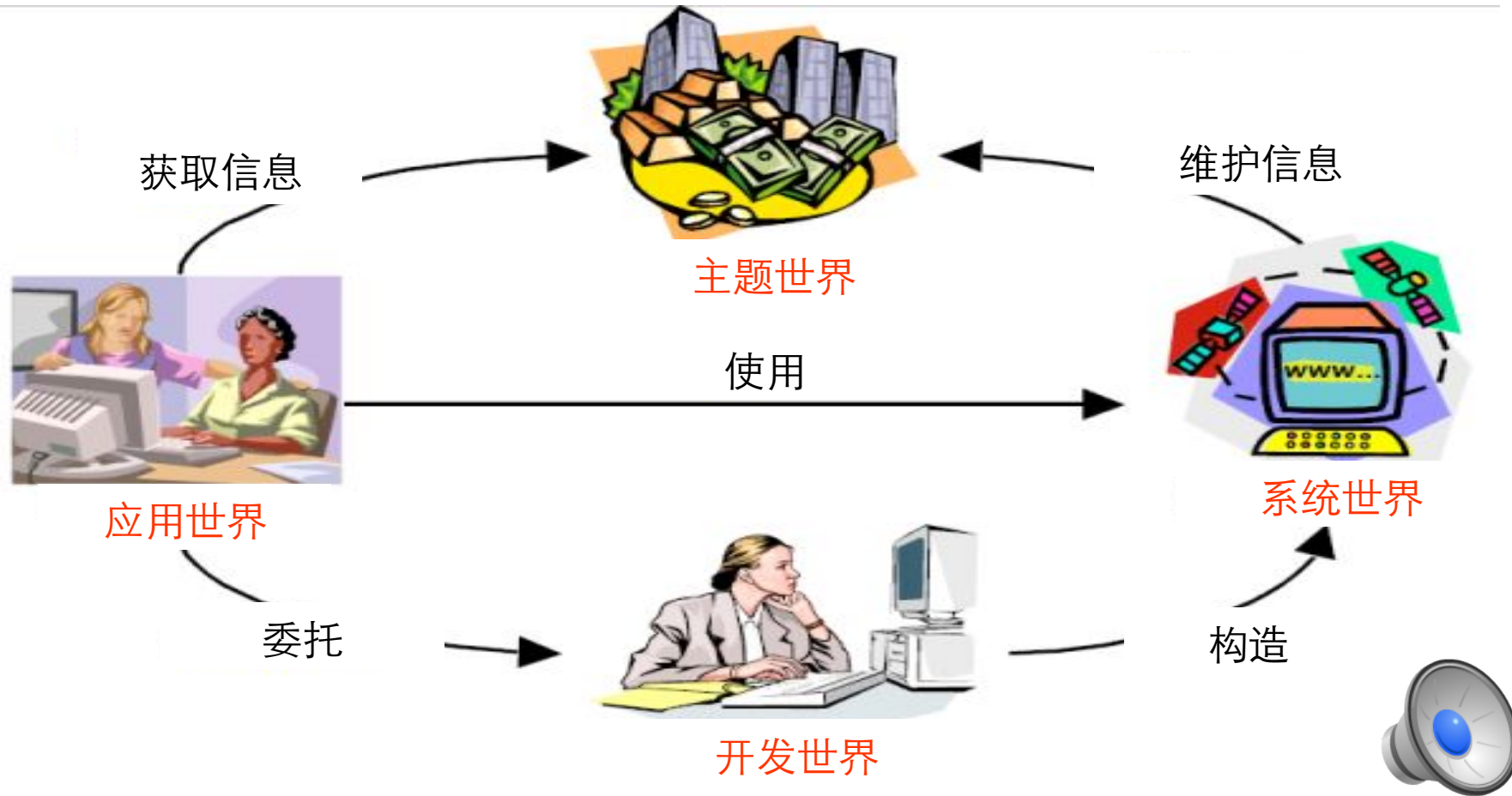


# 四世界模型

Adapted from Loucopoulos & Karakostas, 1995, p73



# 出发点

- 确定干系人

- 这里需要强调与客户之间的联络关系
- 系统的设计到底与谁的利益息息相关

- 定义边界

- 怎样界定问题的范围？

- 定义目标与情景实例

- 目标与情景实例是组织原始需求信息的有效手段

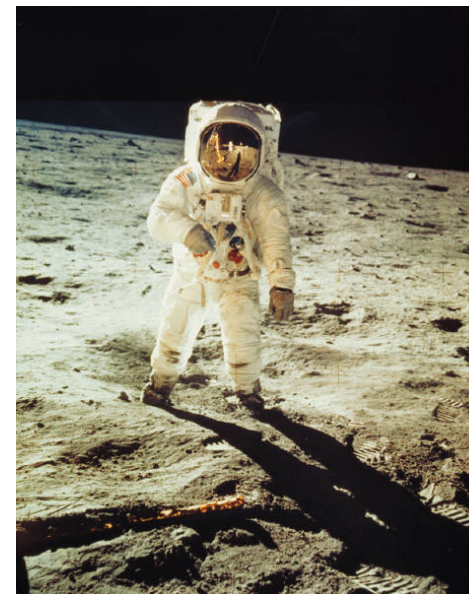
- 分析可行性

- 如何进行可行性研究
- 如何选择好的项目



- 分析风险

- 风险管理应长期、持续进行，而非阶段性、一次性的任务
- 进行灾难及事故分析，以确定风险



# 干系人

- 干系人分析

- 找出所有干系人
- 分析其隶属于哪个世界



- 干系人举例

- **用户**—关心新系统特征和功能
- **设计师**—想要构造完美的系统，尽量重用已有的代码
- **系统分析师**—想要获取正确的需求
- **培训与用户支持人员**—确保系统可用和可管理
- **业务分析师**—想确保“我们做得比竞争对手好”
- **技术文档作者**—为系统准备用户手册及其他相关文档
- **项目经理**—希望按时、按预算、按目标完成项目
- **客户**—为新系统买单的人



# 需求的必要性

- 需求定义过程中，客户与干系人的参与是至关重要的
- 不明确的需求定义及随意的需求变更会产生以下问题

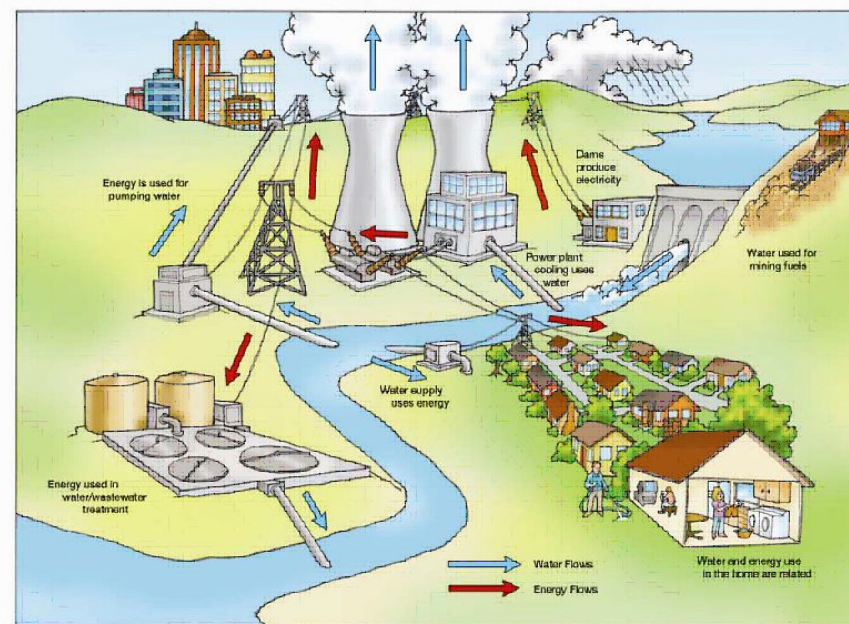
- 客户不满、纠纷
- 不切实际的估算、承诺
- 项目延期、超支
- 项目结束遥遥无期
- 项目团队精疲力竭



- 产品设计目标不明确
- 干系人参与不足
- 干系人之间缺少共识
- 画蛇添足
- 需求快速变化
- 变更管理不足
- 需求分析不足

# 需求内容来源

- 干系人
- 业务过程
- 组织规章制度
  - 组织规范、协议、技术标准
- 现有系统
  - 用户手册
  - 数据样本
  - 界面描述
  - 报告样本
  - 屏幕截图



# 干系人 (stakeholders)

- 干系人是任何和系统有关的人

资方、客户、系统用户、领域专家、项目研发团队

- 识别干系人的问题

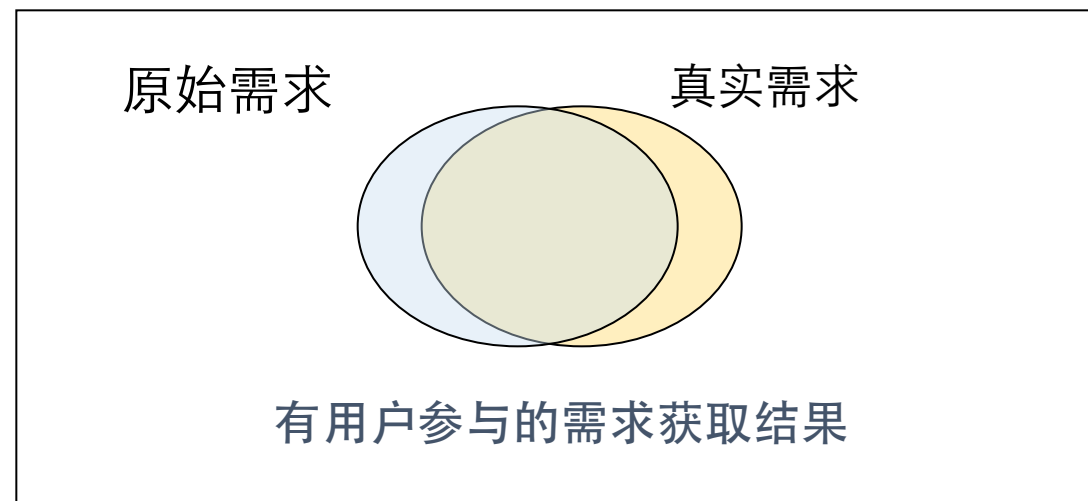
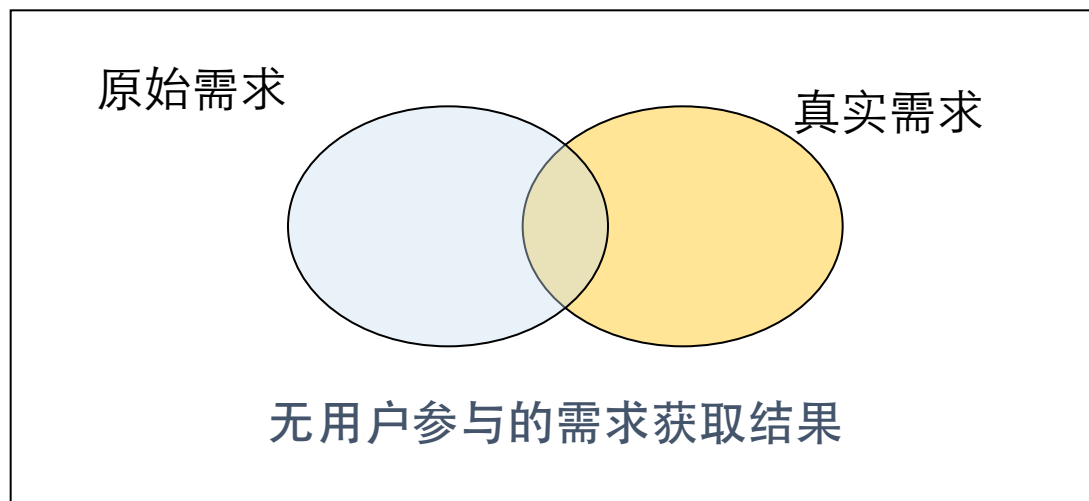
- 产品谁来用?
- 输入谁提供?
- 输出谁要?
- 谁监管?
- 影响谁?
- 奖励谁?
- 惩罚谁?





# 干系人

- 召集单个或一组干系人定义系统细节
  - 从客户或用户角度看的高质量系统
  - 提高用户满意度
  - 通过客户、用户参与，达到系统教育与培训目的



# 干系人

- 干系人的参与至关重要
  - 系统信息及内容的丰富
  - 系统信息质量的提升
  - 系统生产率提升
  - 客户对系统理解加深
  - 与客户达成共识
  - 客户更希望系统成功
  - 增强成功信心
  - 共同确定系统边界
  - 需求质量提升
  - 提高团队整体性





# 业务过程

- 对现有业务过程的分析有助于识别业务问题并加以改进
  - 找出并列举当前业务过程中的问题
  - 分析问题的本质（遗漏？不好用？新需求？）
  - 分析改进的机会
  - 分析改进的实质（自动化？流程改进？）



# 组织规章与制度

- 规章制度定义当前最佳实践
- 分析规章制度有益于确定业务规则和约束条件
  - 业务规则：描述对业务过程的要求，如系统支撑的业务过程的结构、控制、行为效果
  - 约束：对系统开发过程的管理限制，主要涉及经济、政治、技术和环境四个方面，具体包括项目资源、时间、目标环境及系统本身。
- 组织规章中往往还涉及过程自动化、工作流、关系、交互等内容



**KNOW THE  
RULES!**



# 现有系统

---

- 分析现有系统有助于了解未来系统的工作数据
  - 数据对象
  - 数据关系
  - 数据库结构与系统结构
  - 系统报告

