

# 软件过程模型



# 知识回顾

- 软件的概念：文档、数据和程序的集合
- 软件的特点：逻辑产品、需求多变、设计开发、缺陷隐蔽性
- 软件生命周期：经历多个不同的阶段
- 闭源软件和开源软件：实践状况及优势

- ✓ 软件 ≠ 程序；开发软件 ≠ 编写程序；
- ✓ 软件是如何开发出来的？



# 内容

## 1. 何为软件过程模型

- ✓ 基本概念和特点

## 2. 有哪些软件过程模型

- ✓ 有什么类别，各有什么优缺点

## 1. 如何来选择软件过程模型

- ✓ 软件过程模型的选择方式和策略



# 1.1 软件开发的特点

## □基于智力的协作过程

- ✓智力活动：基于逻辑思维来构造软件
- ✓交流协作：软件工程师、用户间的交流和讨论

## □软件项目内在复杂性

- ✓介入的人多、考虑的内容多、产生的制品多
- ✓不同要素间存在关联

## □循序渐进的开发过程

- ✓开展有序的开发活动，如编码、分析、设计
- ✓体现了工程的思想：按步骤、分阶段

按照什么样的过程来有序地开发软件？



# 1.2 软件过程

## □过程(Process)

- ✓ **活动**：明确要做哪些事情，包括具体的活动
- ✓ **关系**：活动间存在逻辑关系，如依赖和先后次序
- ✓ 示例：考研的过程

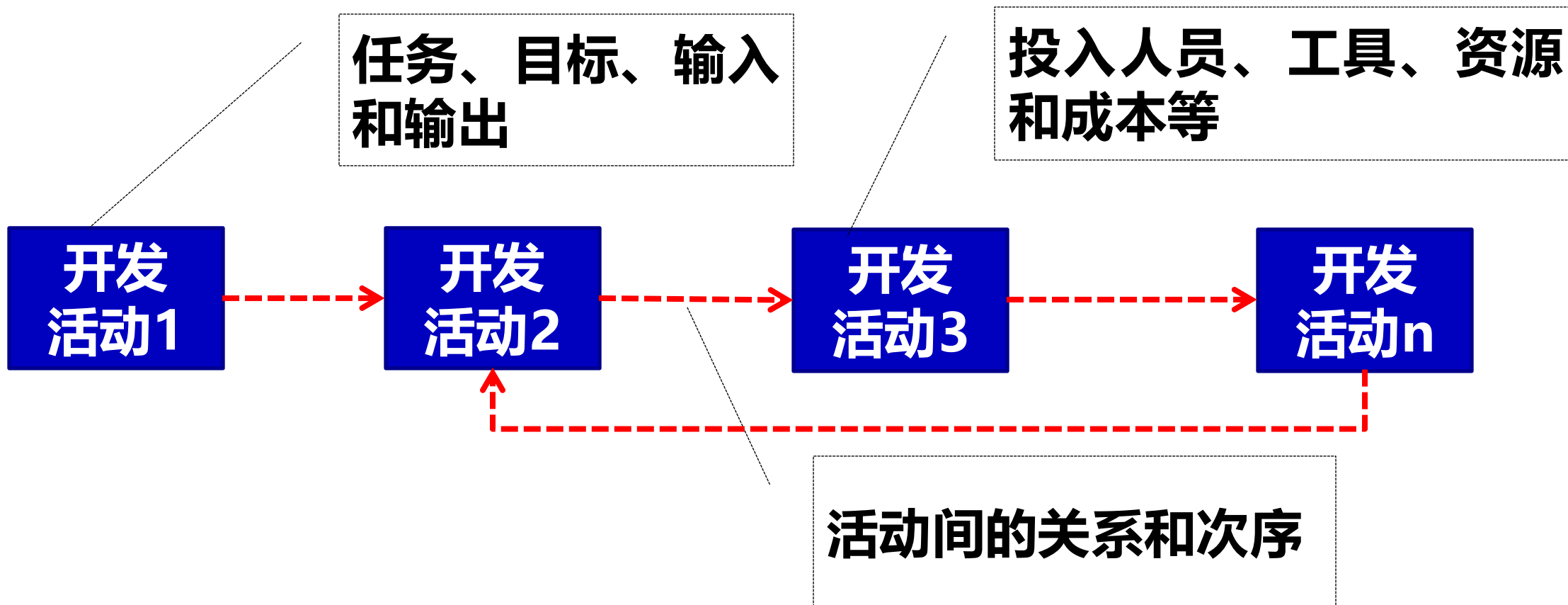
## □软件过程(Software Process)

- ✓ 按照项目进度、成本和质量要求，遵循用户需求，开发和维护软件、管理软件项目的一系列**有序软件开发活动**
- ✓ 软件开发活动：**技术活动和管理活动**

# 1.3 软件过程模型

## □ 软件过程模型(Software Process Model)

- ✓ 定义了软件开发的具体活动以及活动间的逻辑关系



# 内容

## 1. 何为软件过程模型

✓ 基本概念和特点

## 2. 有哪些软件过程模型

✓ 有什么类别，各有什么特点和优缺点

## 3. 如何来选择软件过程模型

✓ 软件过程模型的选择方式和策略





## 2.1 软件过程模型的产生背景



**作坊式的  
个人创作**

**软件工程产生之  
前的软件开发**

➤ 聚焦于编写代码

➤ 依靠个体技能，缺乏合作

➤ 关注时空利用，精雕细琢

➤ 程序规模小且功能单一

➤ 无系统性方法和标准流程

# 软件过程模型的产生背景



## ❑ OS/360 超大型软件项目(1960s初):

- ✓ 通用系统, 支持多道程序, 最多可同时运行15道程序
- ✓ 软件工程师超2000人, 花费超5亿美元, 工作量超5000人年

## ❑ 有史以来最可怕的软件开发泥潭

- ✓ Brooks, 《人月神话》The Mythical Man-Month作者、图灵奖获得者



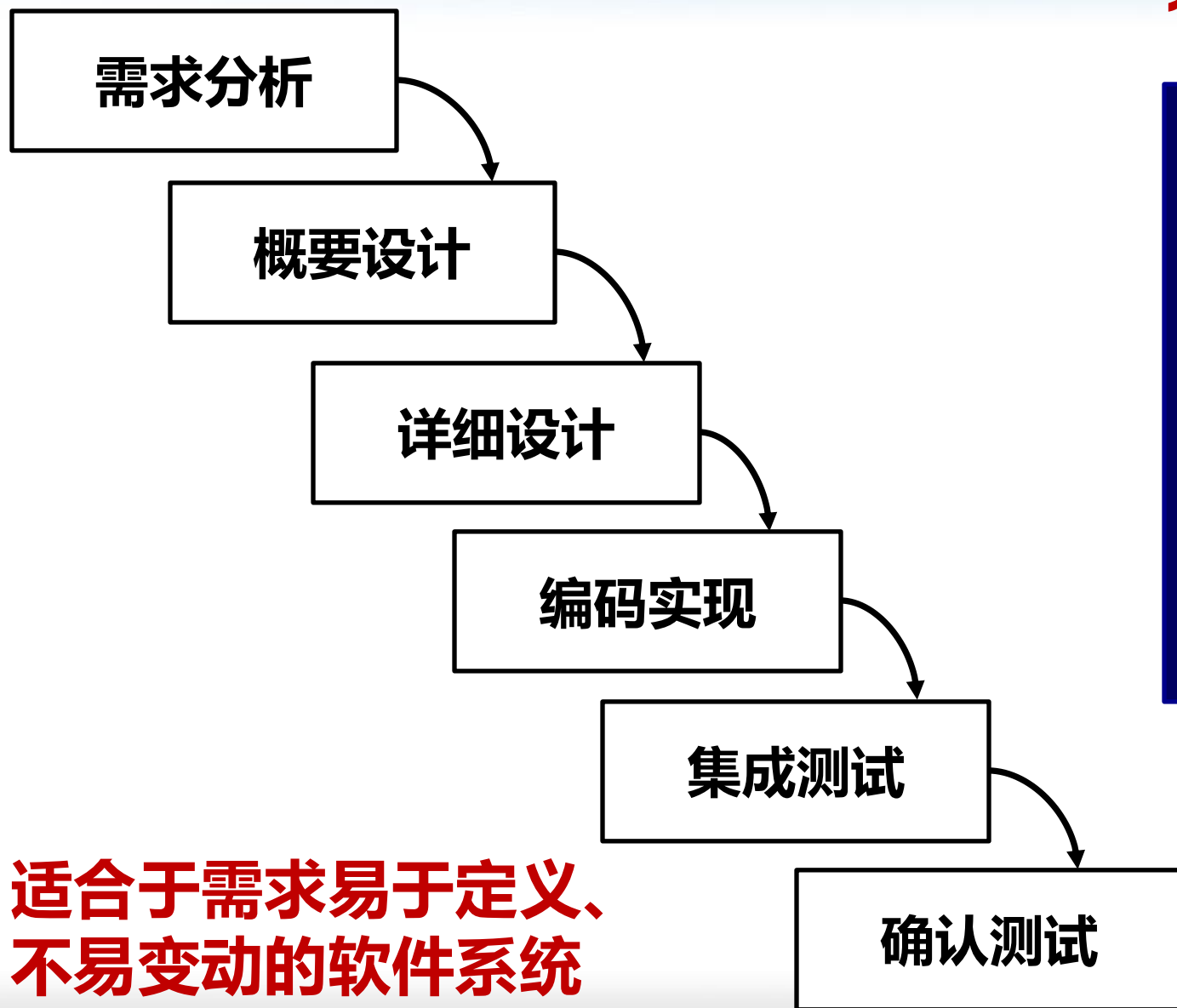
# 典型的软件过程模型

- 瀑布模型
- 增量模型
- 迭代模型
- 原型模型
- 螺旋模型
- 基于构件的过程模型
- UP模型

- 需要系统、规范性的软件过程模型的指导
- 每种软件过程模型有其各自的特点和适用的场所

## 2.2 瀑布模型(Waterfall Model)

1970提出的第一个软件过程模型



### ➤ 特点

- ✓ 与软件生命周期相互一致
- ✓ 每个活动结束后需要评审
- ✓ 相邻活动间存在因果关系

### ➤ 优点

- ✓ 简单，一目了然，易理解、掌握、应用和管理

适合于需求易于定义、  
不易变动的软件系统

# 需求分析(Requirement Analysis)

## □活动

- ✓任务：**定义软件需求**，包括功能、非功能需求
- ✓关注点：**要做什么？(What, Problem)**
- ✓层次和视角：用户角度，仅描述问题和需求

问题是什么？

## □方法

- ✓依据：用户的期望和要求
- ✓不断与用户进行交流和商讨，抽象、问题分解、多视点等技术

## □产出

- ✓**软件需求模型、软件需求文档、软件确认测试计划**
- ✓**文档类的软件制品**

# 概要设计(Architecture Design)

## □活动

- ✓任务：**建立软件总体架构、制定集成测试计划**
- ✓关注点：**软件高层设计？(How, Solution)**
- ✓层次和视角：宏观、全局、整体、战略性

问题如何解决？

## □方法

- ✓依据：软件需求文档
- ✓自顶向下, 逐步求精, 抽象, 模块化, 局部化, 信息隐藏 .....

## □产出

- ✓**软件概要设计模型、软件概要设计文档、软件集成测试计划**
- ✓**文档类的软件制品**



# 详细设计(Detailed Design)

## □活动

- ✓任务：**设计模块内部细节(算法、数据结构)**，制订单元测试计划
- ✓关注点：**详细设计? (How, Solution)**
- ✓层次和视角：微观、局部、细节性

**问题如何解决?**

## □方法

- ✓依据：概要设计文档、软件需求文档
- ✓高质量的软件设计原则，如单入口单出口

## □产出

- ✓**软件详细设计模型、软件详细设计文档、单元测试计划**
- ✓**文档类的软件制品**

# 编程实现(Implementation)

实际解决问题

## □活动

- ✓任务：编写程序代码并进行单元测试和调试
- ✓关注点：如何最终做出这个东西？(How, Code)
- ✓层次和视角：最终的实现代码

## □方法

- ✓依据：软件概要和详细设计文档、单元测试计划
- ✓采用某种程序设计语言(如C、C++、Java)

## □产出

- ✓经过单元测试的源程序代码
- ✓程序类的软件制品



# 集成测试(Integration Test)

## □活动

- ✓任务：**组装软件模块并进行测试以发现问题**
- ✓关注点：**集成后软件中的缺陷 (Bug)**
- ✓层次和视角：自底向上组装、全局

问题解决如何？  
软件有缺陷吗？

## □方法

- ✓依据：软件概要设计文档、软件集成测试计划
- ✓软件集成测试工具

## □产出

- ✓**经过集成测试、修复缺陷的源程序代码，集成测试报告**
- ✓**数据、文档和代码类的软件制品**

# 确认测试(Validation Test)

## □ 活动

- ✓任务：**测试软件是否满足用户需求**
- ✓关注点：**软件在满足用户需求方面是否存在缺陷**
- ✓层次和视角：从用户角度，聚焦需求是否得以正确实现

问题解决如何？  
软件有缺陷吗？

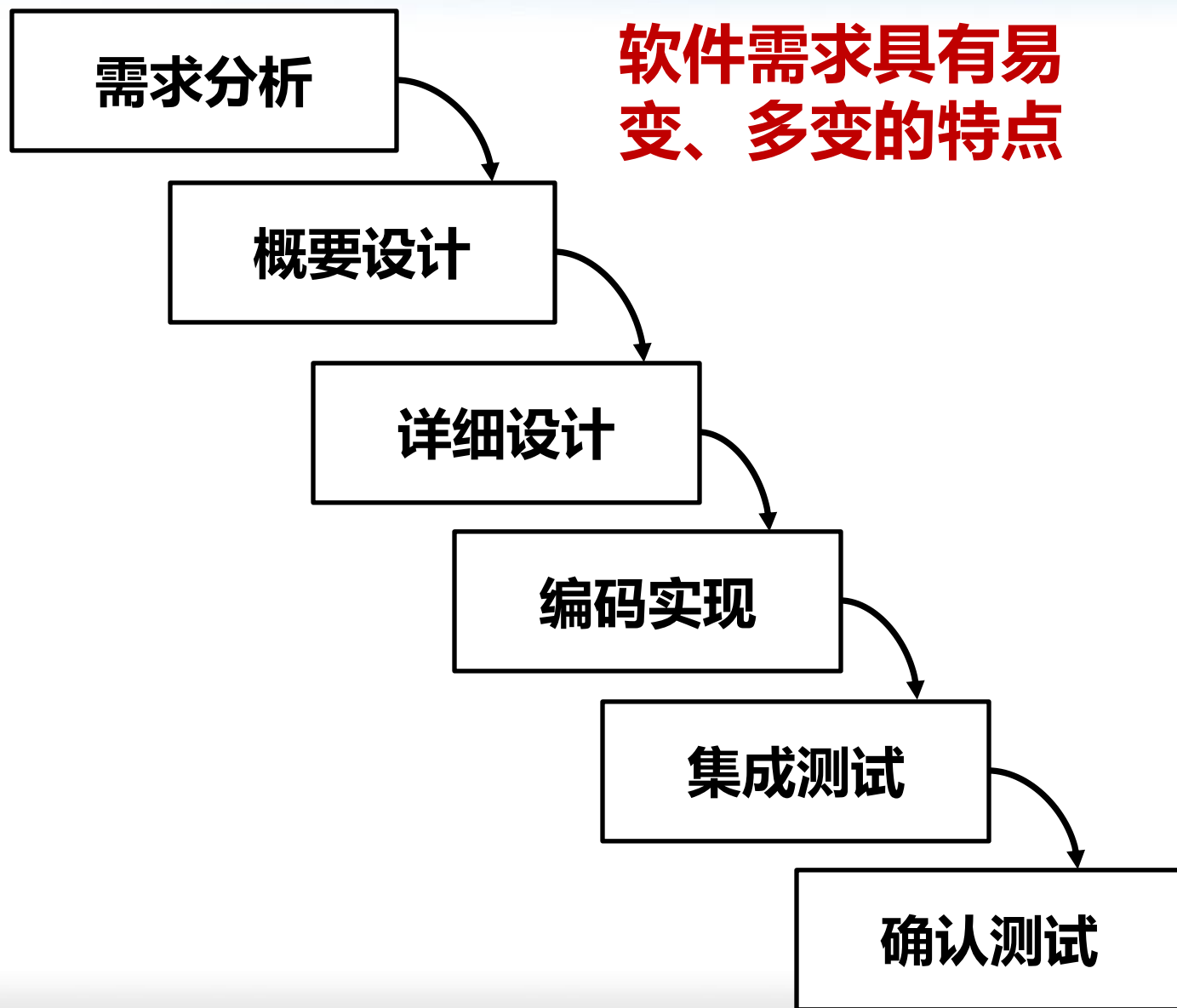
## □ 方法

- ✓依据：软件确认测试计划、软件需求文档
- ✓软件测试支撑工具

## □ 产出

- ✓**经过确认测试、修复缺陷后的代码，软件确认测试报告**
- ✓**数据、文档和代码类的软件制品**

# 瀑布模型的局限性?



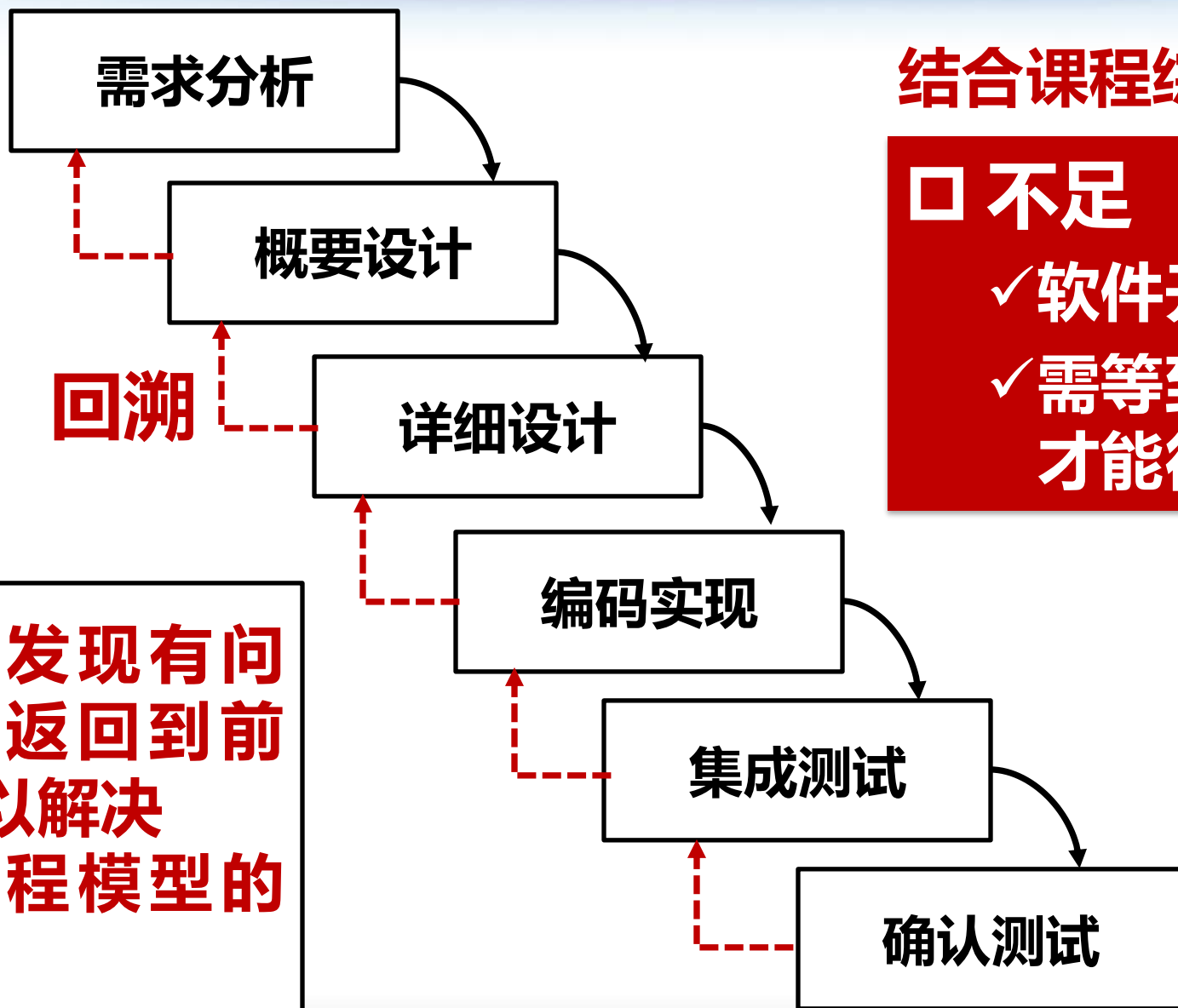
结合课程综合实践思考

□ 不足

- ✓ 需求确定，过于理想化
- ✓ 缺乏变通，难应对变化



## 2.3 改进的瀑布模型：带反馈和回溯



结合课程综合实践思考

□ 不足

- ✓ 软件开发处于动荡之中
- ✓ 需等到所有功能实现后，才能得到可运行软件



- ✓ 后期活动发现问题后，可返回到前面活动加以解决
- ✓ 提高了过程模型的灵活性

## 2.4 增量模型(Incremental Model)

需求分析

概要设计

详细设计

编码实现

集成测试

- 渐进式、增量式地实现软件功能
- 优点：渐进快速交付，并行开发，提高效率

增量1

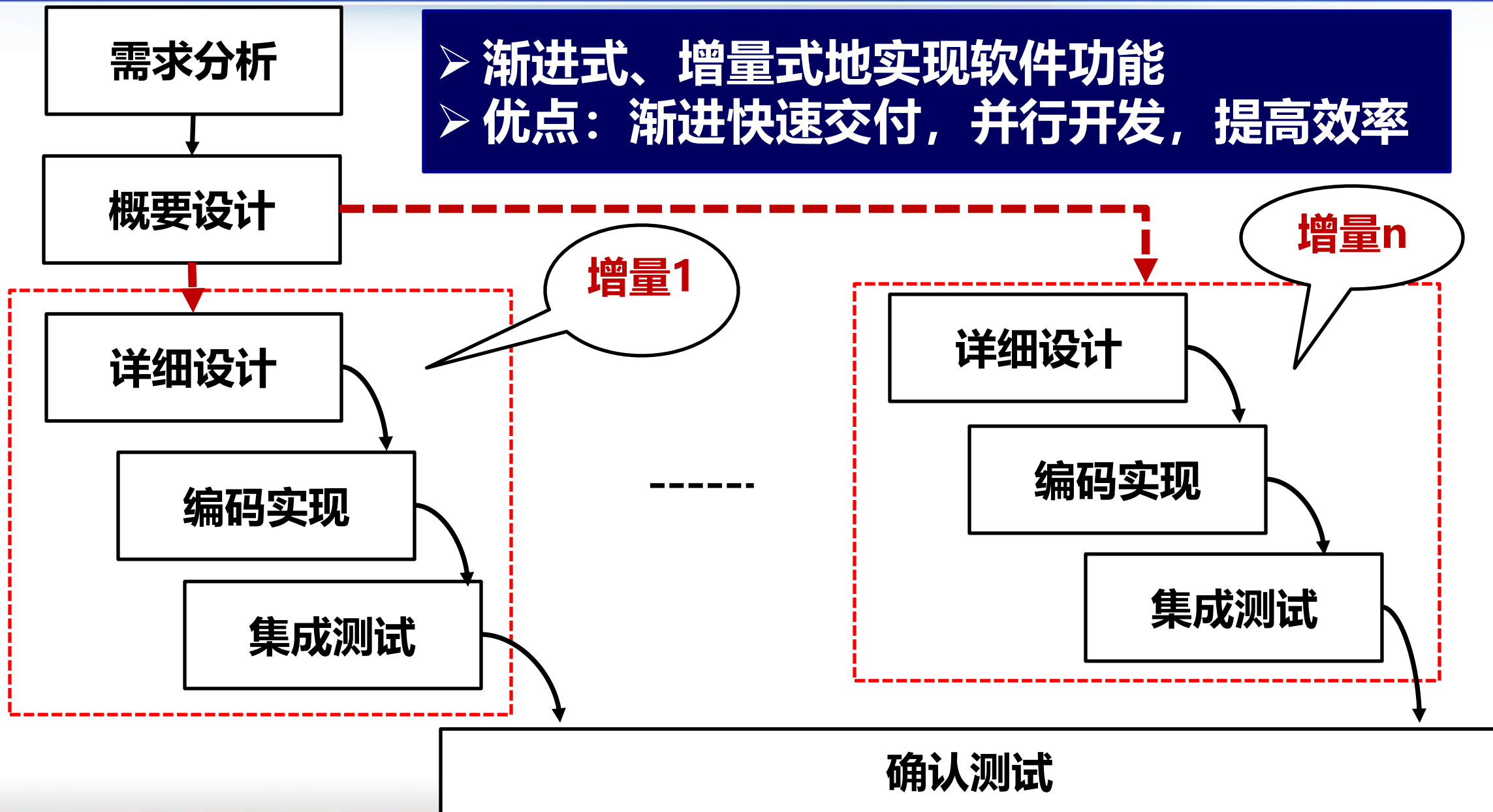
详细设计

编码实现

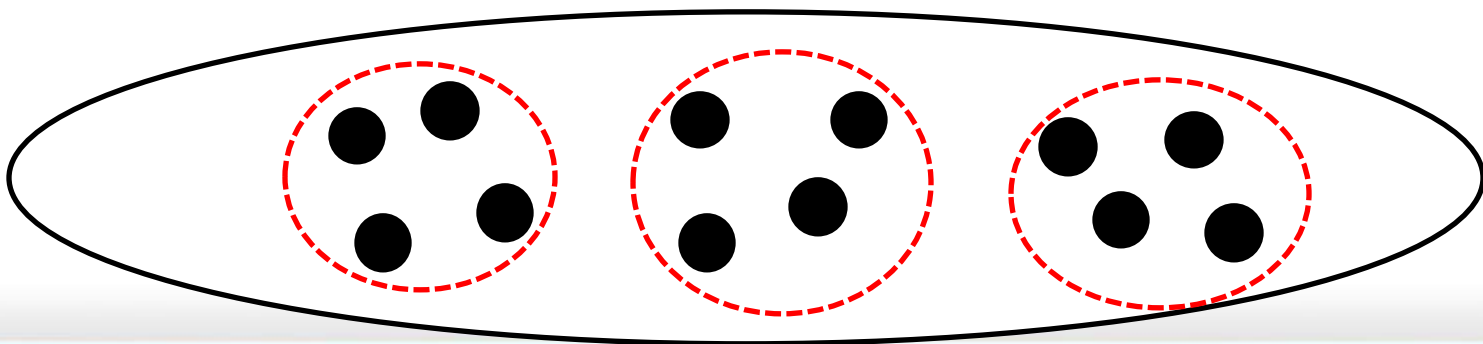
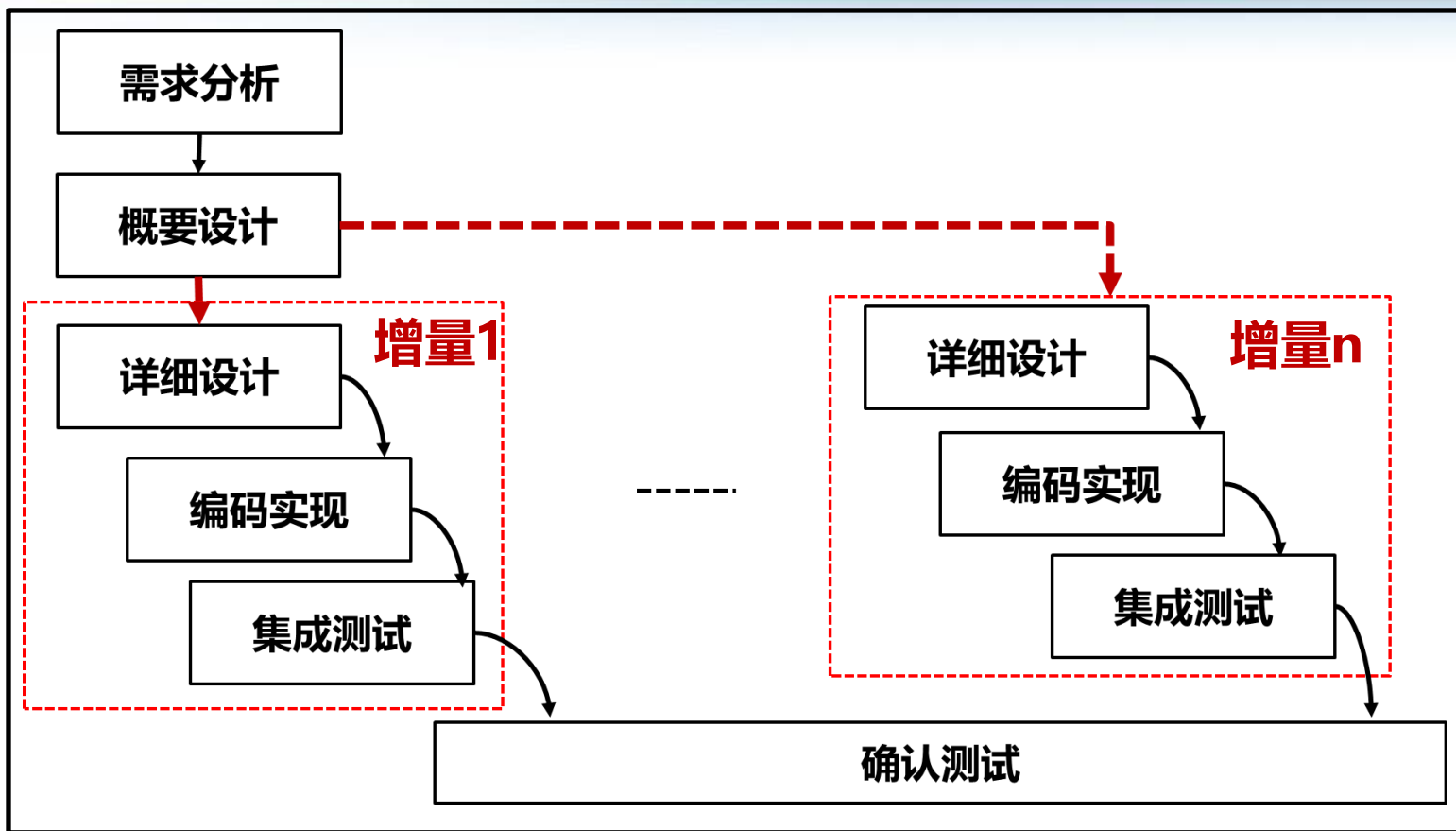
集成测试

增量n

确认测试



# 增量模型的局限性?



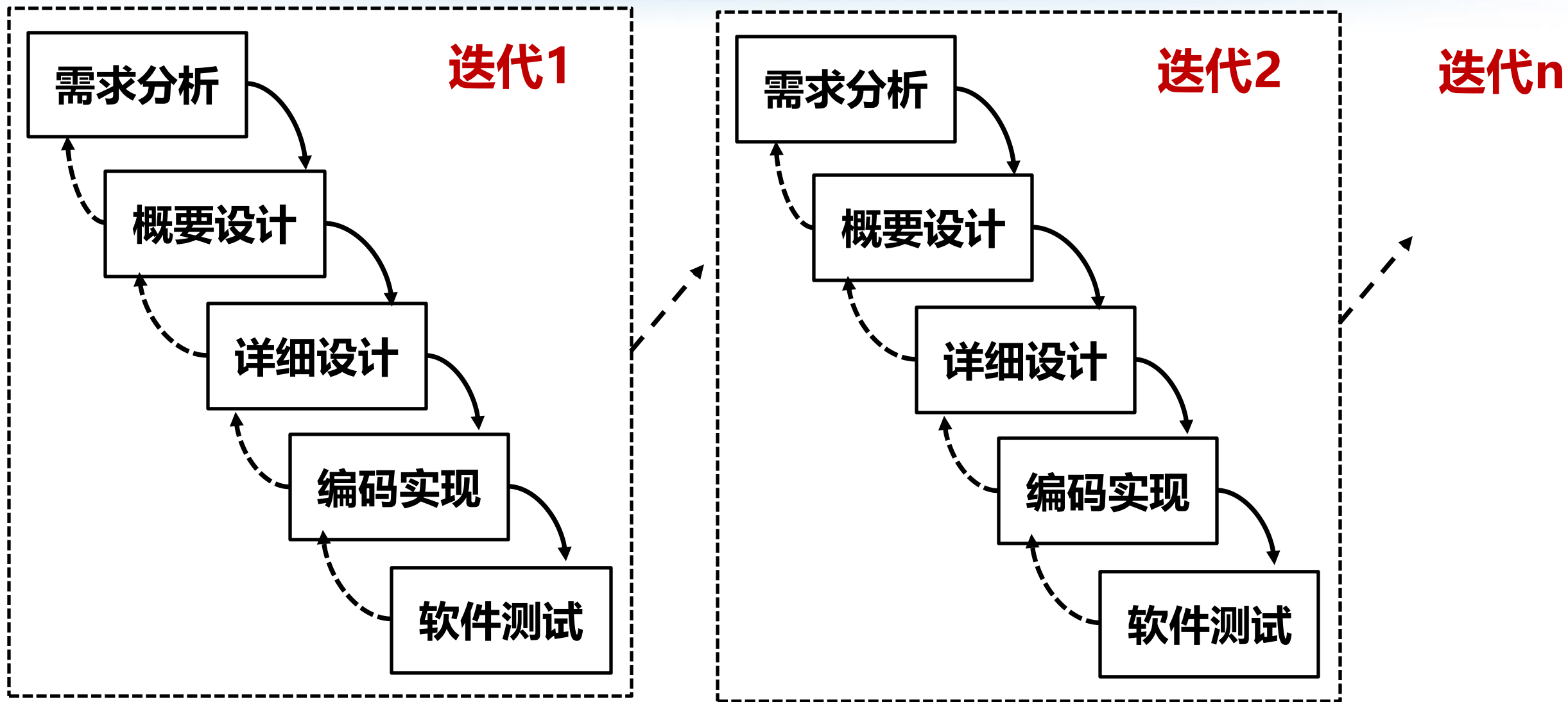
结合课程综合实践思考

□ 不足

✓ 软件需求可确定且不易于变化

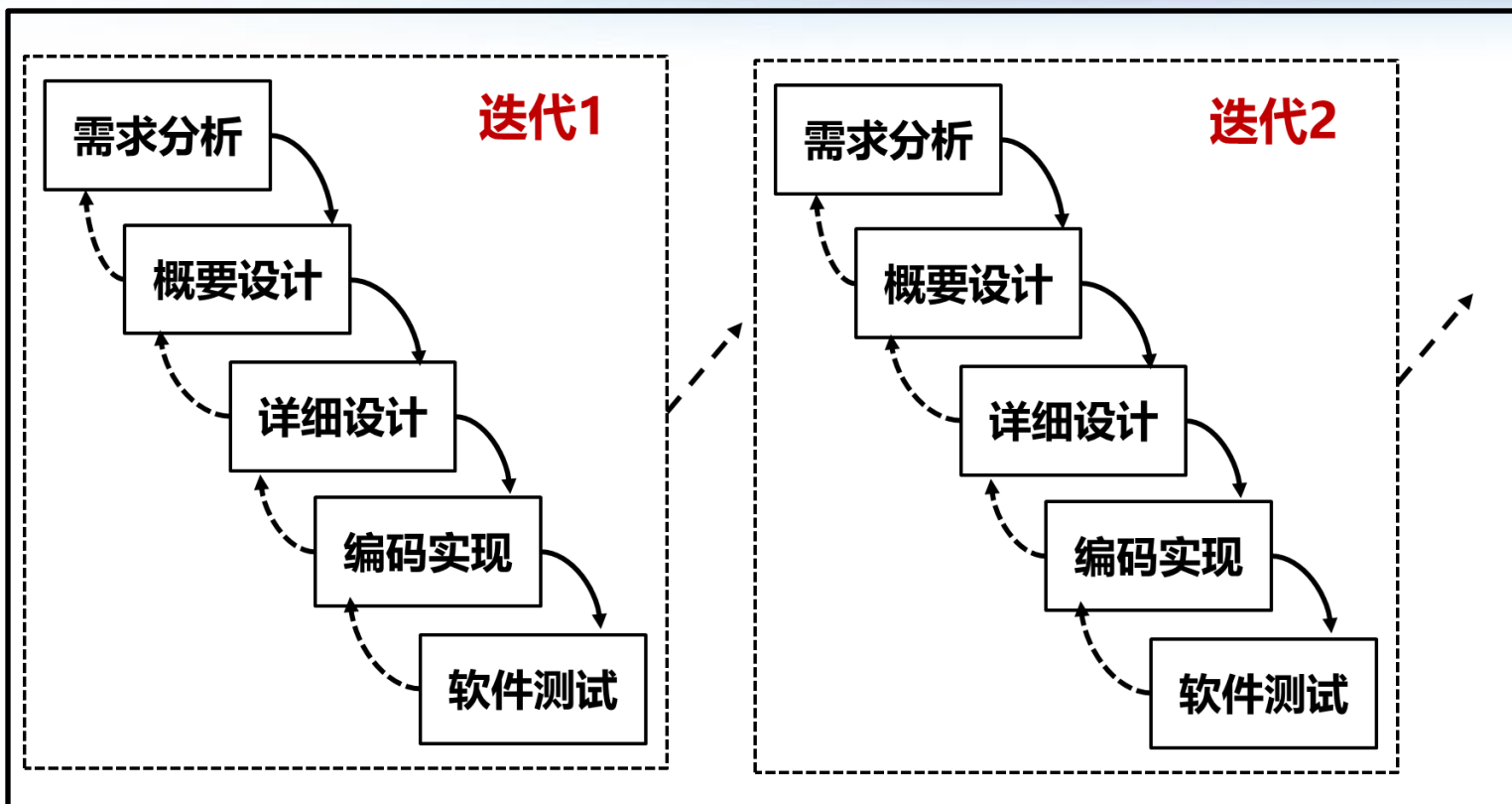


## 2.5 迭代模型(Iterative Model)

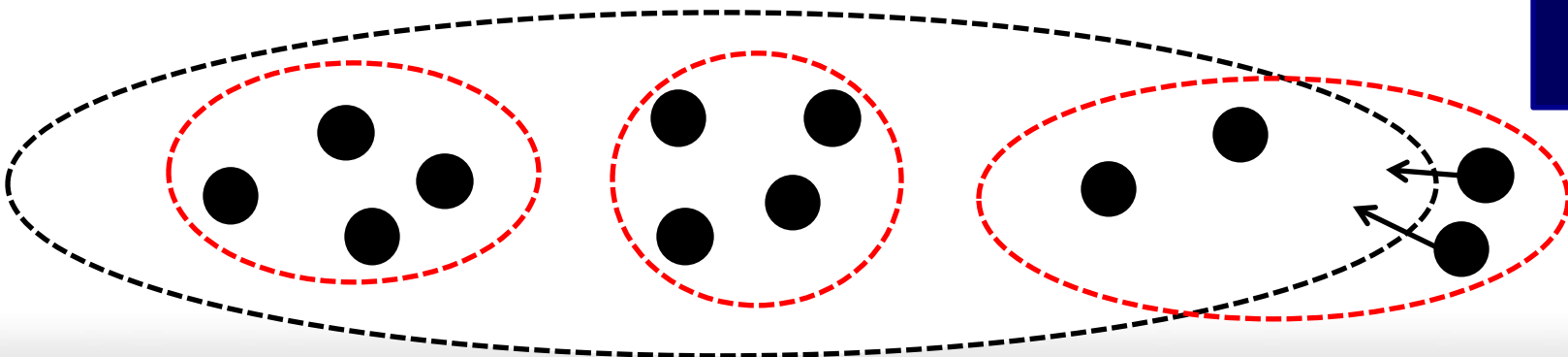


每次迭代完成部分可确定的软件需求

# 迭代模型的特点



- 每次迭代是一完整过程
- 体现了小步快跑的开发理念
- 适合需求难导出、不易确定且持续变动的软件





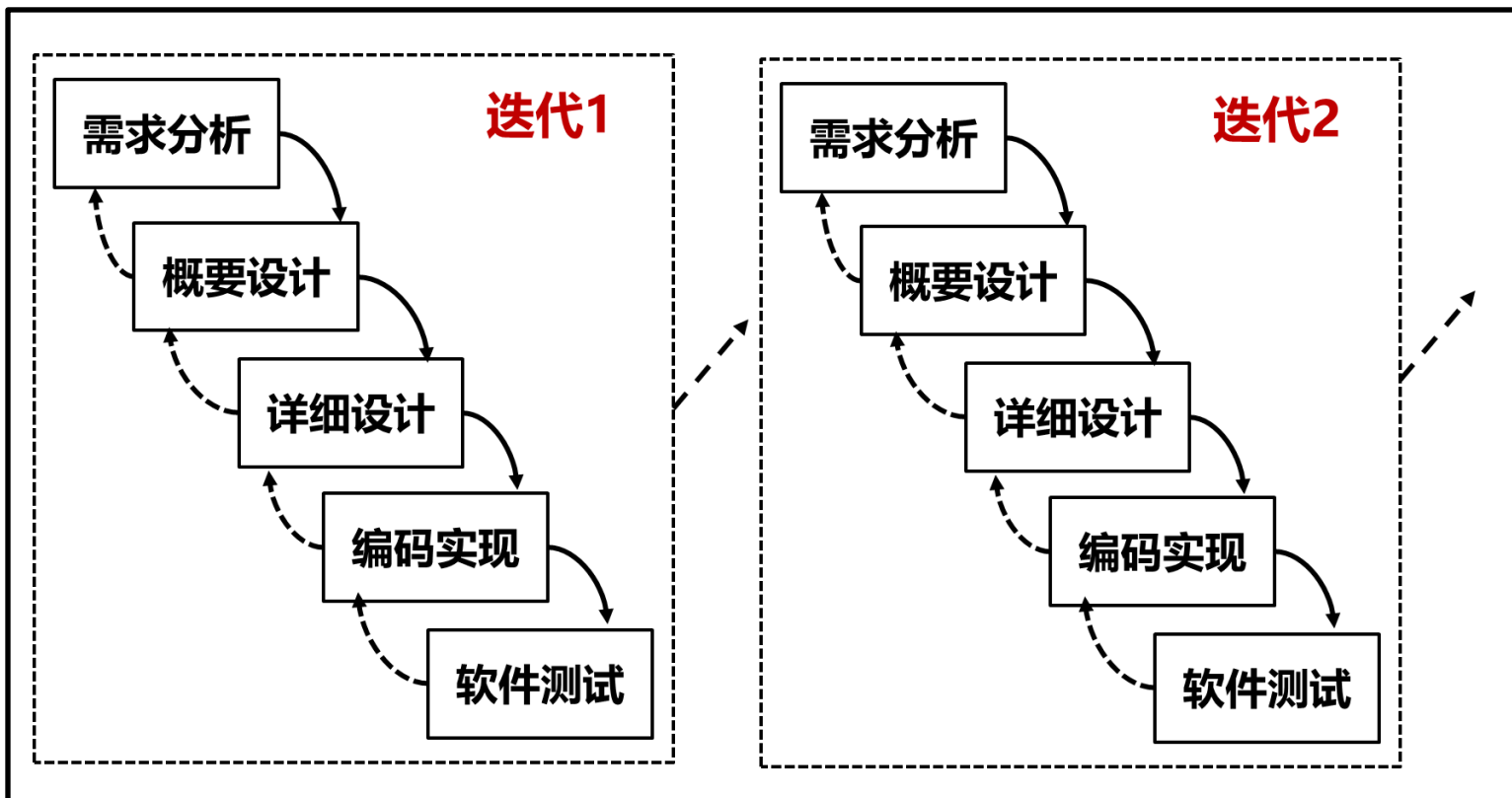
# 迭代模型的局限性?

结合课程综合实践思考

❑ 不足

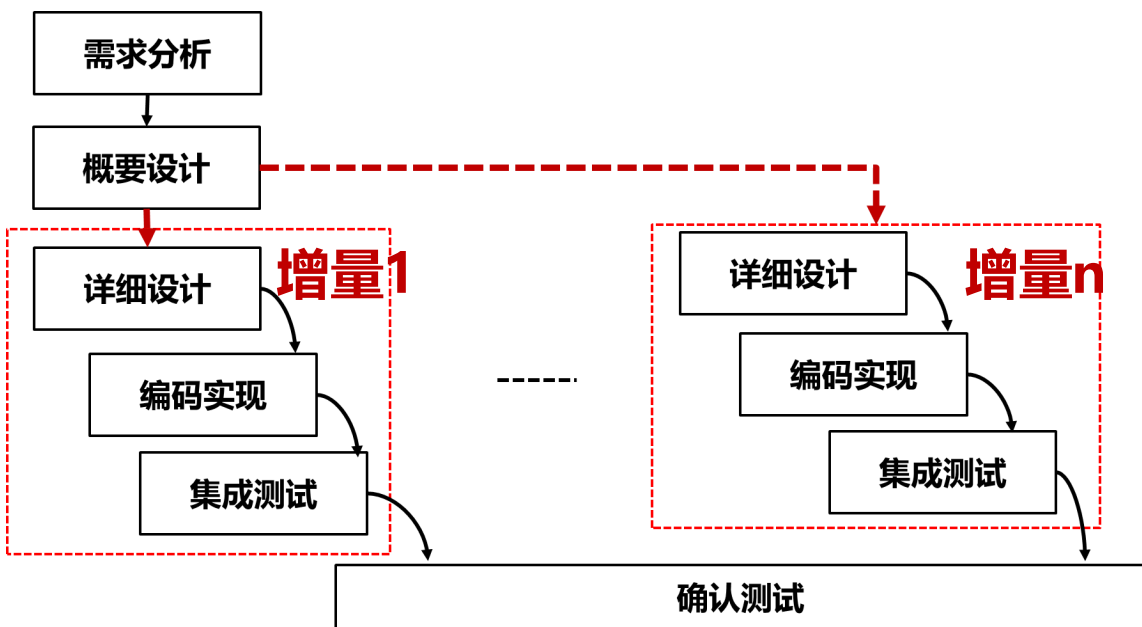
✓ 迭代多少次不确定

✓ 管理较为复杂

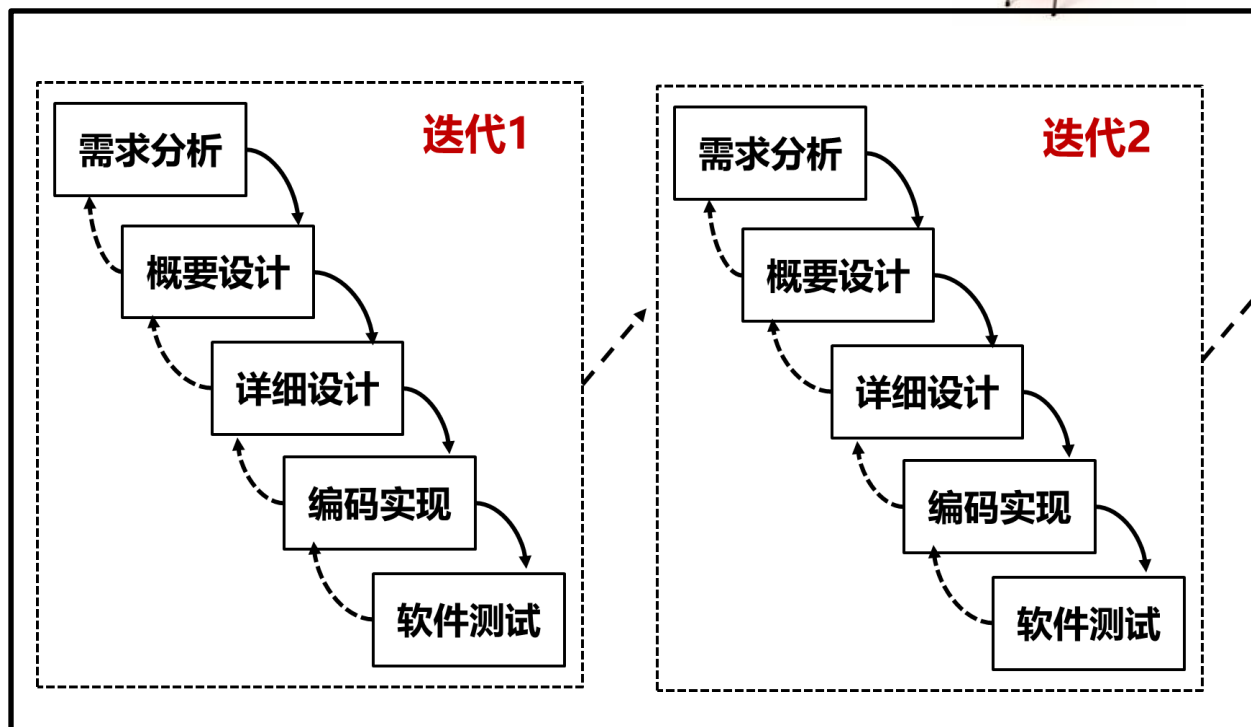


# 思考和讨论

## □ 增量过程模型与迭代过程模型有何区别？



增量模型



迭代模型

# 软件需求获取是一关键和瓶颈问题

## □软件需求非常关键

- ✓ 软件开发的基础、验收的依据

## □用户讲不清楚软件需求有哪些、是什么？

- ✓ 说不清、道不明
- ✓ 尤其当软件较为复杂和庞大之时

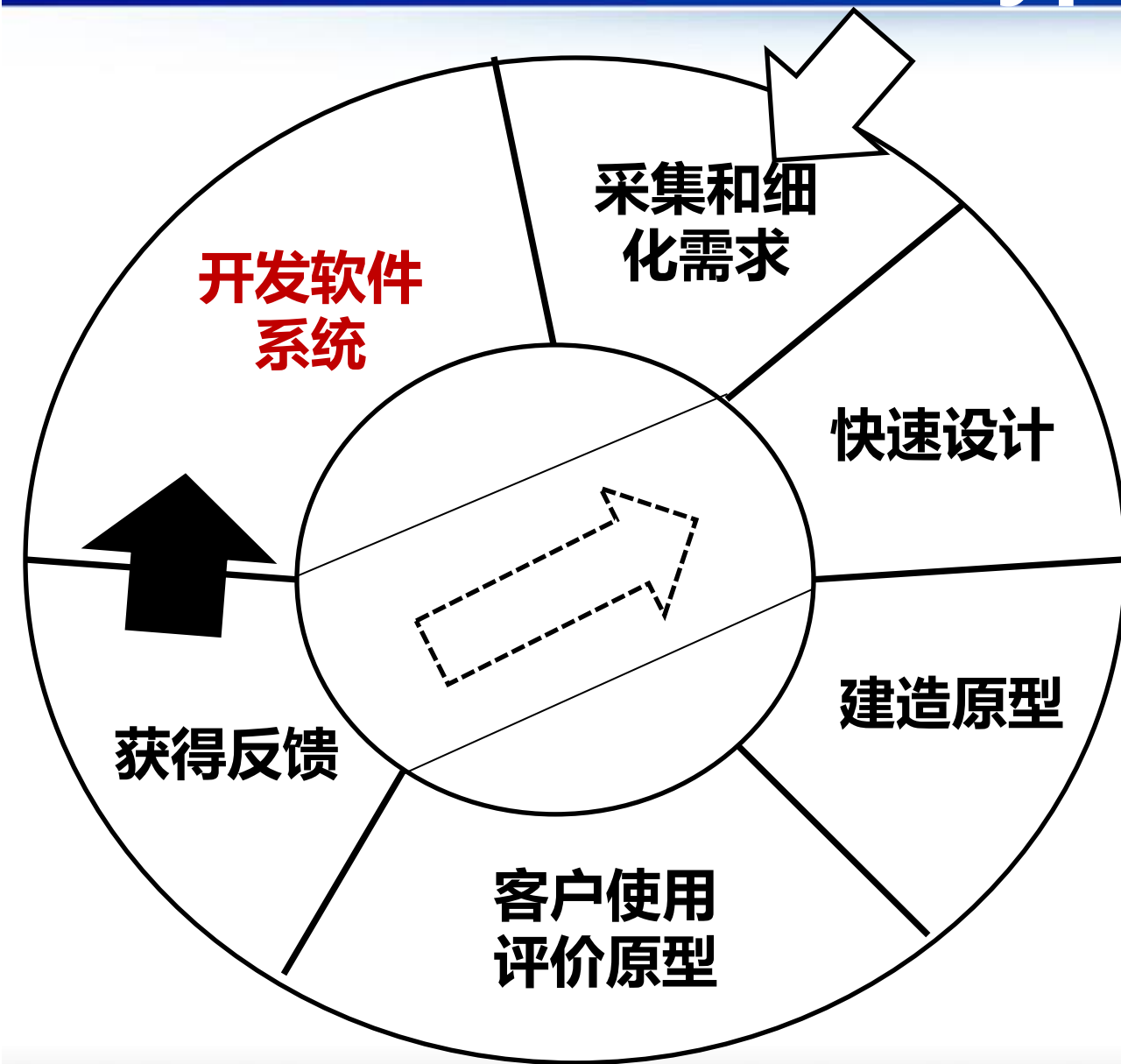
## □用户与软件工程师对软件需求理解存在偏差

- ✓ 对软件需求描述的歧义性、二义性、不准确等造成的
- ✓ “应该是这样的”、“实际是这样的”

**如果软件需求分析结果不正确、不完整、有歧义，会带来什么样的问题？**



## 2.6 原型模型(Prototype Model)



### □ 何为软件原型?

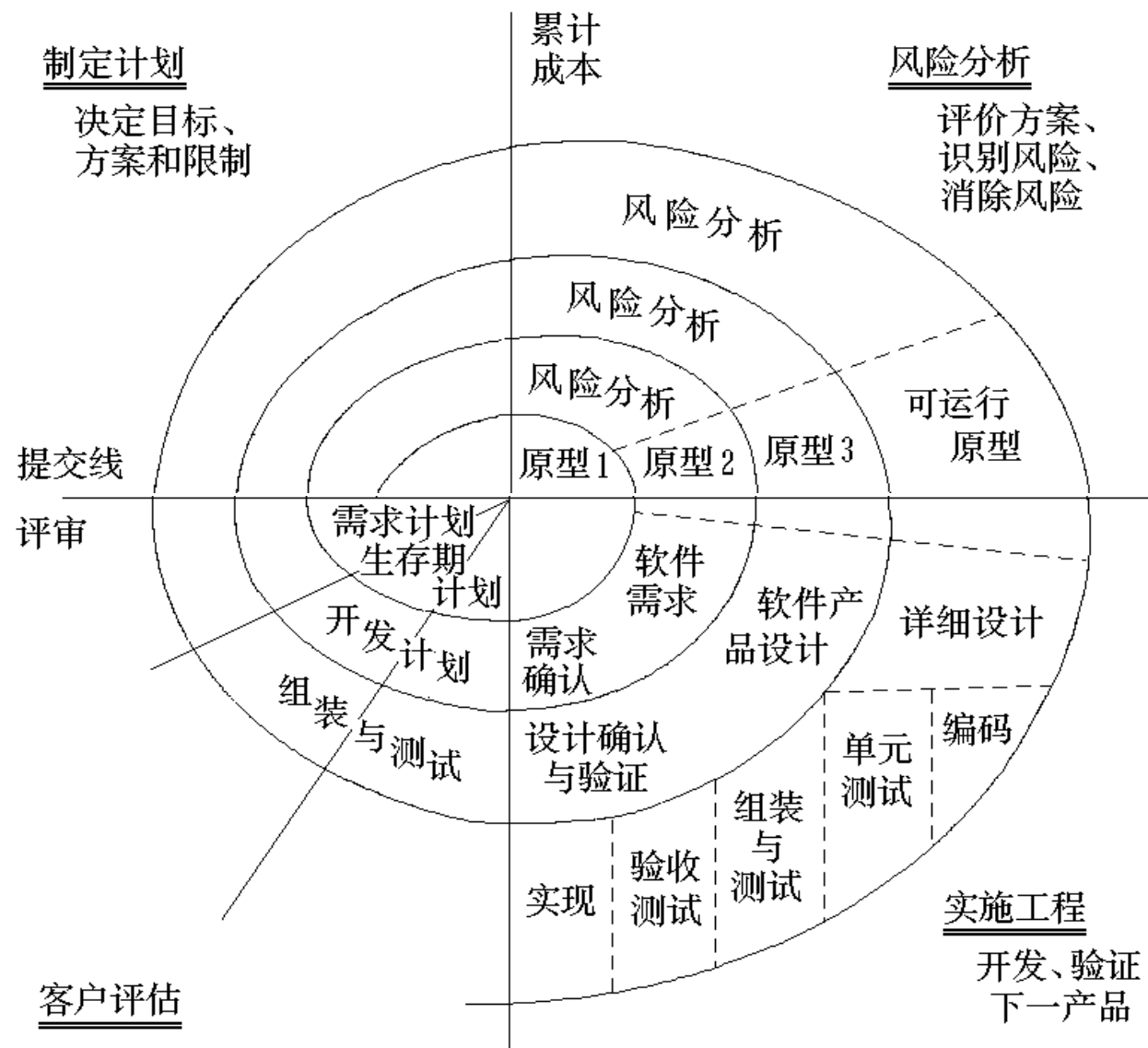
- ✓ 用户界面
- ✓ 执行流程



### ➤ 特点

- ✓ 软件原型作为交流载体和媒介
- ✓ 支持用户参与到软件开发中
- ✓ 持续、渐进地导出用户要求
- 适合于需求难导出、不易确定且持续变动的软件

## 2.7 螺旋模型(Spiral Model)



### □ 软件风险

✓ 使软件开发受到影响和损失、甚至导致失败的、可能会发生的事件

- 集成迭代模型和原型模型
- 引入风险分析，风险驱动的过程模型
- 每个迭代四个阶段，若干活动
- 适合于需求不明确、开发风险高、开发过程中需求变更更大的软件项目
- 不足：管理复杂

# 不同软件过程模型的特点

模型名称	指导思想	关注点	适合软件	管理难度
瀑布模型	提供系统性指导	与软件生命周期相一致	需求变动不大、较为明确、可预先定义的应用	易
原型模型	以原型为媒介指导用户的需求导出和评价	需求获取、导出和确认	理解需求难以表述清楚、不易导出和获取的应用	易
增量模型	快速交付和并行开发	软件详细设计、编码和测试的增量式完成	需求变动不大、较为明确、可预先定义的应用	易
迭代模型	多次迭代，每次仅针对部分明确软件需求	分多次迭代来开发软件，每次仅关注部分需求	需求变动大、难以一次性说清楚的应用	中等
螺旋模型	集成迭代模型和原型模型，引入风险分析	软件计划制定和实施，软件风险管理，基于原型的迭代式开发	开发风险大，需求难以确定的应用	难



# 内容

## 1. 何为软件过程模型

✓ 基本概念和特点

## 2. 有哪些软件过程模型

✓ 有什么类别，各有什么特点和优缺点

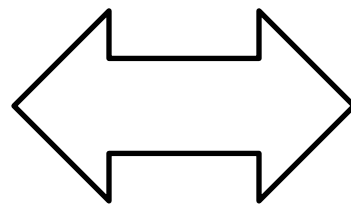
## 3. 如何来选择软件过程模型

✓ 软件过程模型的选择方式和策略



# 软件项目 与 软件过程模型

**软件项目及开  
发团队的具体  
特点**



**软件过程模型  
的特点、优点  
和局限**



# 3.1 软件过程模型的选择

## □考虑软件项目的特点

- ✓尤其是所开发软件的业务特点，如业务领域是否明确、软件需求是否易于确定、用户需求是否会经常性变化等等
- ✓是否可以预估到潜在的软件开发风险

## □软件开发团队的水平

- ✓需要结合软件开发团队的能力和水平来选择过程模型，以防开发团队和管理人员无法掌控和驾驭过程模型

## □分析软件过程模型特点

- ✓优缺点以及适合的场所

# 示例：如何选择合适的过程

## □互联网应用软件开发过程模型

- ✓特点：软件需求不确定且快速变化
- ✓如：12306 APP软件，微信软件，淘宝软件
- ✓选用瀑布模型不合适，迭代模型较为合适



## □装备软件开发过程模型

- ✓特点：软件需求确定且较为稳定
- ✓如：飞行控制软件
- ✓可考虑选用瀑布模型，用迭代模型不是很合适



# 思考和讨论

□结合课程综合实践的具体特点和要求，思考**选用什么样的软件过程模型较为合适，为什么？**

- ✓软件**有创意**：问题及基于软件的解决方法有新意
- ✓软件**上规模**：软件具有一定规模，代码量  $> 15000 + \text{LOC}$
- ✓如空巢老人看护软件、多无人机联合搜寻软件

课程实践软件项目有何特点？

软件项目开发团队有何特点？



## 3.2 传统软件过程模型的特点和不足

- 软件开发和运维的大量工作用于**撰写软件文档**，而非去编写程序代码
- 软件开发过程中会花费大量时间和精力用于**软件文档的评审**，以确保软件质量
- 一旦**软件需求发生变化**，开发人员需要修改软件需求文档，并据此来调整其他的一系列文档，最后再修改程序代码
- 等**较长时间**才能得到**可运行软件系统**

- **瀑布模型**
- **增量模型**
- **迭代模型**
- **原型模型**
- **螺旋模型**
- **.....**

以文档为中心的重型软件开发方法，非常笨重

# 敏捷软件开发方法(Agile Method)

- 重视人和交互、重视可运行软件系统、重视客户合作、重视响应用户需求变化
- 少写软件文档，以代码为中心，快速响应变化

## □软件开发需要过程指导

- ✓明确步骤、活动、次序、关系

## □多样化的软件过程模型

- ✓瀑布、增量、迭代、原型、螺旋等
- ✓各自有其优缺点

## □选择合适的软件过程模型

- ✓考虑软件项目的特点和要求
- ✓结合软件过程模型的优缺点
- ✓考虑开发团队的经验和水平

# 布置课程综合实践任务

## □综合实践1：阅读、分析和维护开源软件

- ✓结合代码阅读以及sonarqube工具扫描，分析小米便签开源软件的质量，撰写质量分析报告
- ✓要求：分析代码质量（从软件设计到代码规范）好的一面；存在的不足和问题，在具体开发中应用高质量的软件开发规范和实践

## □综合实践2：开发有创意、上规模和高质量的软件

- ✓结合要求构思软件需求，撰写软件需求构思和描述文档
- ✓要求：有创意，上规模；力争将软件需求讲清楚：what & why

# 下一次课程的内容

## □学生

- ✓汇报开源代码质量分析报告
- ✓汇报软件需求构思和描述文档

## □教师

- ✓讲评学生的课程实践



# 思考题

**迭代模型能够有效的支持  
课程实践软件项目的开发  
和管理工作？为什么？**



# 问题和讨论

