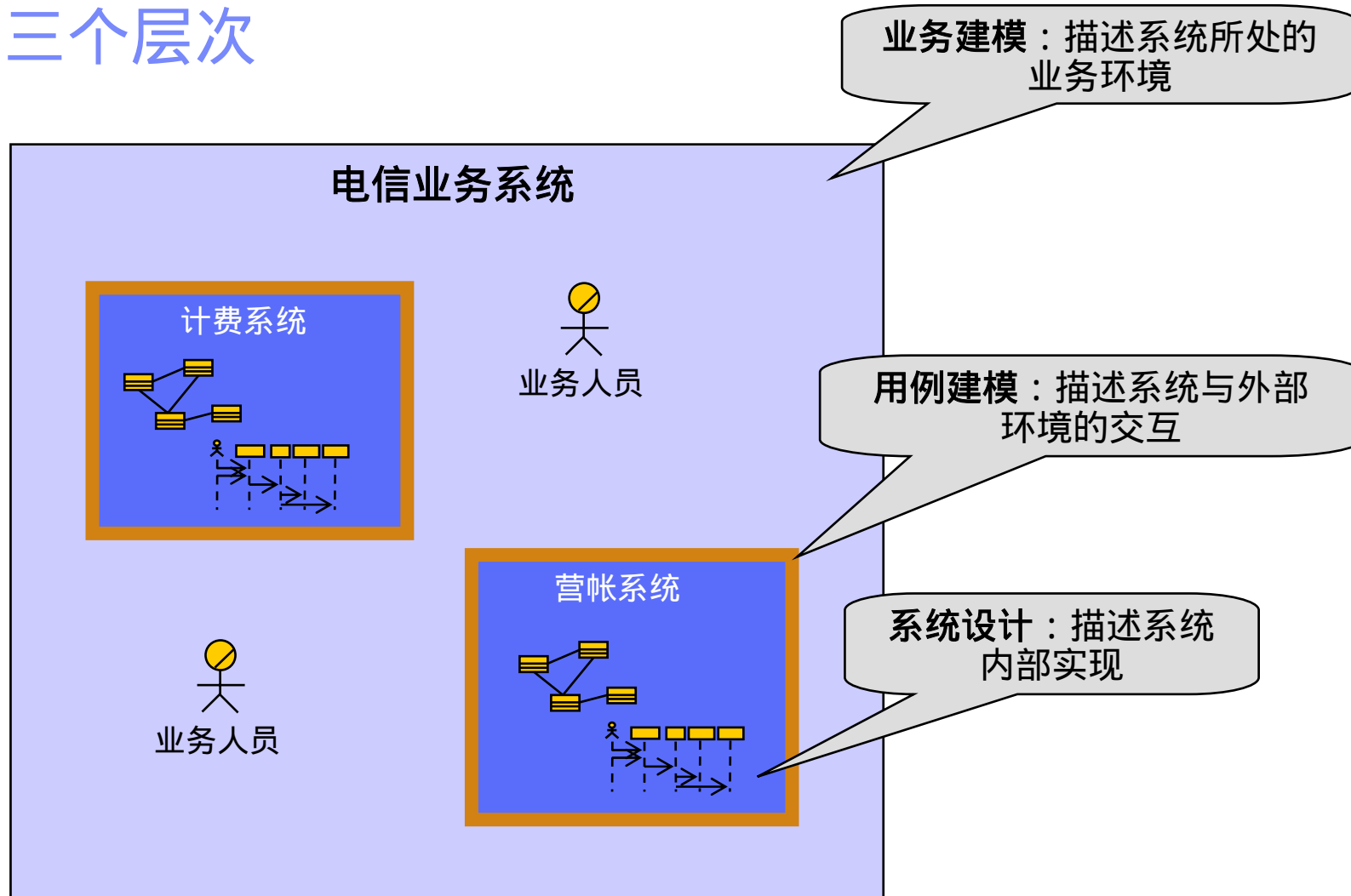


## 需求开发中的常见问题

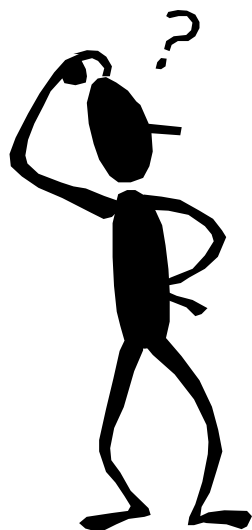
- 很难理解客户的业务需求
  - ▶ 业务部门提出的业务需求根本看不懂
  - ▶ 开发团队定义的需求客户不愿意看
- 不知如何定义需求
  - ▶ 每个项目采用了不同的方法来定义需求
  - ▶ 在需求文档中将业务流程、软件需求、概要设计



# 建模的三个层次



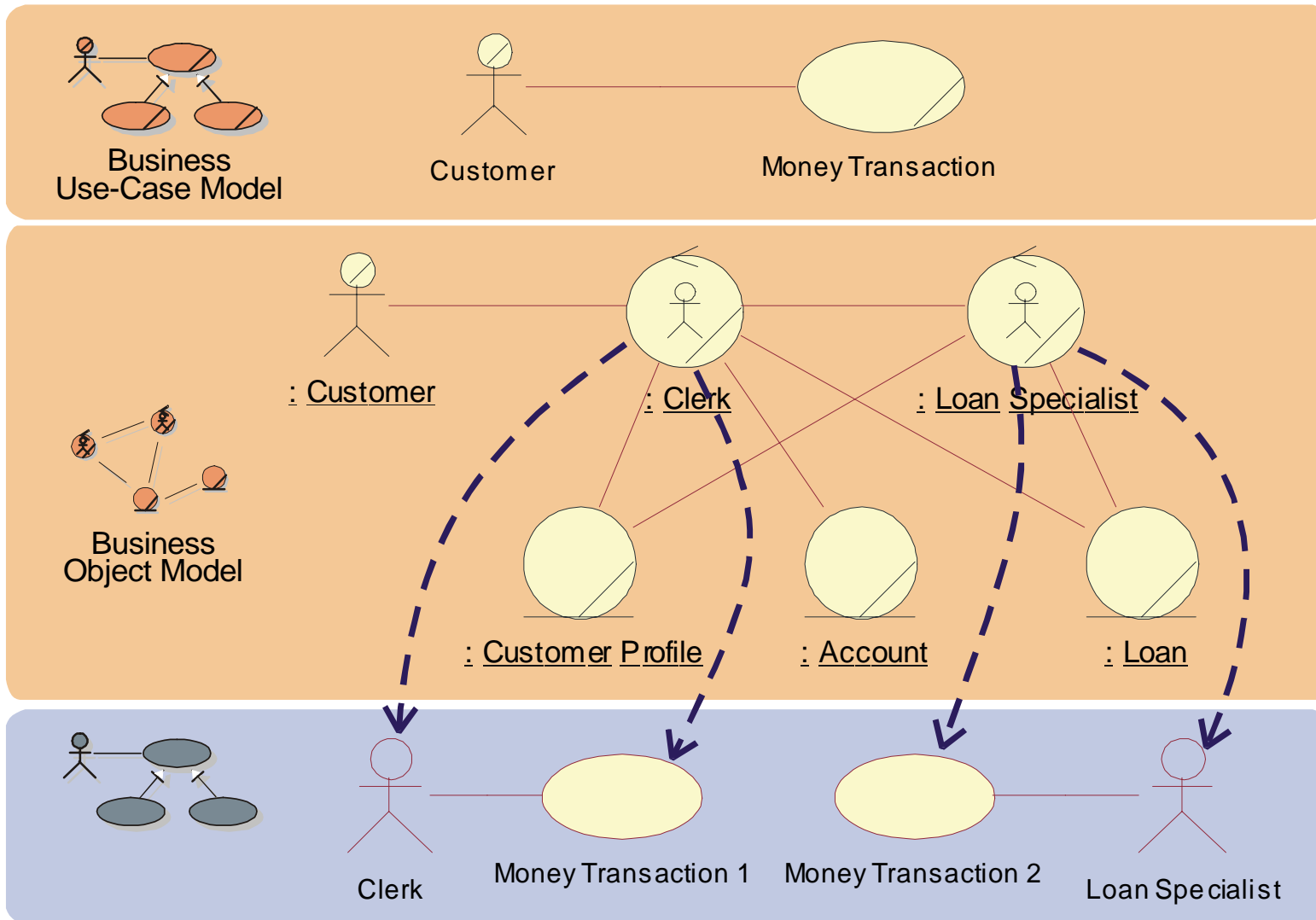
## 业务模型 – 描述业务流程



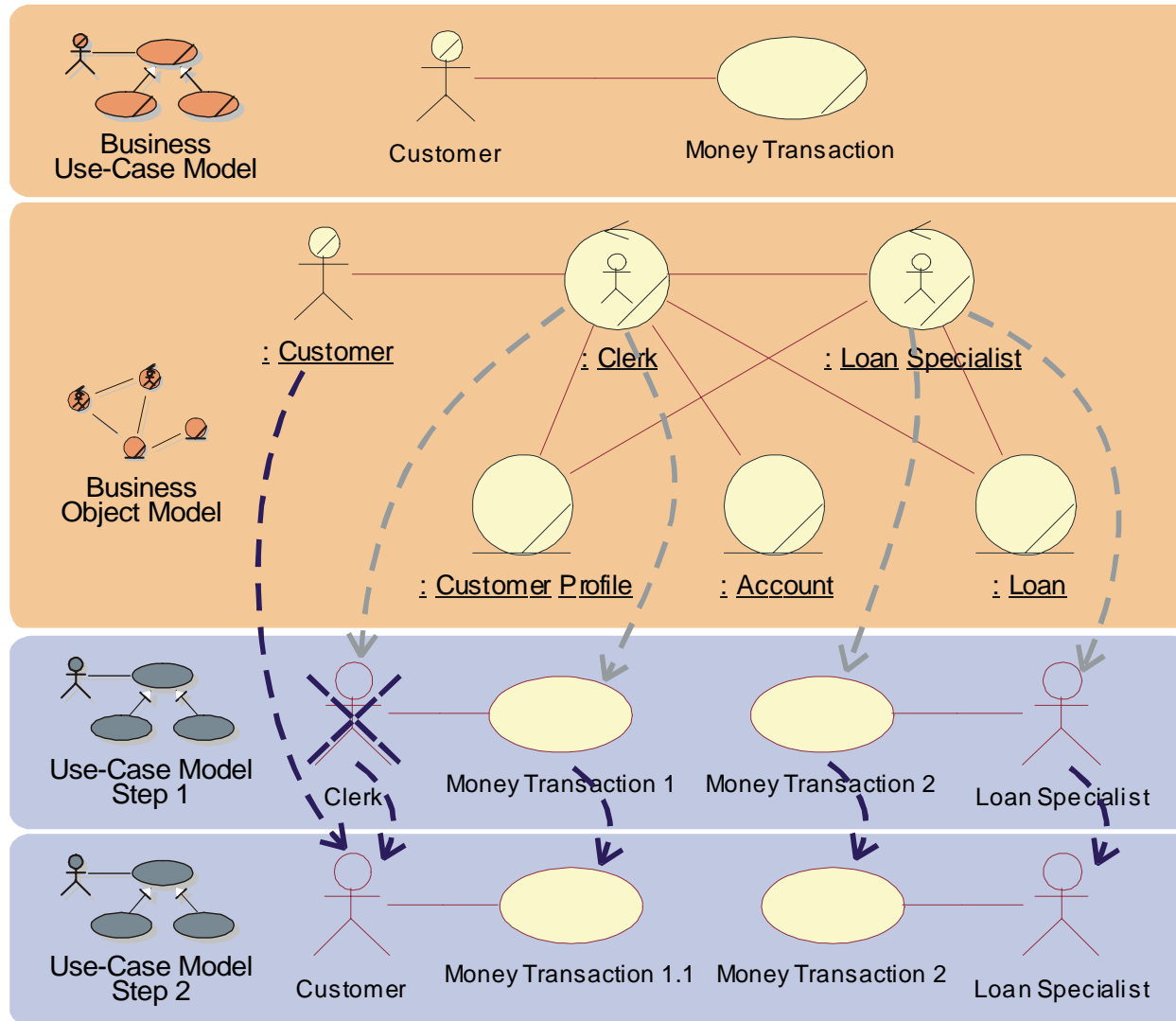
- 业务模型可以描述：
  - ▶ 组织结构
  - ▶ 业务流程
  - ▶ 角色及其责任
  - ▶ 客户
  - ▶ 产品
  - ▶ 事件
  
- 业务模型可以帮助理解所要解决的问题



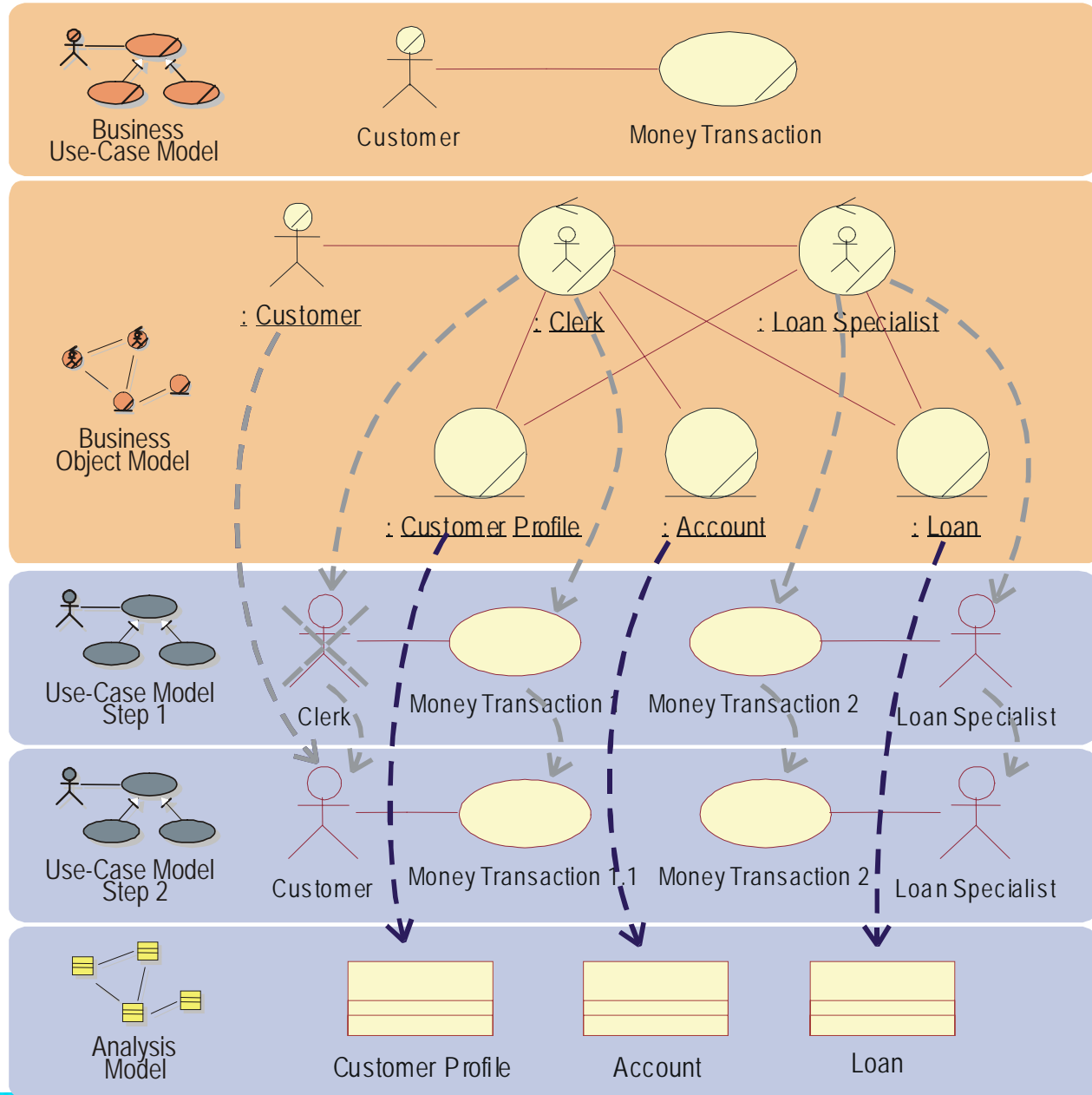
# 业务模型到用例模型的映射(1)



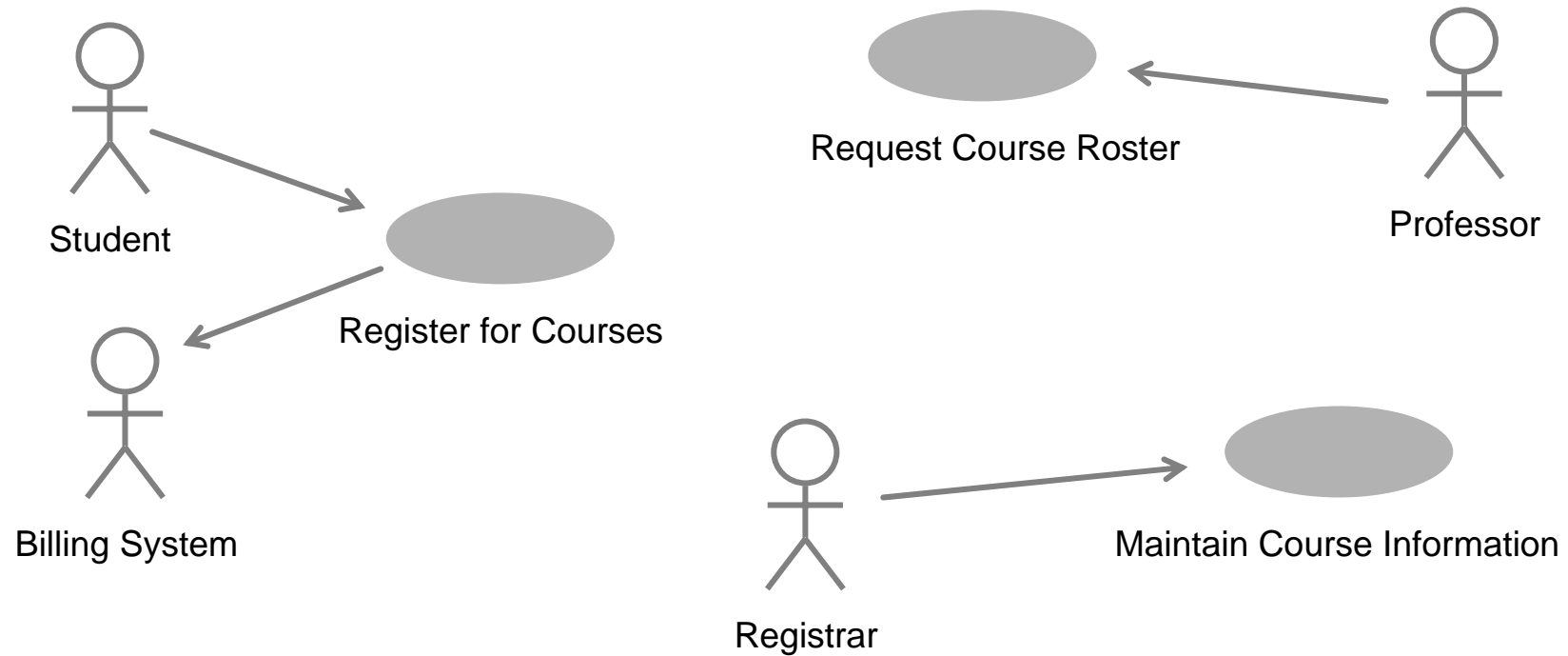
# 业务模型到用例模型的映射(2)



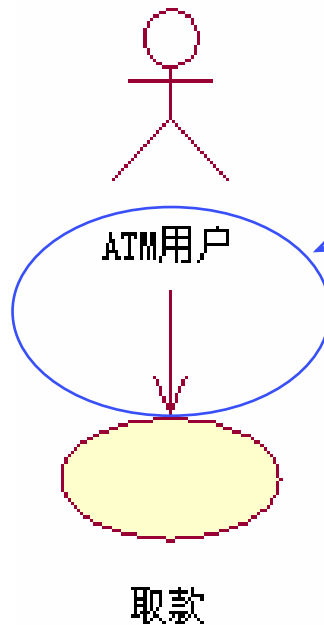
业务模型到设计模型的映射



# 用Use Case捕获需求



## 用例图



Use Case立足用户视角的描述，为具体的需求提供了充分的上下文信息，是衔接用户和开发者的纽带和沟通方式

## 用例规约

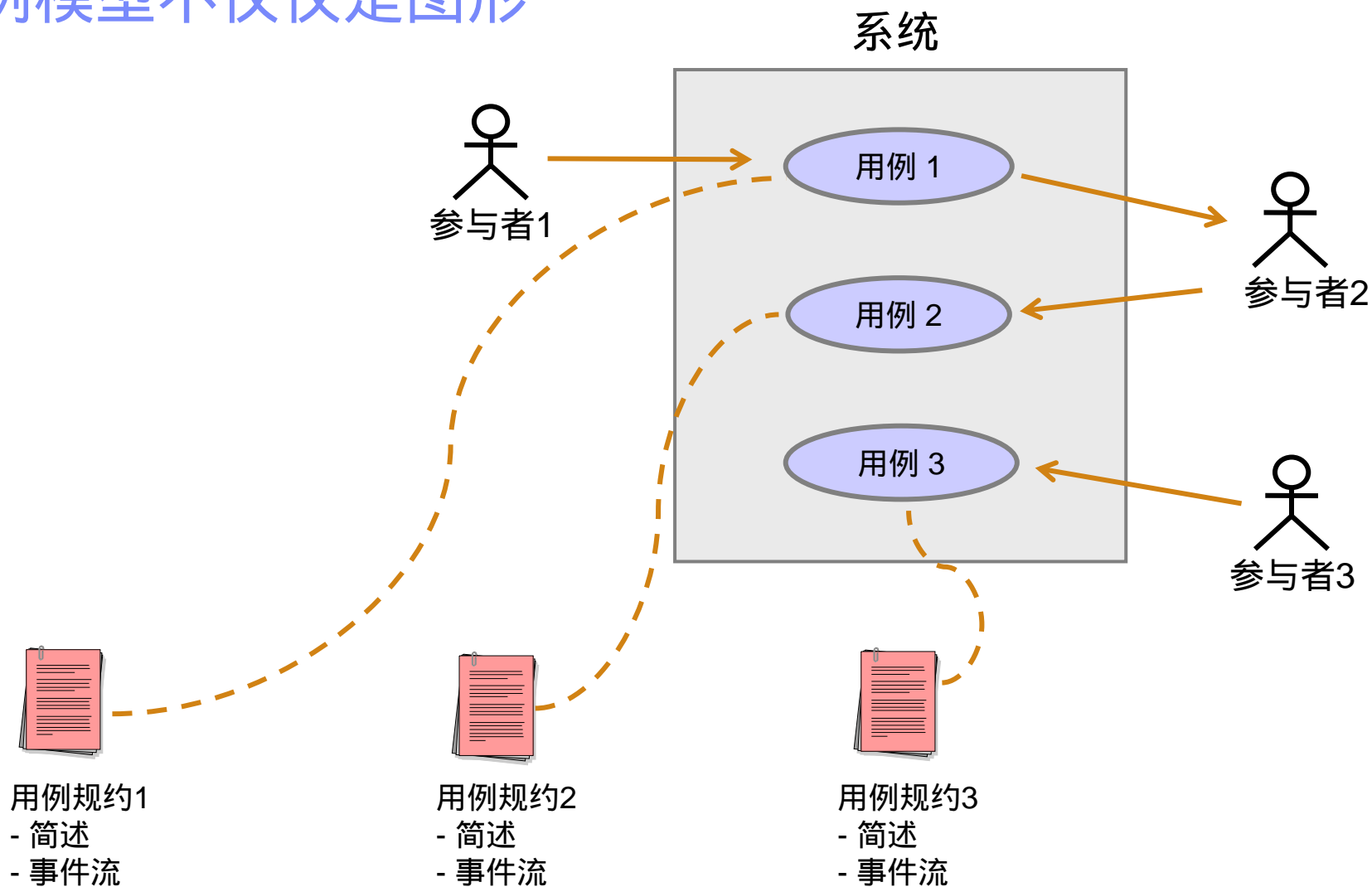
## 基本事件序列( Basic Flow )

- ◆ 用户插入ATM卡
- ◆ 系统要求输入合法的密码
- ◆ 用户输入正确密码，如果用户输入的密码有误，转至备选事件流A1
- ◆ 系统提示用户选择“存款”或者“取款”
- ◆ 用户选择“取款”
- ◆ 系统提示用户输入取款金额
- ◆ 用户输入(合理)取款金额并确认，如果取款金额不合理,转至备选事件序列A2
- ◆ 系统从帐户中扣除取款金额，提示用户“打印收据”或者“不打印收据”
- ◆ 用户要求不打印收据，如果要求打印收据，转至备选事件序列A3
- ◆ 系统显示“交易结束”

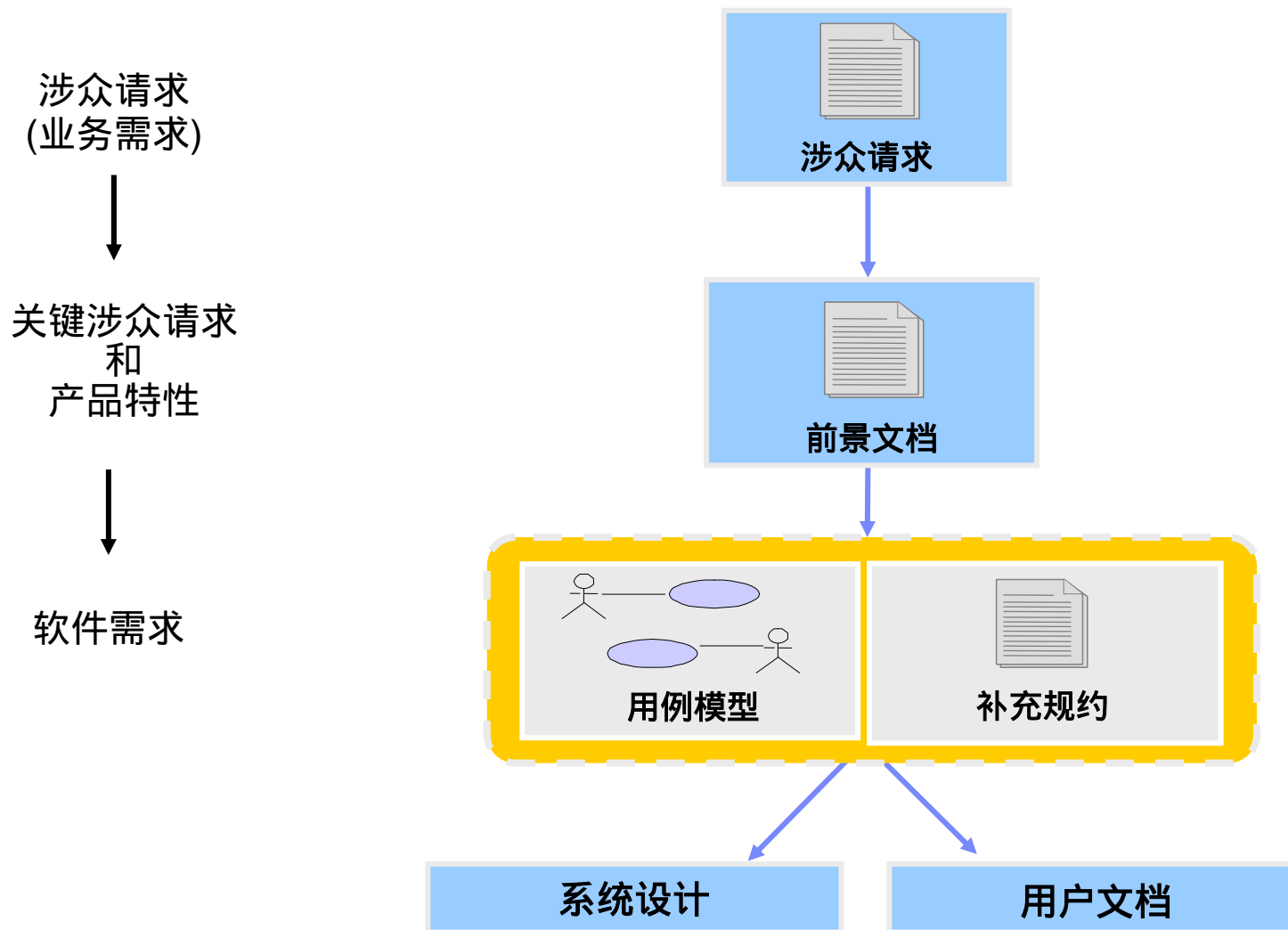
## 备选事件序列( Alternative Flows )

- ◆ A1. ...
- ◆ A2. ...
- ◆ ...

# 用例模型不仅仅是图形

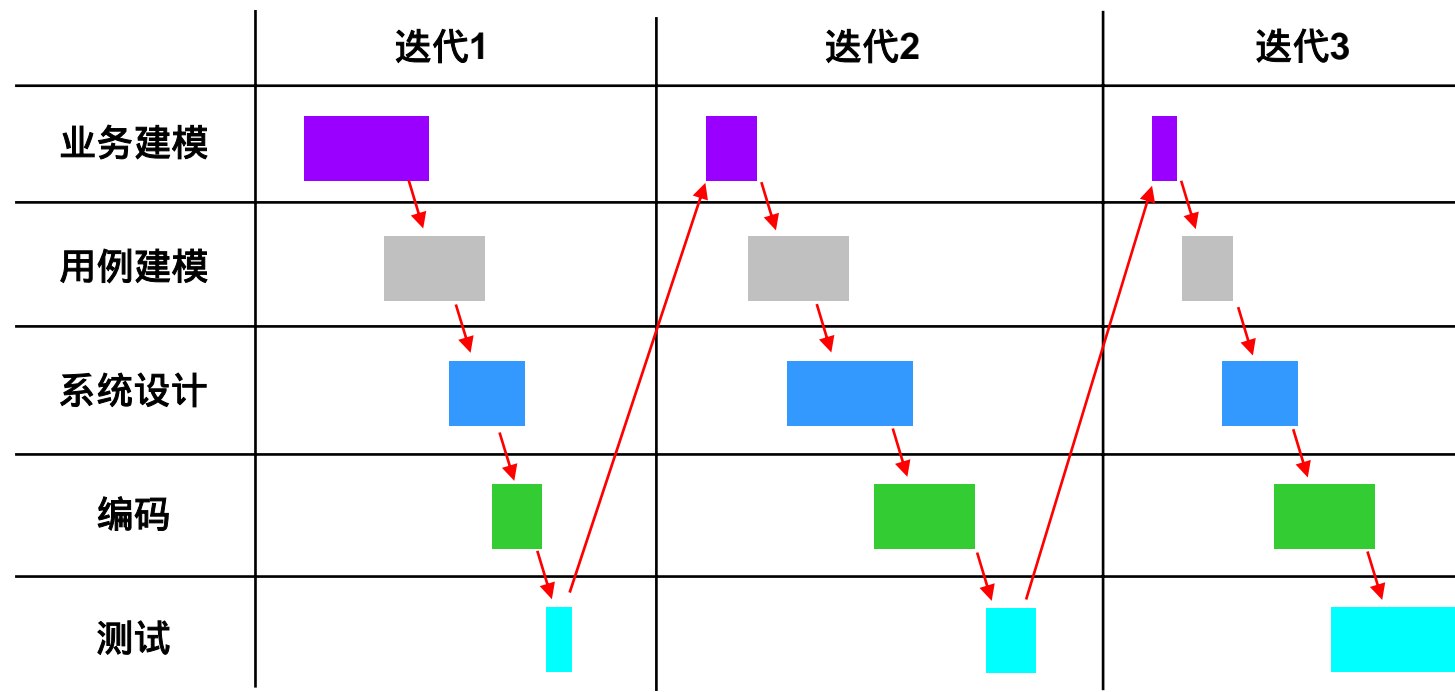


# RUP 提供了成熟的需求文档体系



## 先做业务建模，后做用例建模？

- 实际工作中，应该把这几种建模工作看作是对业务、需求和设计的记录
- 这些建模工作贯穿迭代开发的整个过程





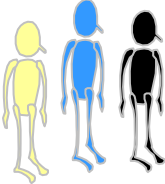
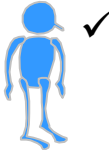
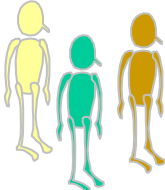
## 需求工作的两个方面

- 需求开发 RD (Requirement Development)
  - ▶ 业务建模 - 更好地理解客户的业务流程
  - ▶ 用例建模 - 清晰地描述软件需求
  - ▶ RUP – 统一需求工件模板
- 需求管理 REQM (Requirement Management)
  - ▶ 管理需求的追踪性
  - ▶ 管理需求的变更

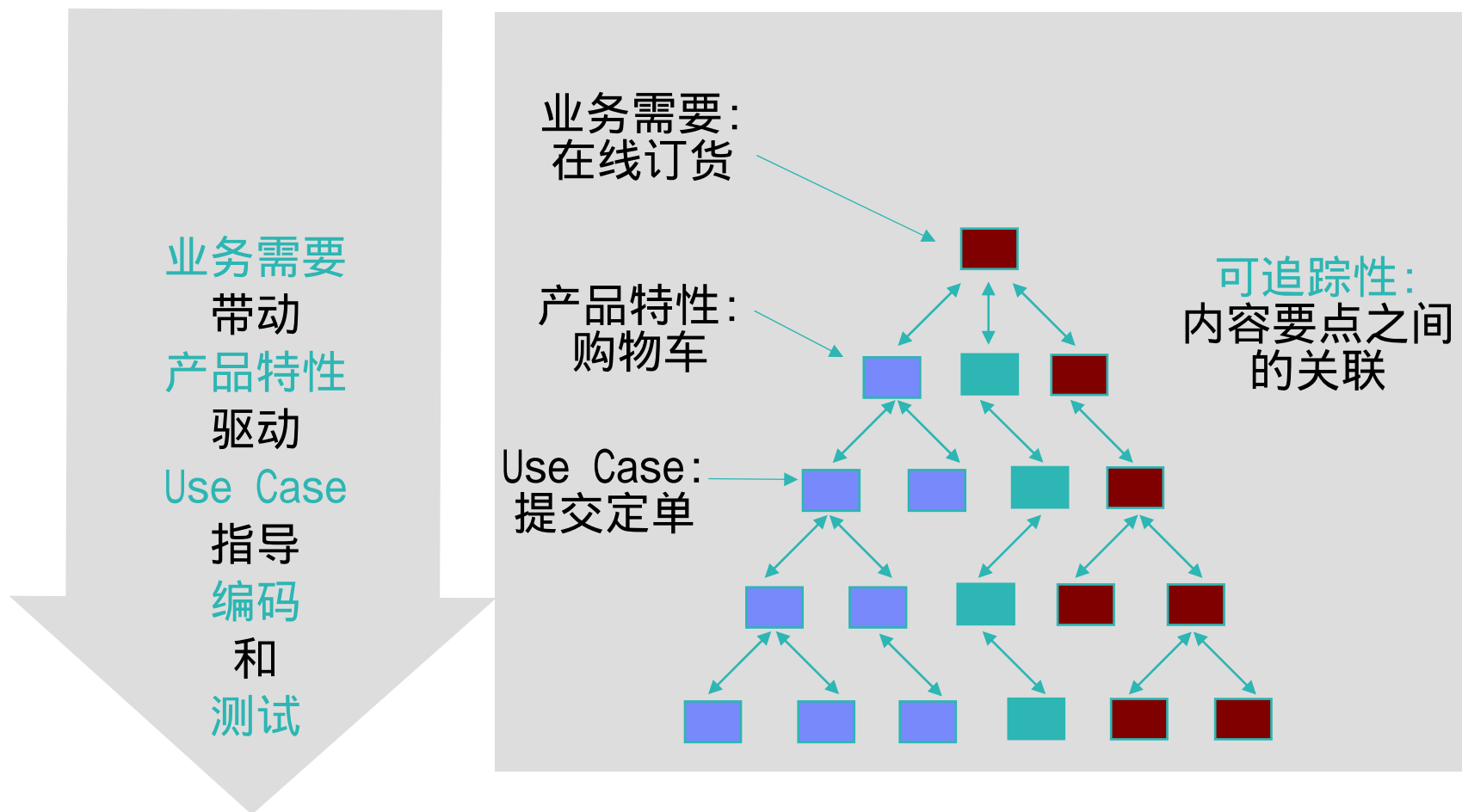


## 利用属性来更全面地描述需求

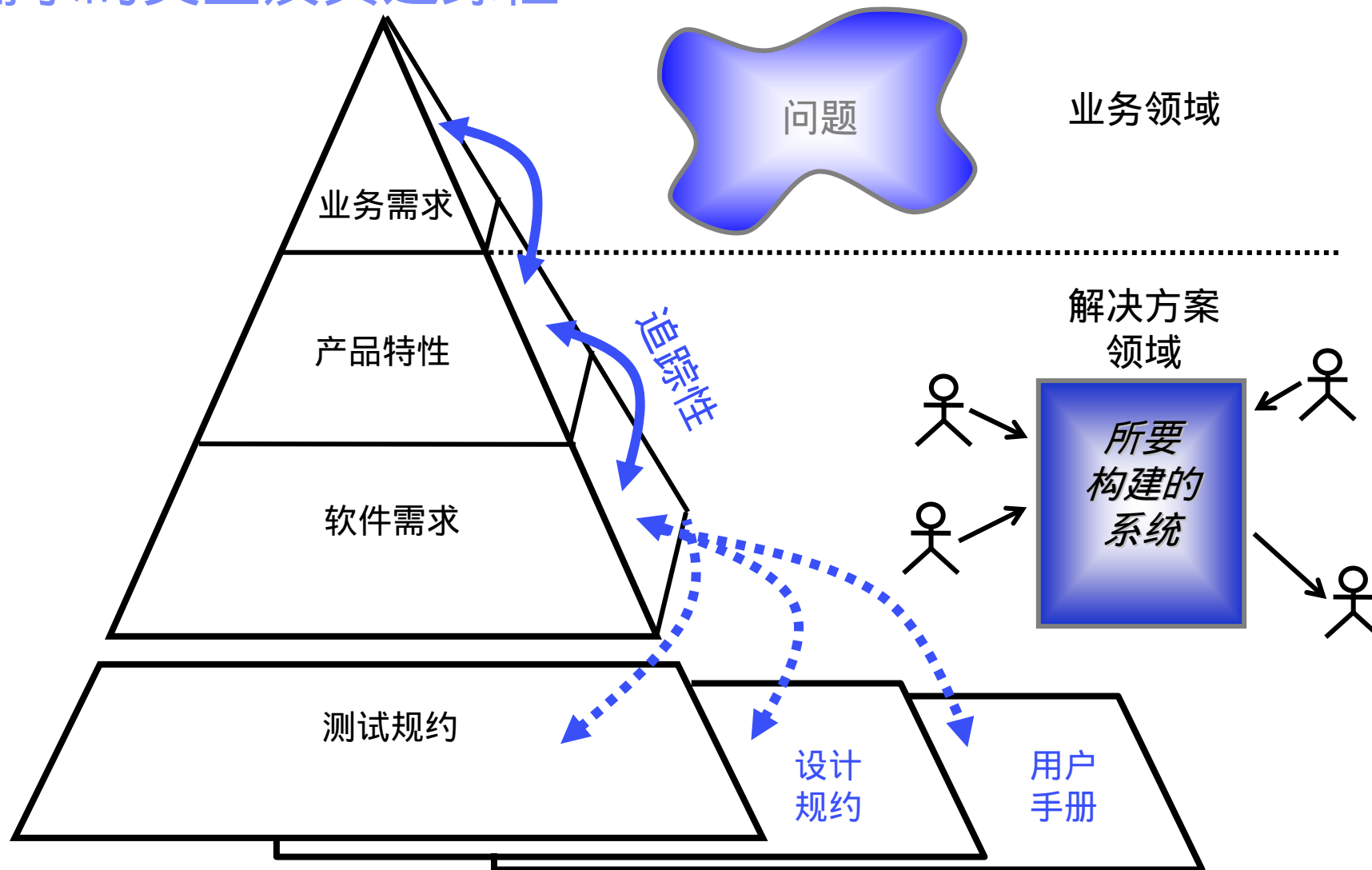
需求的属性为规划基线提供了丰富的信息，它将需求和项目的其他要素有机地联结在一起...

	状态	风险	优先级	工作量	成本	稳定性	来源
需求 10	批准的	低	高 ✓		\$\$\$	高	客户 ✓
需求 13	被提议	中等	低		\$\$	低	用户
需求 40	强制的 ✓	高 ✓	高		\$ ✓	高	用户

# 需求变化和变化的传播



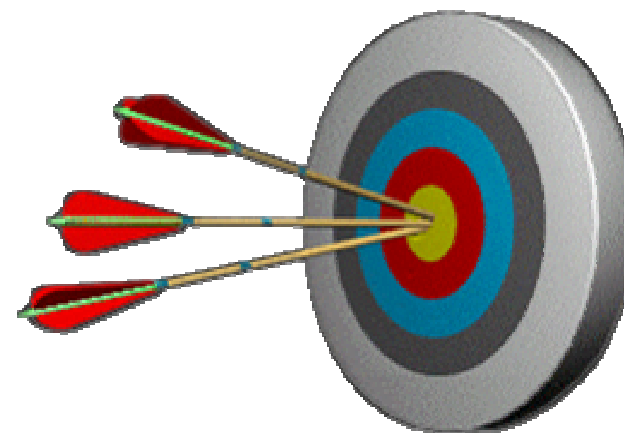
# 需求的类型及其追踪性





## 你的需求受控后

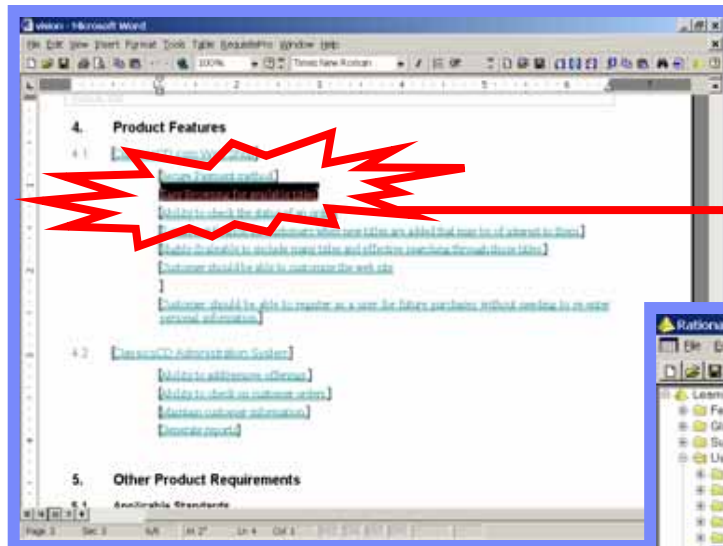
- 访问最新的需求
  - ▶ 安全、中央控制存储库
- 组织并对需求排序
  - ▶ 和数据库动态链接
- 了解项目状态
  - ▶ 过滤、排序、度量



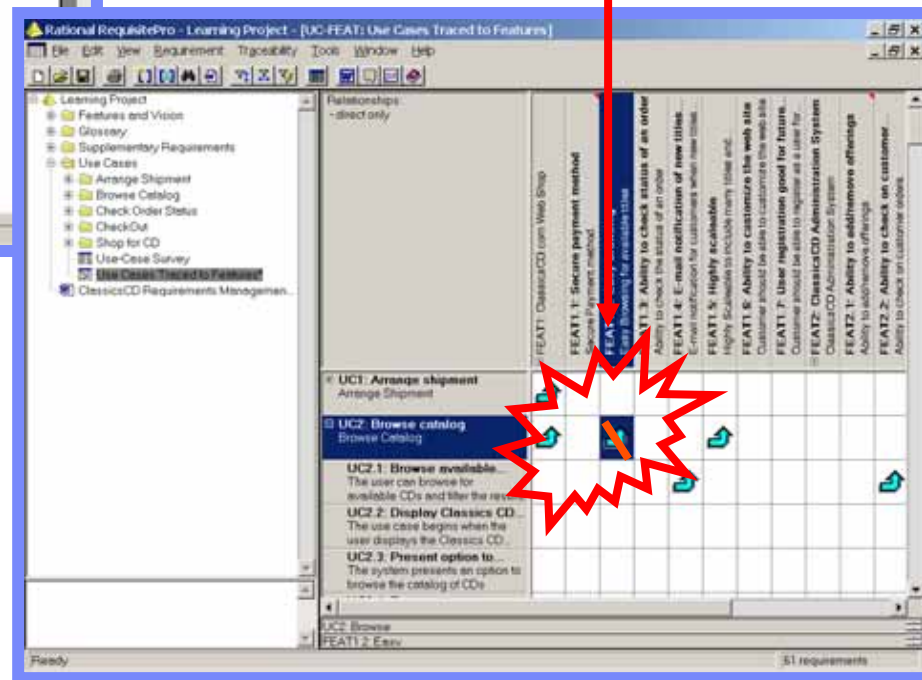
*结合数据库的强大功能和  
word的灵活性*

# RequisitePro 管理需求之间的追踪性

保证需求的覆盖，所有关键的需求都被正确地实现了



评估需求变化对于项目所造成的影响



# 所有人都需要访问需求



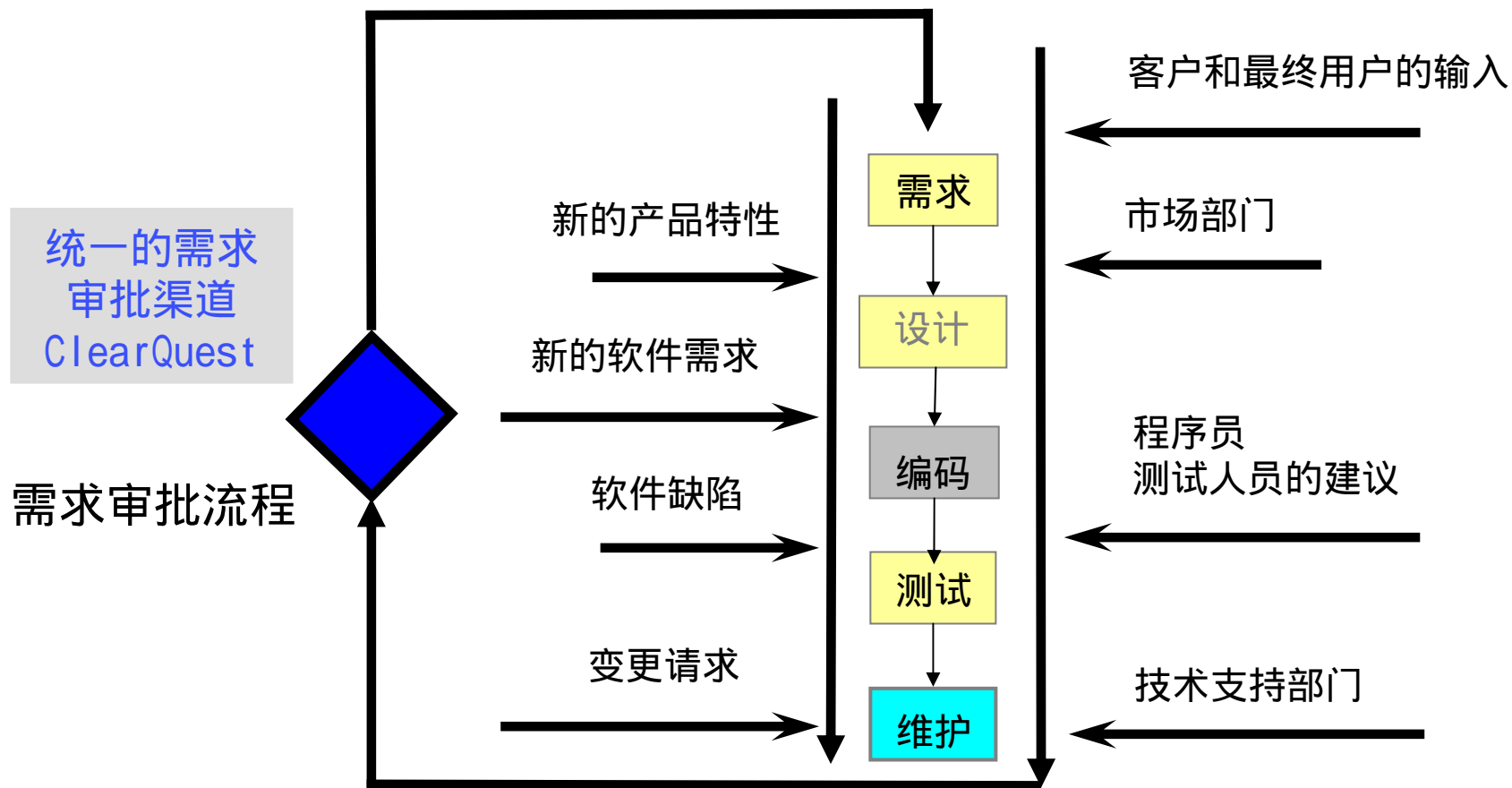
*“Rational RequisitePro 把我们的项目  
团队有机地整合起来。”*

Mireille Mallouh  
MCI WorldCom

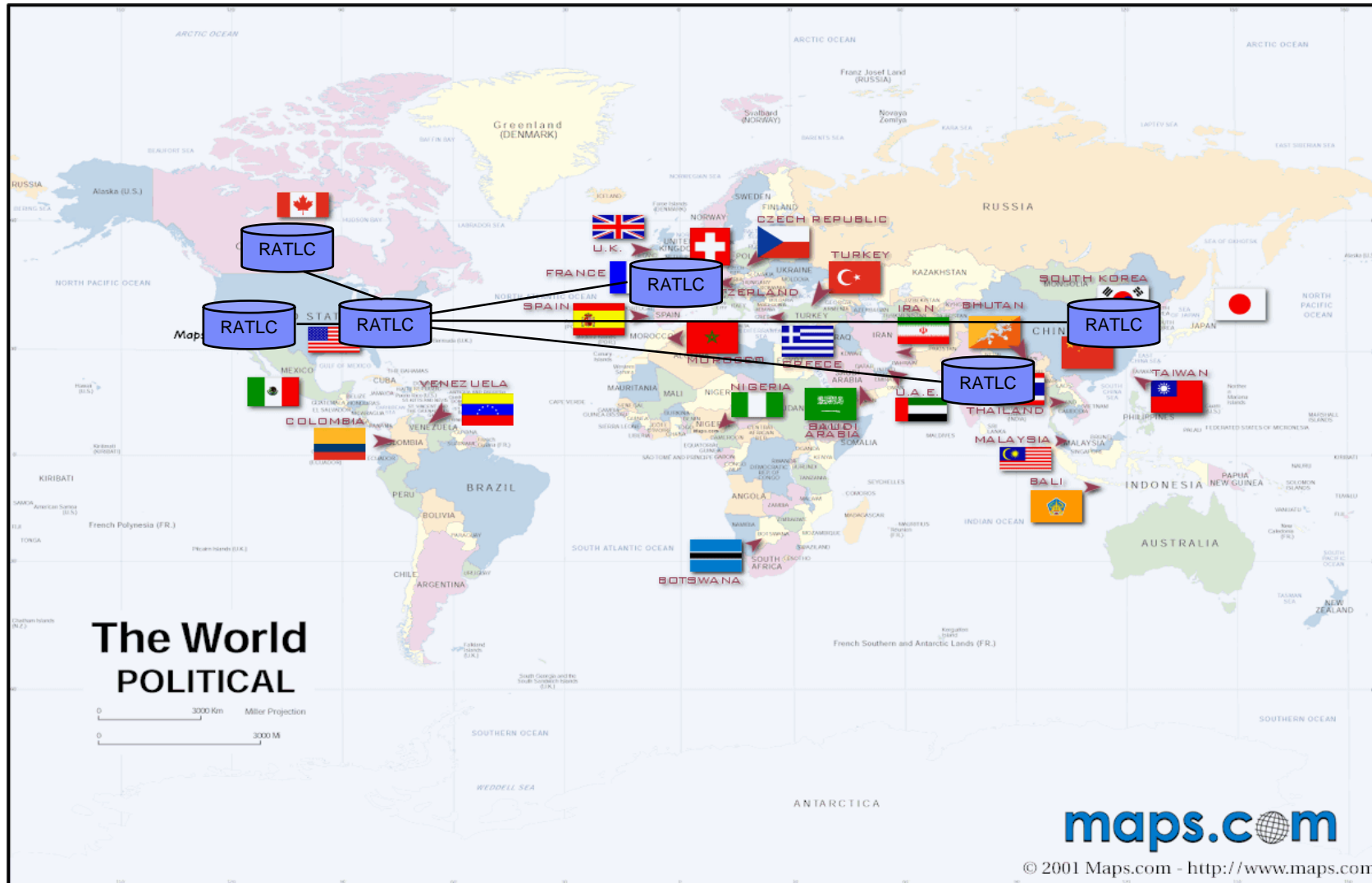


# 所有的需求变更都要经过统一的审批渠道

在产品开发生的过程中，需求的变更请求可能来自于各种渠道



# 案例学习：Rational 自己的需求管理系统 RATLC

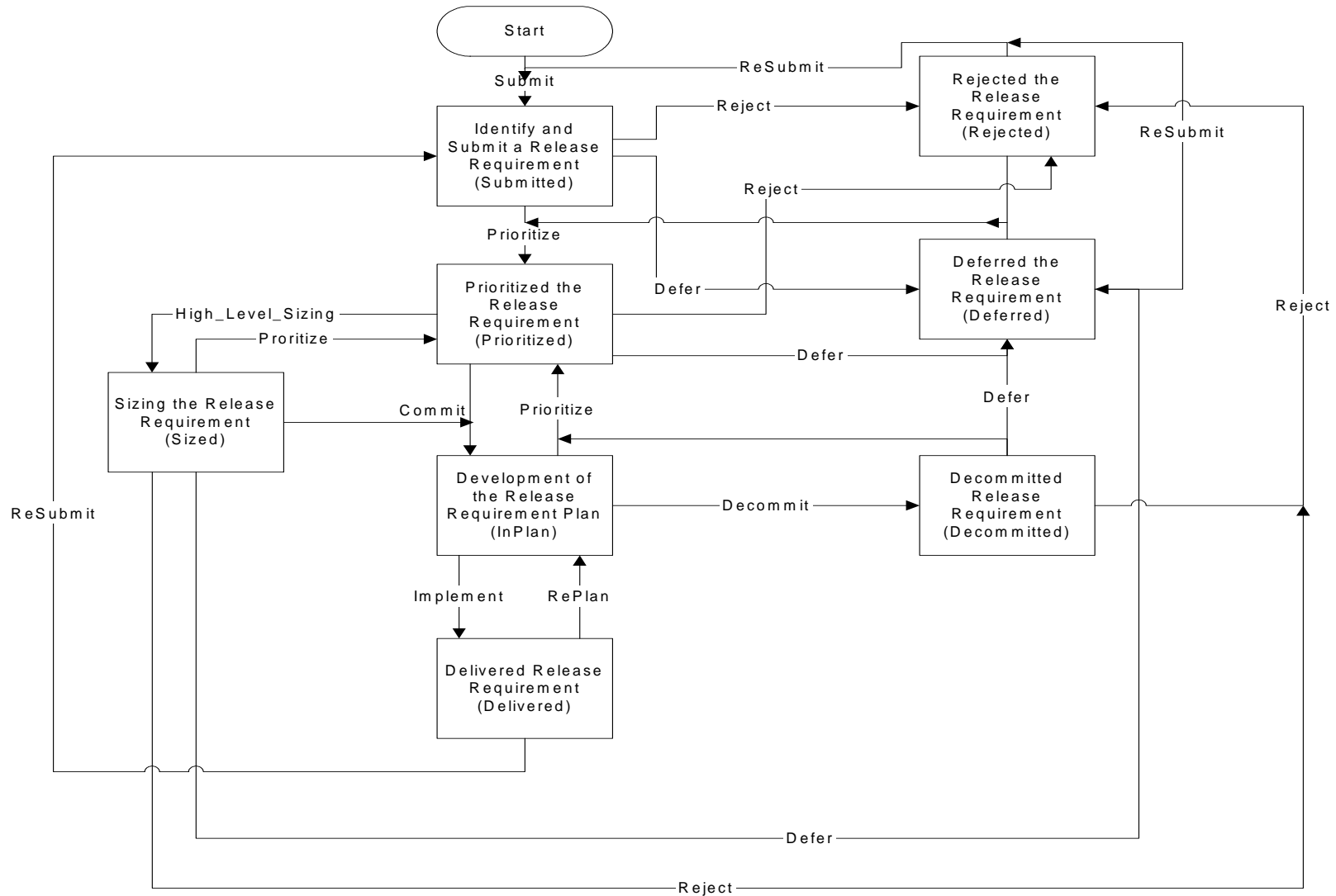


## 什么是 RATLC

- RATLC 是 IBM Rational 用于管理自己产品的软件缺陷、功能增强请求、发布需求的一个 ClearQuest 系统
- 它同时又是一个分布式系统，通过 ClearQuest MultiSite 分布在全球各地，总共有 11 个 Replica，分别位置北美、印度、法国和中国
- RATLC 中包含有超过250,000 条记录，支持全球超过2,700 名用户(开发人员)



# RATLC - 工作流程定义



## 总结

- 需求开发 RD (Requirement Development)
  - ▶ 业务建模：更好地理解客户的业务流程 – RSM
  - ▶ 用例建模：清晰地描述软件需求 – RSM
  - ▶ RUP：统一需求工件模板
- 需求管理 REQM (Requirement Management)
  - ▶ 管理需求的追踪性 – RequisitePro
  - ▶ 管理需求的变更 – ClearQuest



# Questions

THANK YOU