南京大学
NANJING UNIVERSITY

人机交互的软件工程方法
—— 交互式系统设计过程

主讲教师：冯桂焕
fgh@software.nju.edu.cn
2012年春季

为什么要学习设计过程？

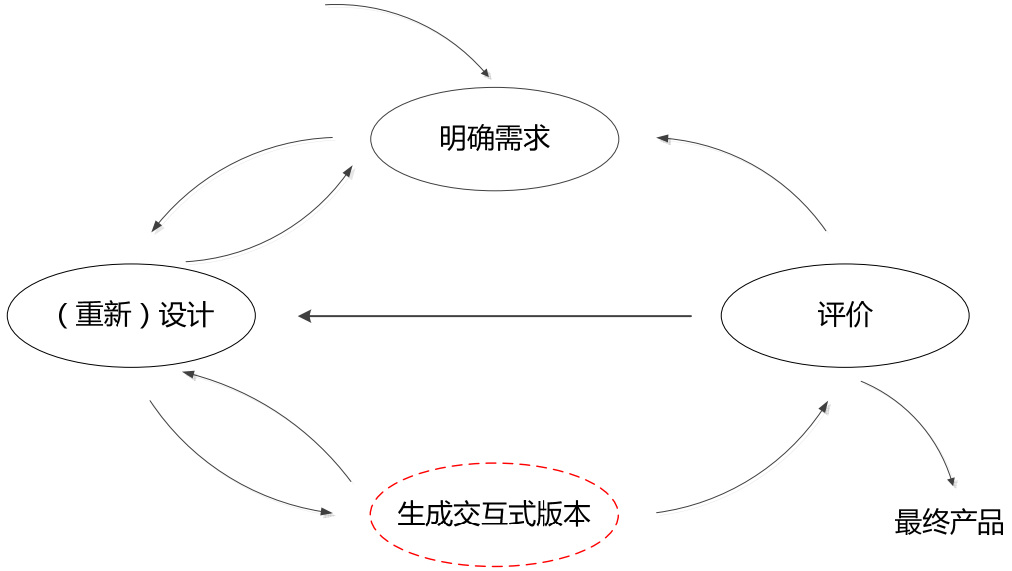
- 任何新产品的开发都不是一蹴而就的，而是由一系列基本开发活动所构成
  - 交互式软件系统的开发也不例外
  - 设计者50%的时间花费在用户界面的编码设计上
- 旨在了解
  - 软件设计中各个阶段的工作
  - 如何确保软件质量
  - 如何避免因前期问题而导致的后期开发损失
  - 软件工程不仅仅是写代码

设计过程的基本活动

- 标识用户需要并建立需求
  - 必须了解谁是目标用户
  - 交互式产品应提供哪些支持
  - 最基本的
- 开发满足需求的候选设计方案
  - 设计的核心活动
  - 概念设计和物理设计
- 构建设计的交互式版本
  - 评价设计的最佳方法就是让用户与产品交互
  - 不一定是可运行的软件版本
- 评估设计
  - 评估它的可用性和可接受性
  - 制定各种评估标准

各项活动间的关系

- 说明
  - 不是所有的交互式产品都必须采取这个开发过程
  - 代表了交互设计领域的实践经验



关键特征

- 以用户为中心
  - 人机交互领域的一个核心观点
- 稳定的可用性标准
  - 有助于设计人员选择不同的候选方案
  - 并在产品开发过程中随时检查
- 迭代
  - 设计人员不可能一次就找出正确的解决方案
  - 利用反馈来改进设计

设计过程中的问题

- 用户的选取
  - 用户：直接与产品交互、以期完成某个任务的人？
  - 当事人（stakeholders）：被系统影响的、而且对系统需求有直接或间接影响的个人或机构
    - 举例：电子日历
    - 你，与你约会的人；你要记住生日的亲友；甚至是生产记事本的公司
  - 不建议让所有的当事人都参与开发
    - 应该了解产品会造成多大范围的冲击

- 需求的确定
  - 错误：简单地问：“你需要什么”？
  - 正确：必须理解用户的特征和能力，了解他们想要达到什么目标，以及可满足的支持
    - 用户的能力、特征将影响产品的设计
    - 新产品的用户和有代表性的任务较难确定
  - 现状
    - 开发人员往往倾向于创建自己想要的产品或者是过去开发过的类似产品
  - 当前或过去的行为是对未来行为的很好启示

- 候选设计方案
  - 个别设计人员的才干、创造力？
    - 有时候是这样
  - 正解：候选方案来自考虑其他相似的设计
    - 竞争对手的产品
    - 类似系统的早期版本
    - 甚至是完全不同的东西
    - “所谓‘专家’，指的是那些能够从先前的经验中找到正确灵感并应用于当前工作的人”
  - 注意：不应局限于已有系统的限制

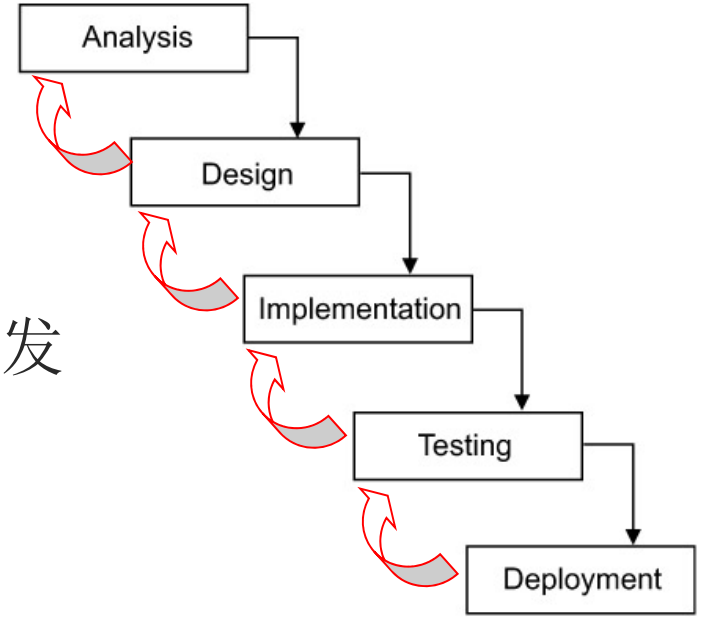
- 最终方案的决策
  - 用户与产品的交互方式是设计的推动力
    - 把注意力集中在可见、可测量的外部行为上
    - 例如，检索数据库（或网页）时，响应时间是4秒
    - 产品内部的工作详情只有在牵涉外部行为时才是重要的
  - 正解（以用户为中心）
    - 让用户和当事人与各种方案相交互
      - 听取他们的体验、偏好和改进建议等
    - 制定明确、无歧义的质量基准
      - 不明确：系统应能够快速响应
      - 明确：系统响应时间不应超过1秒

设计过程的生命周期

- 1960s~1970s
  - 大部分系统和商业中数据处理应用程序相关，较少交互
  - 从用户的角度来看，有关可用性的问题不十分重要
- 1970s后期
  - 个人计算机获得巨大商业成功，系统存在更多交互
  - 对那些不期望知道系统如何设计的一些人来说，易于操作是产品成功的关键
- 生命周期模型都是现实的简化表示，是现实的一种抽象
- 软件工程领域
  - 瀑布模型、螺旋模型、动态系统开发方法
- 人机交互领域
  - 星型生命周期模型、可用性生命周期

瀑布模型

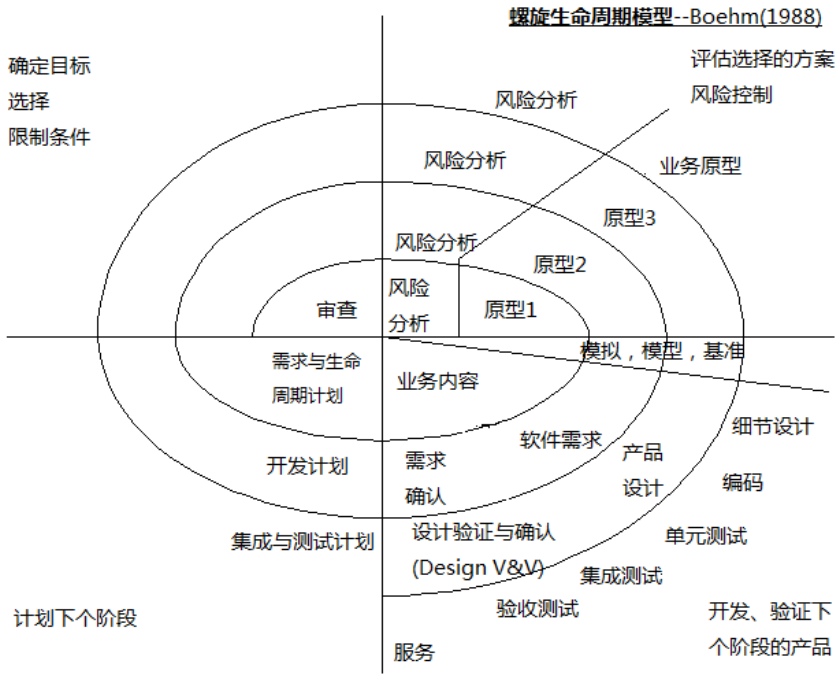
- 贡献
  - 把分析和编码作为了模型的两个主要组成部分
  - 第一个得到广泛承认的模型



- 缺点
  - 以文档为中心，用户难以理解
  - 不适用于交互式软件产品的开发
  - 假设需求是不变的

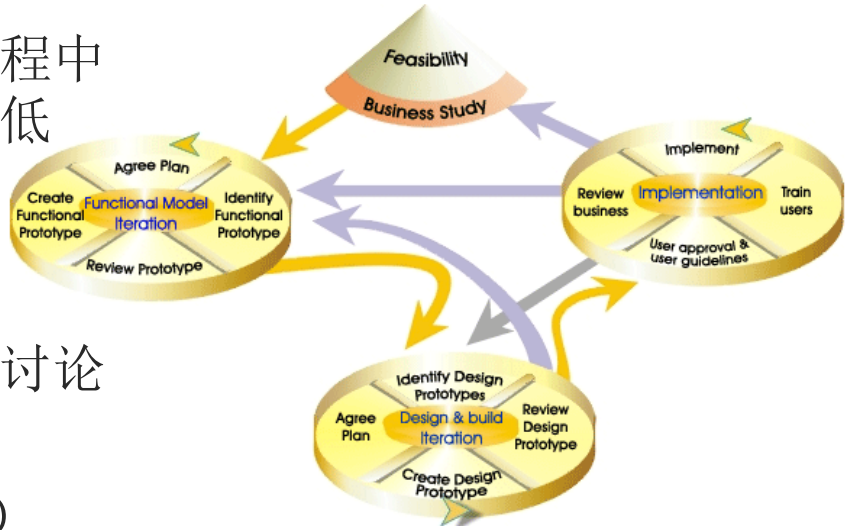
螺旋模型

- Boehm指出，使用这个模型设计的系统，在所有的项目中至少增加了50%的生产力
  - 以降低风险为中心
  - 引入了“迭代”的思想
- 缺点
  - 过于复杂，客户难以掌握
  - 风险评估需大量信息
  - 未发现的风险是棘手的



快速应用开发

- 出现于1990s，采用以用户为中心的方法
  - 目的是要把由于需求在开发过程中不断变化而导致的风险降至最低
- 两个关键特征
  - 周期时限大致为6个月
  - 用户和开发人员共同参与专题讨论，确定需求
- 动态系统开发方法（DSDM）
  - 可行性研究、商务研究、开发功能模型（迭代）、设计及构建（迭代）、实现



原型法

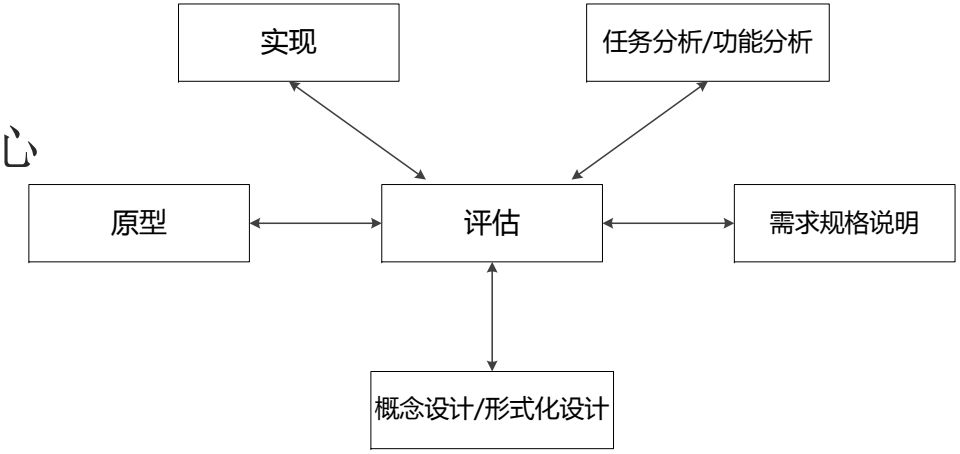
- 允许设计者从一个屏幕到另一个屏幕来讨论诸如外观和感觉、范围、信息流等设计问题
- 抛弃原型模型
  - 客户不清楚项目范围和客户不能精确描述项目需求时
- 进化原型模型
  - 随着项目的进展不断进化，直到实现最终产品
- 优点
  - 易于用户提供反馈；
  - 减少了开发时间和成本；
  - 用户参与到开发过程中。

传统软件生命周期小结

- 适应于原则性强的场合
  - 如果从开始就知道想要生产什么，则为了达到目标，可以构建我们自己的设计方法
- 一个交互式软件系统的所有需求在开始时是无法确定的
- 如何使得系统更有用处？
  - 建造系统
  - 观察和评估与用户的交互

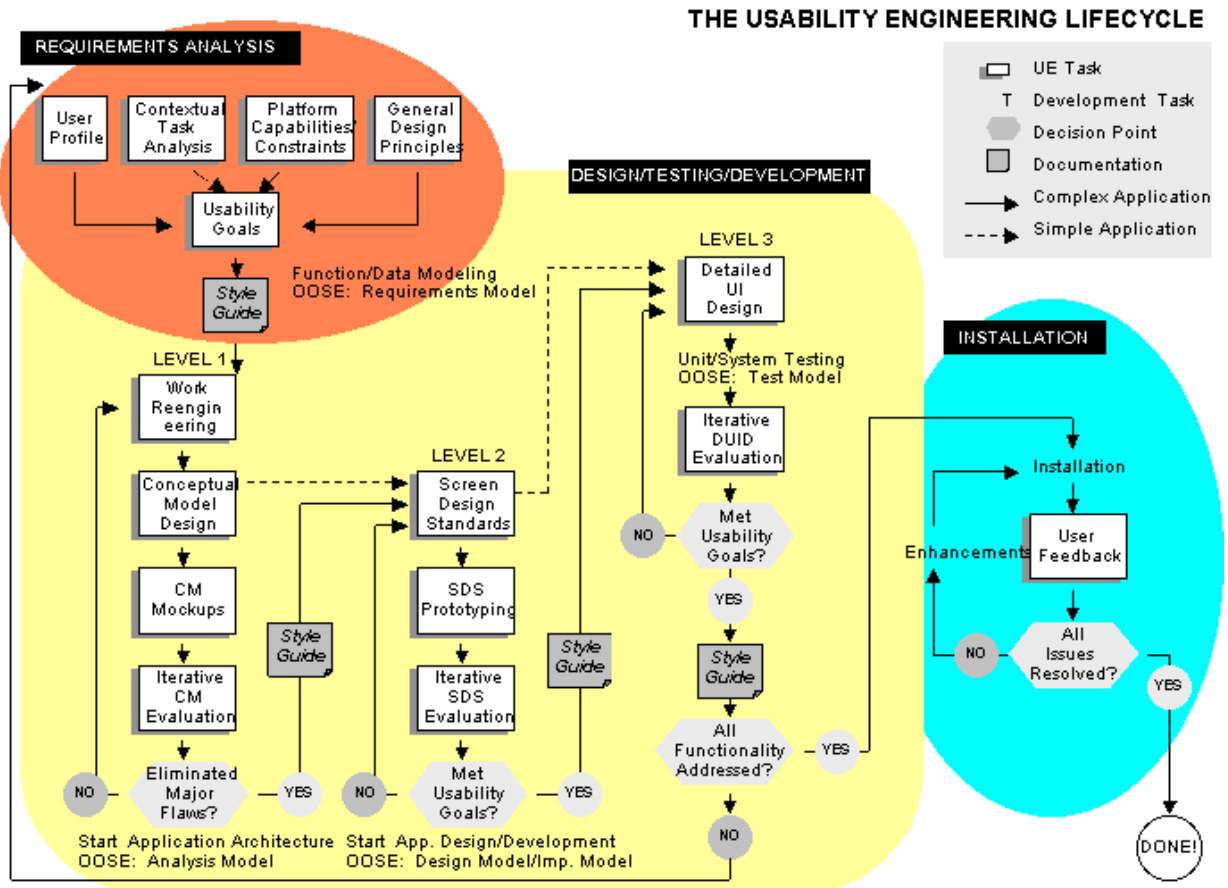
星型生命周期模型 - HCI领域

- 分析模式
  - 自顶向下、组织化、判定和正式化，它是从系统到用户的方法
- 合成模式
  - 自底向上、自由思考、创造性，这是由用户至系统的方法
- 特点
  - 没有指定任何活动次序
  - “评估”是这个模型的核心
  - 源于开发人员的实际经验



可用性工程生命周期 – HCI领域

- Deborah Mayhew，1999
  - 体现了可用性工程的总体概念
  - 详细描述了如何执行可用性任务
  - 说明了如何把可用性任务集成到传统的软件开发生命周期中



设计过程的管理

- 据统计，未实现目标的软件项目高达60%
  - 主要问题：开发者和他们的商业委托人或者开发者和他们的客户之间缺乏沟通
- 传统软件工程方法
  - 不足以提供清晰的进程来研究用户
- 以用户为中心的设计
  - 在开发过程中产生尽可能少的错误
  - 开发费用少和维护成本低
  - 易于学习、执行速度更快
  - 鼓励用户探索
  - 应尽早使用

合理的以用户为中心的交互设计，LUCID

- 预想
  - 发展清晰、共享的产品场景，使概念草案具体化
- 发现
  - 研究用户以决定高端的用户需求、术语和智力模型
- 设计基础
  - 发展概念设计，创造关键屏幕画面原型
- 设计细节
  - 将高端设计加以充实，形成完备的详细说明书
- 构建
  - 通过回顾和后期改变管理方式来支持生产过程
- 发布
  - 通过有力的推广来支持用户向新产品的过渡；进入最后的可用性测试

- 特色
  - 将注意力集中在关键屏幕画面原型
    - 促进用户的早期参与，并为项目的进展创造动力
  - 快速原型法是安排进度和预算的关键
    - 依赖于用户界面开发工具
- 目标
  - 通过加强对开发、交付和评审等方面的重视，强调了可用性工程在软件开发过程中的地位，促进有序的开发过程

小结

- 设计的四个主要活动
- 设计的三个关键特征
- 软件生命周期
  - 传统软件生命周期模型
  - HCI领域的生命周期模型
- 设计的管理