

软件过程模型

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

3. 如何选择软件过程模型

- ✓ 软件过程模型的选择方式和策略



1.1 软件过程

□过程(Process)

- ✓ **活动**：具体要做的事
- ✓ **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)

- ✓ 定义**软件开发和维护**中的一组**有序活动集合**
- ✓ 软件开发活动：**技术活动和管理活动**

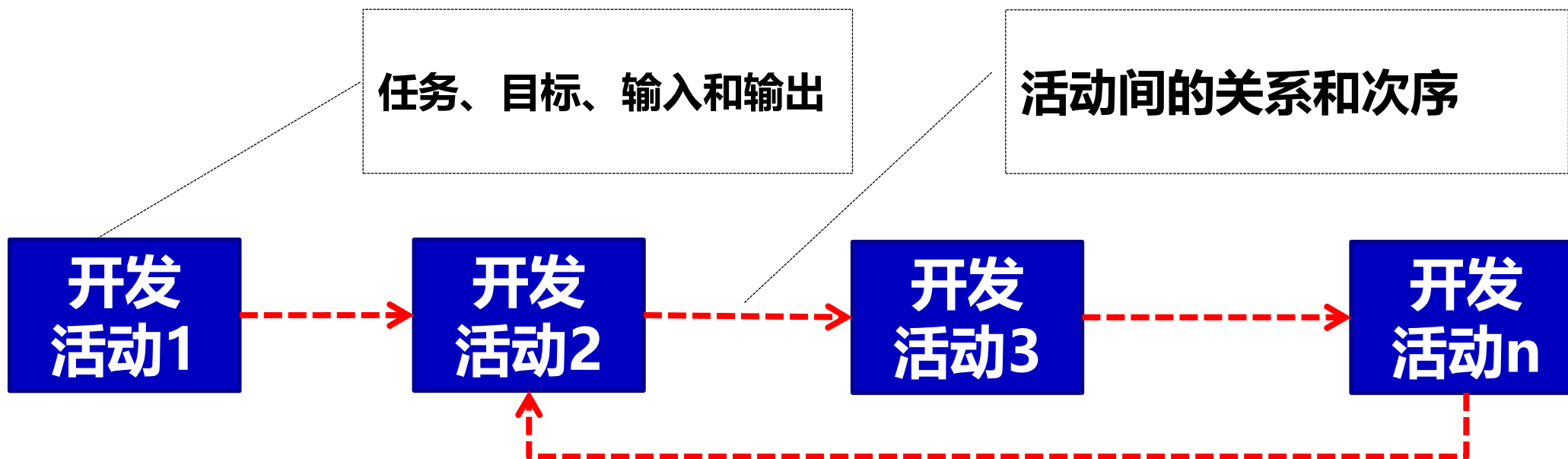
按照什么样的过程来有序地开发软件？



1.2 软件过程模型

□ 软件过程模型(Software Process Model)

- ✓ 用于描述软件过程中的**活动**以及活动间的逻辑**关系**



类比建筑过程

规划设计-建设准备-地基施工-主体结构施工-室内装修-外墙装修

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

➤ 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

没有好的过程

就没有好的产品！

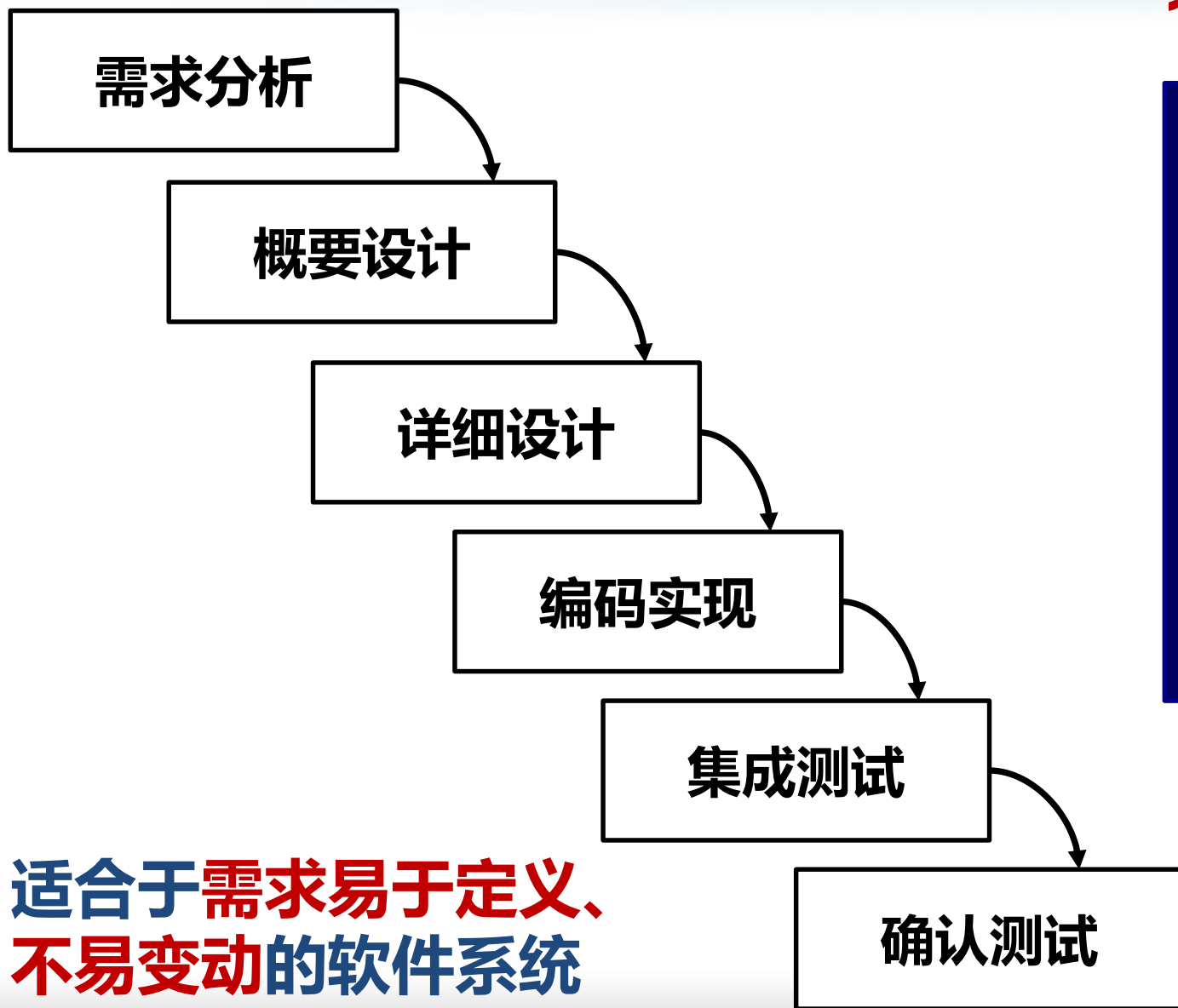
典型的软件过程模型

- 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

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

- ✓ 与软件生命周期一致
- ✓ 每个活动结束后均有制品输出并需要评审
- ✓ 相邻活动间存在先后关系

➤ 优点

- ✓ 简单，易理解、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？

□产出

- ✓**软件需求模型、软件需求文档、测试计划**
- ✓**文档类软件制品**



产品说，做一个需求：手机摇一摇，让周围人的支付宝余额都转我卡里！

概要设计(Architecture Design)

□活动

- ✓任务：设计**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

问题如何解决？

□产出

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



详细设计(Detailed Design)

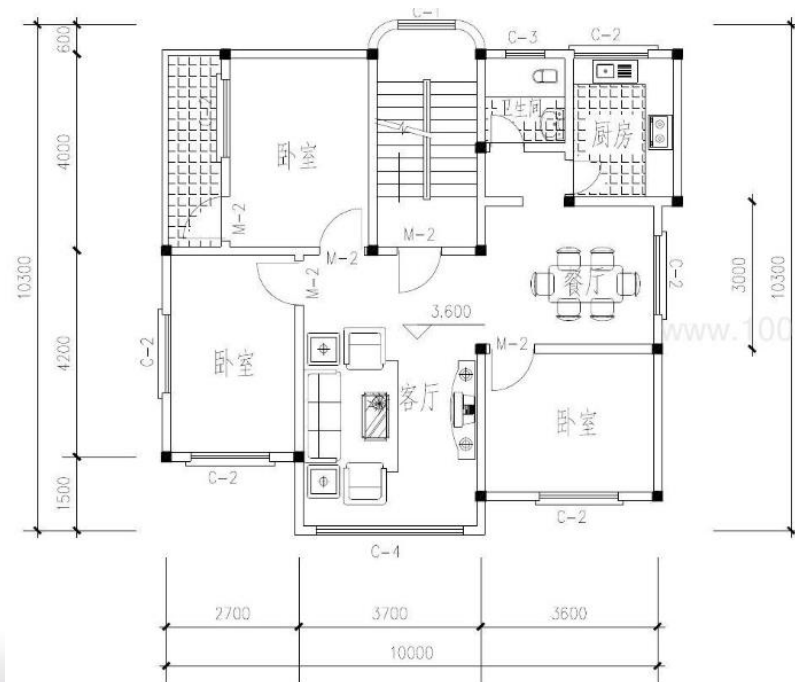
□活动

- ✓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?

□产出

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



编程实现(Implementation)

□活动

- ✓任务：编码、单元测试、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

实际解决问题

□产出

- ✓经过单元测试的源代码
- ✓程序类软件制品



集成测试(Integration Test)

□活动

- ✓任务： **组装软件模块并进行测试**
- ✓关注点： **集成后软件中的缺陷 (Bug)**

**问题解决如何？
软件有缺陷吗？**

□产出

- ✓ **集成后的源程序代码，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？

□ 产出

- ✓完整的系统代码，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

严重依赖需求的正确性、完整性、不变性。

□现实

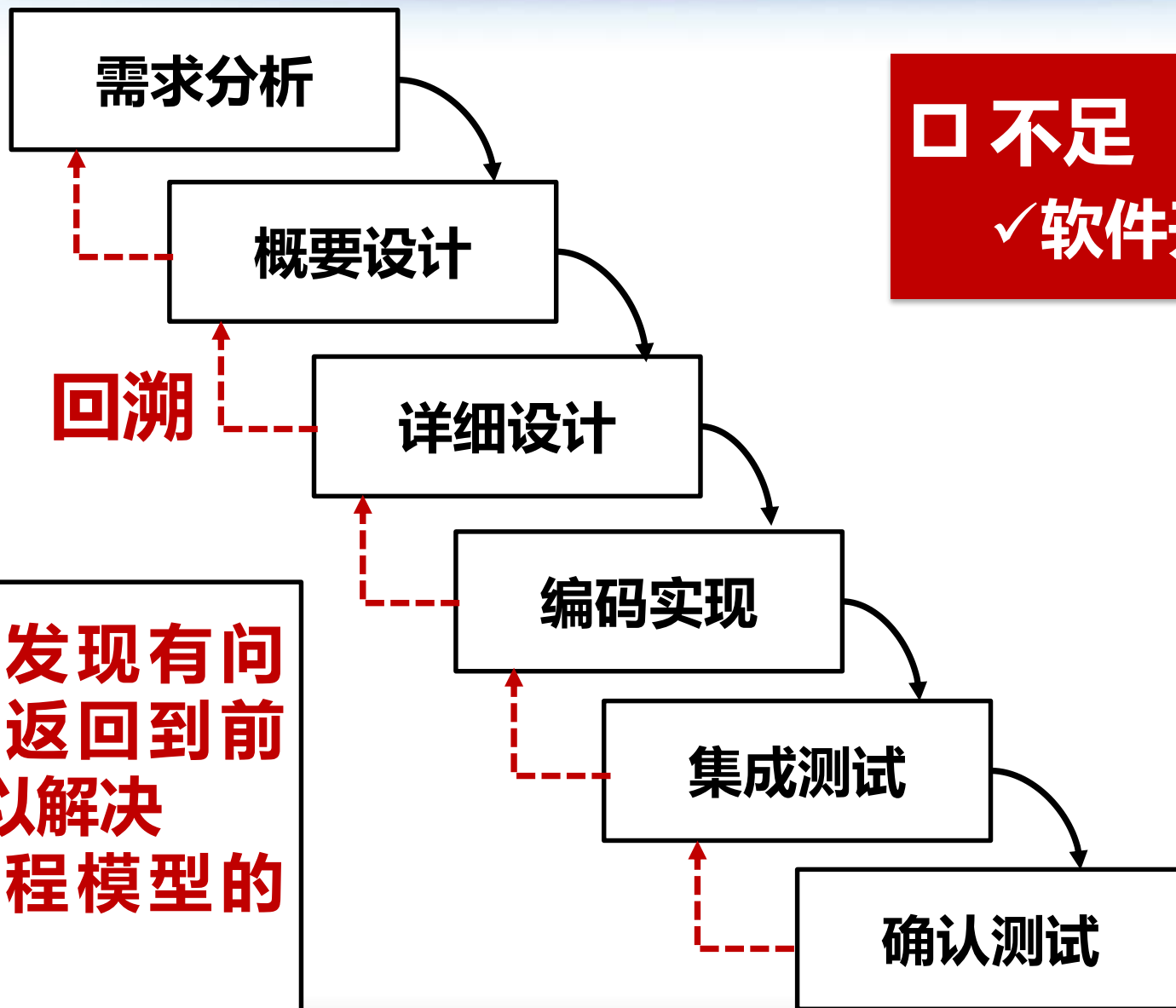
客户没有看到软件前，无法可靠的描述产品需求。

□不足

- ✓项目开始就确定需求，过于理想化
- ✓缺乏变通，难应对变化
- ✓文档过于复杂，占用大量时间
- ✓用户要等到开发后期才能见到可运行的产品，增加了项目失败的风险



2.3 改进的瀑布模型：带反馈和回溯



□ 不足

✓ 软件开发处于动荡之中

回溯

详细设计

编码实现

集成测试

确认测试

- ✓ 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

增量1

详细设计

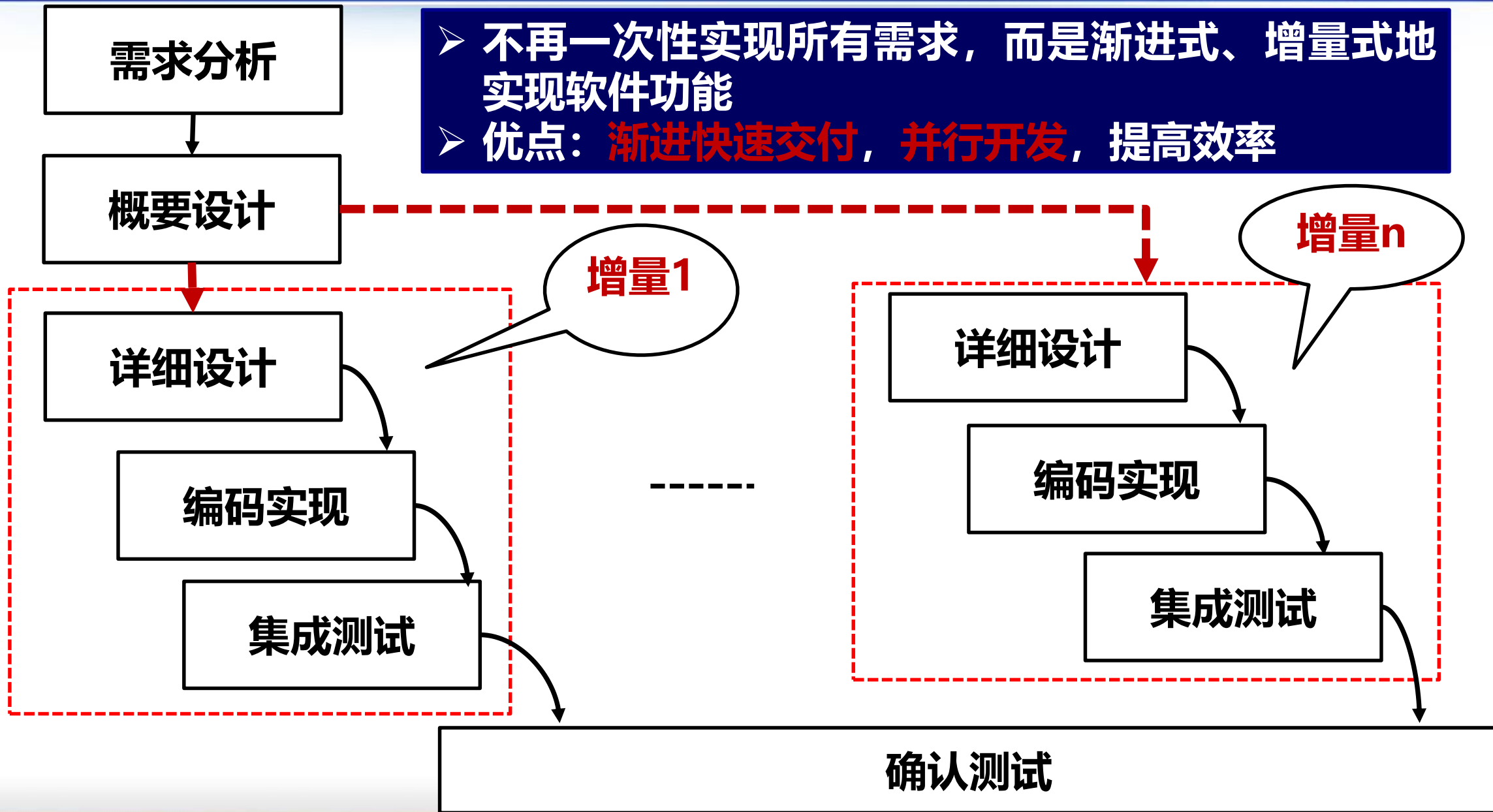
编码实现

集成测试

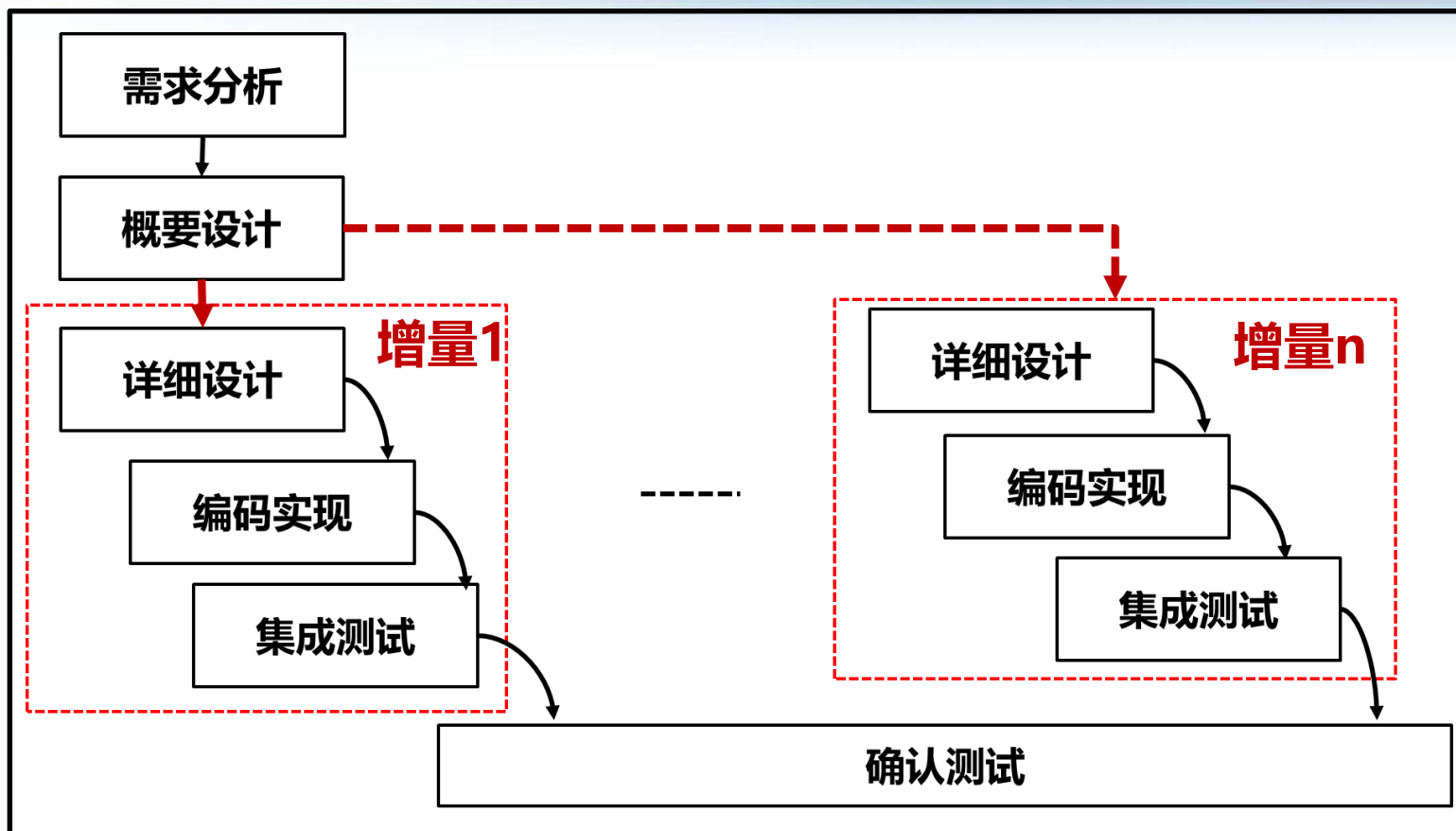
增量n

确认测试

- 不再一次性实现所有需求，而是渐进式、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

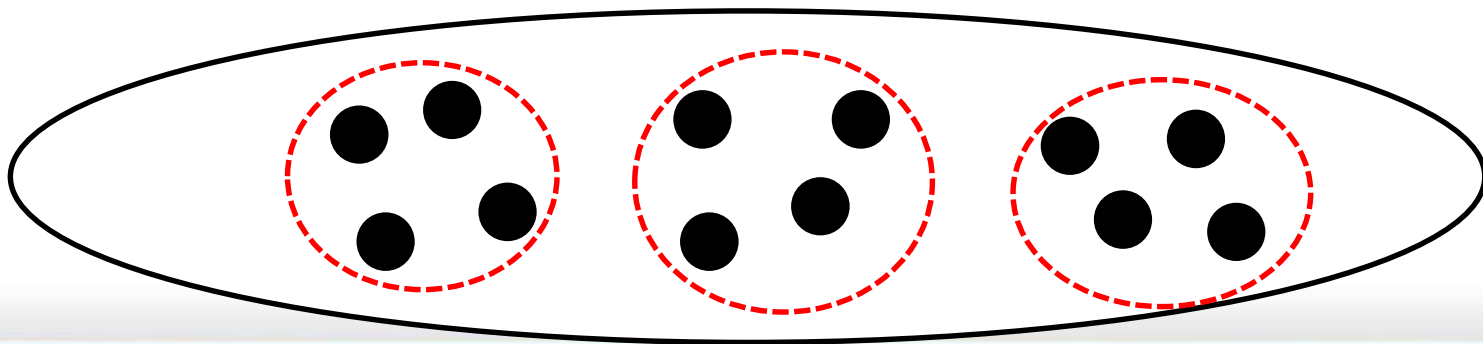


增量模型的局限性?



□ 不足

✓ 仍然无法处理需求变更导致的问题

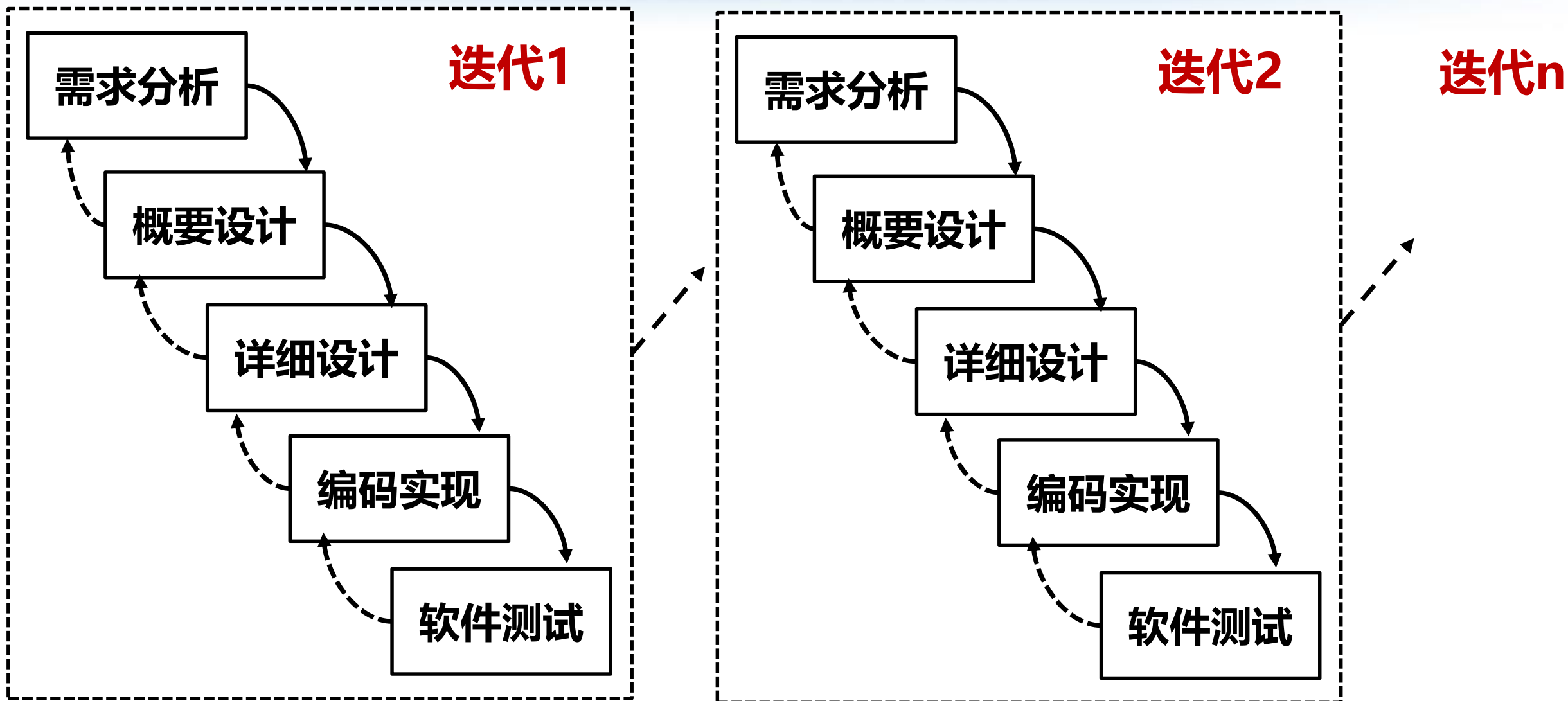


增量模型使用要点

□ 注意

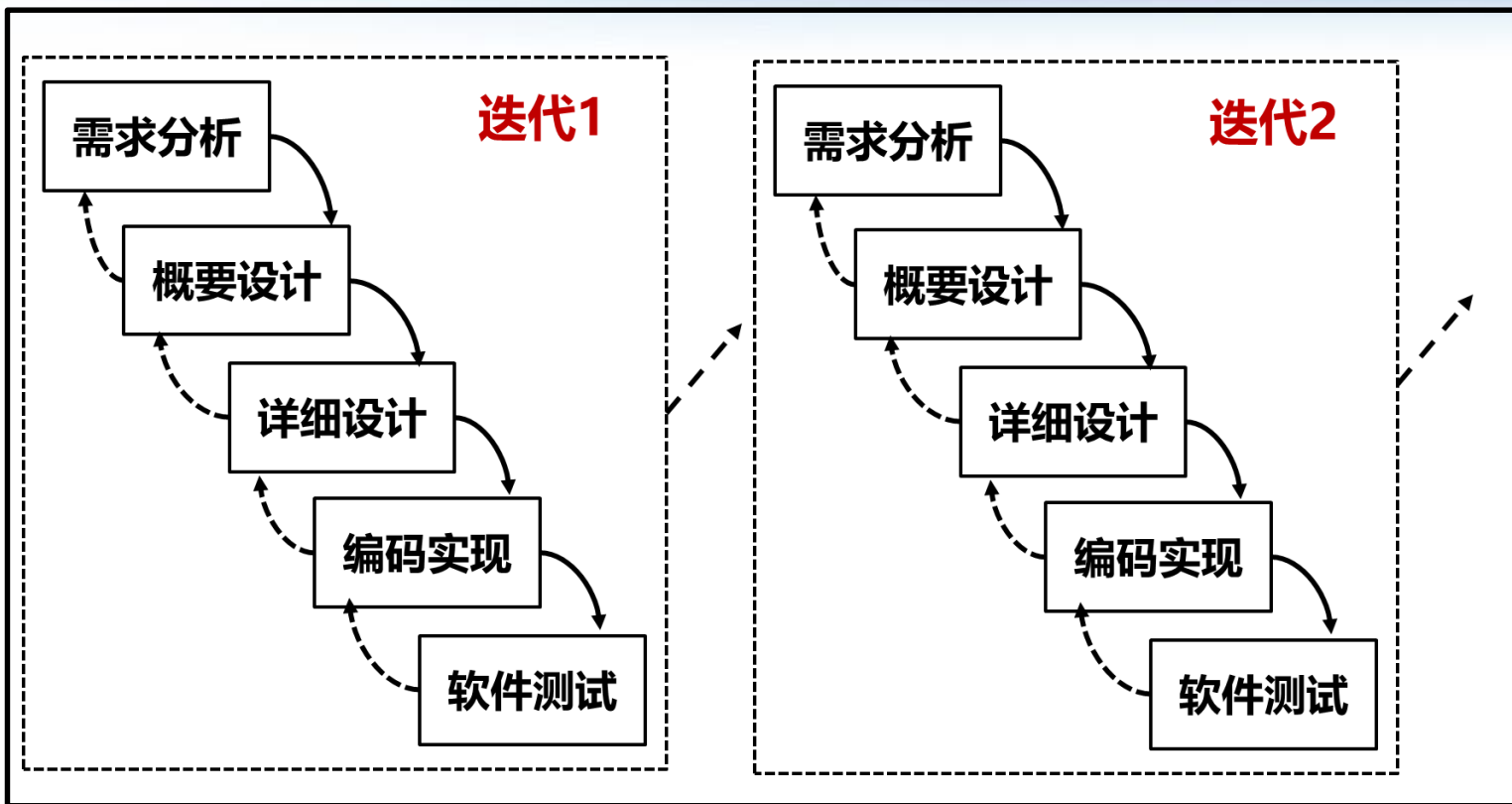
- ✓ 第一个增量最重要，优先考虑最核心、最稳定的需求。
- ✓ 由于一些模块必须在另一个模块之前完成，所以要注意增量间的顺序和良好的接口。
- ✓ 在团队成员不足以支持项目完整开发时尤其适用。

2.5 迭代模型(Iterative Model)

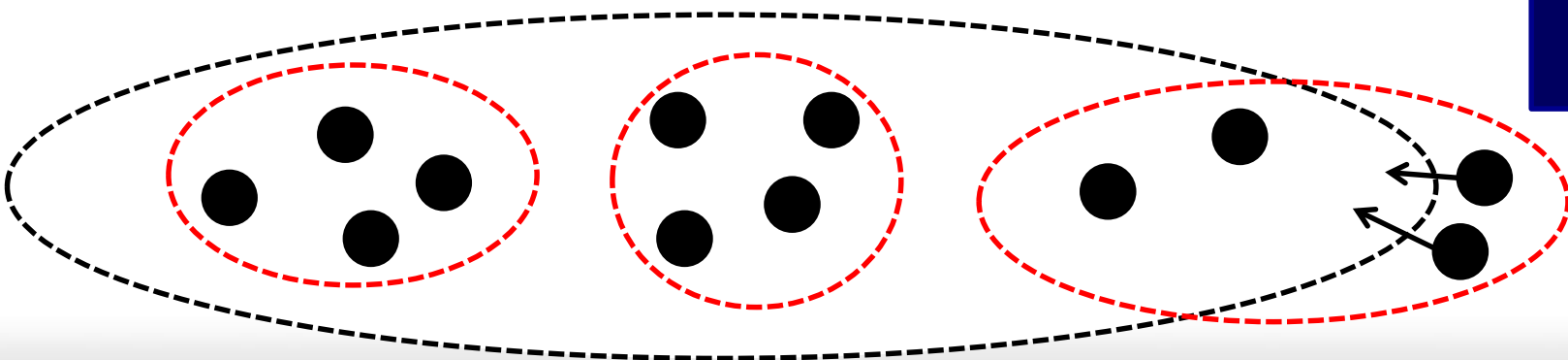


每次迭代只针对**当前已确定的软件需求** (这有别于增量模型)

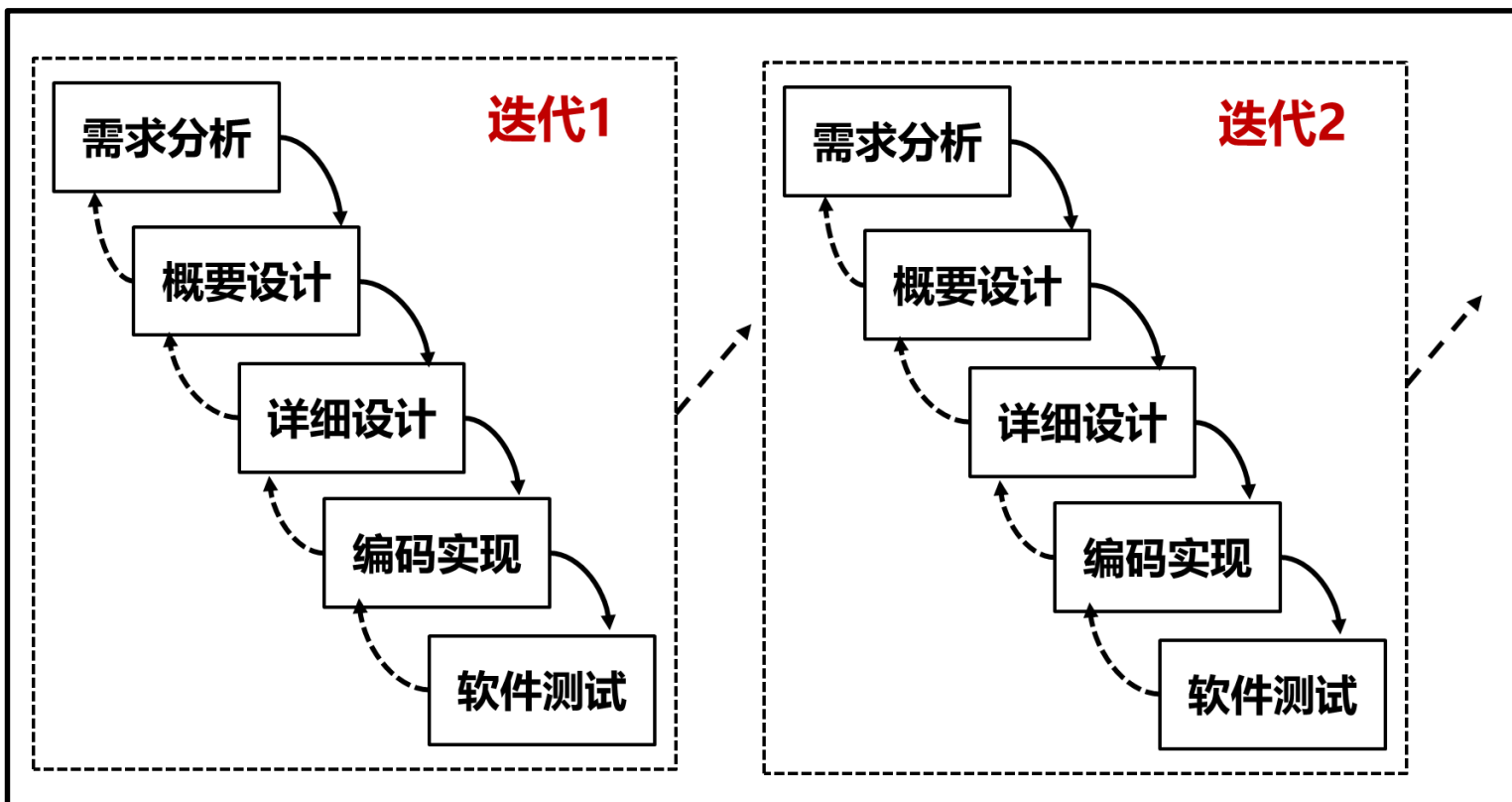
迭代模型的特点



- 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?



□ 不足

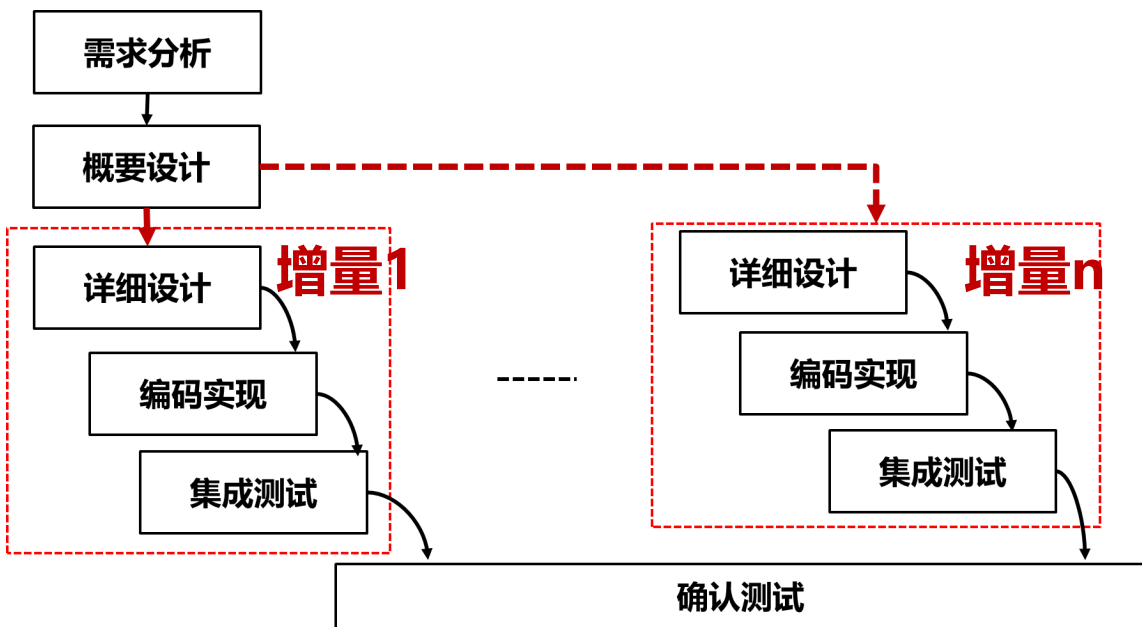
✓ 迭代次数不确定

✓ 管理较为复杂

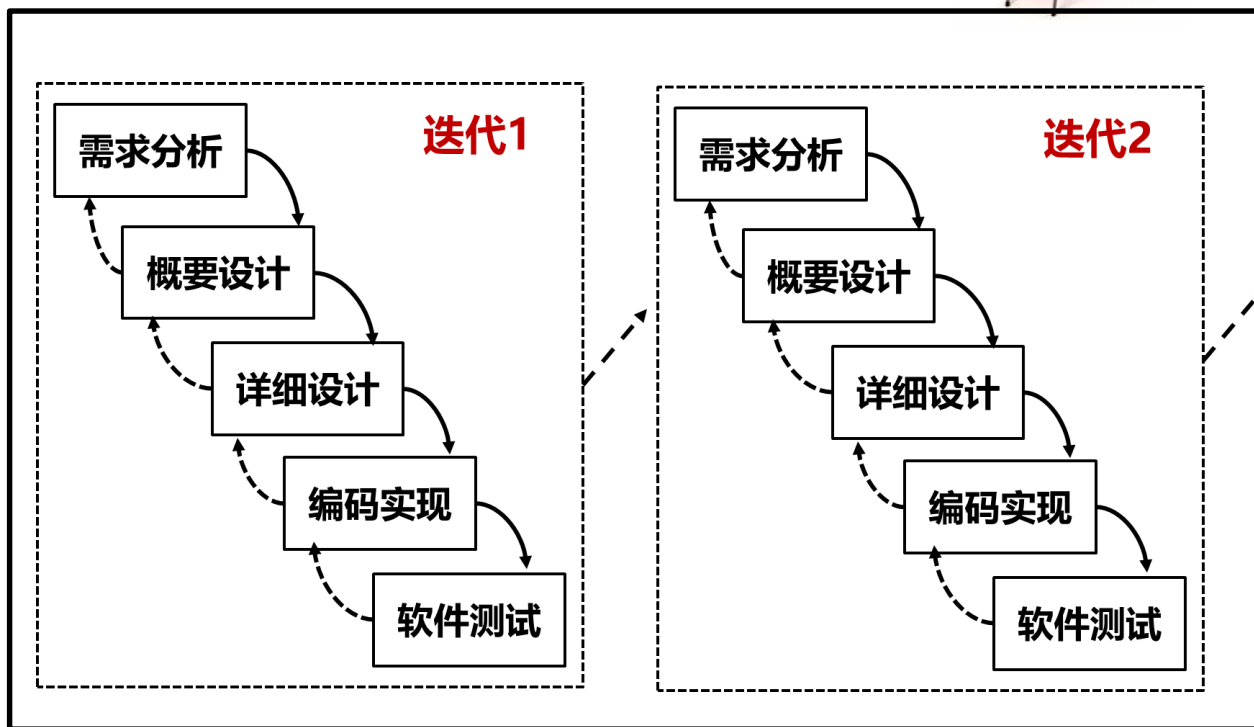


思考和讨论

□ 增量过程模型与迭代过程模型有何区别？



增量模型



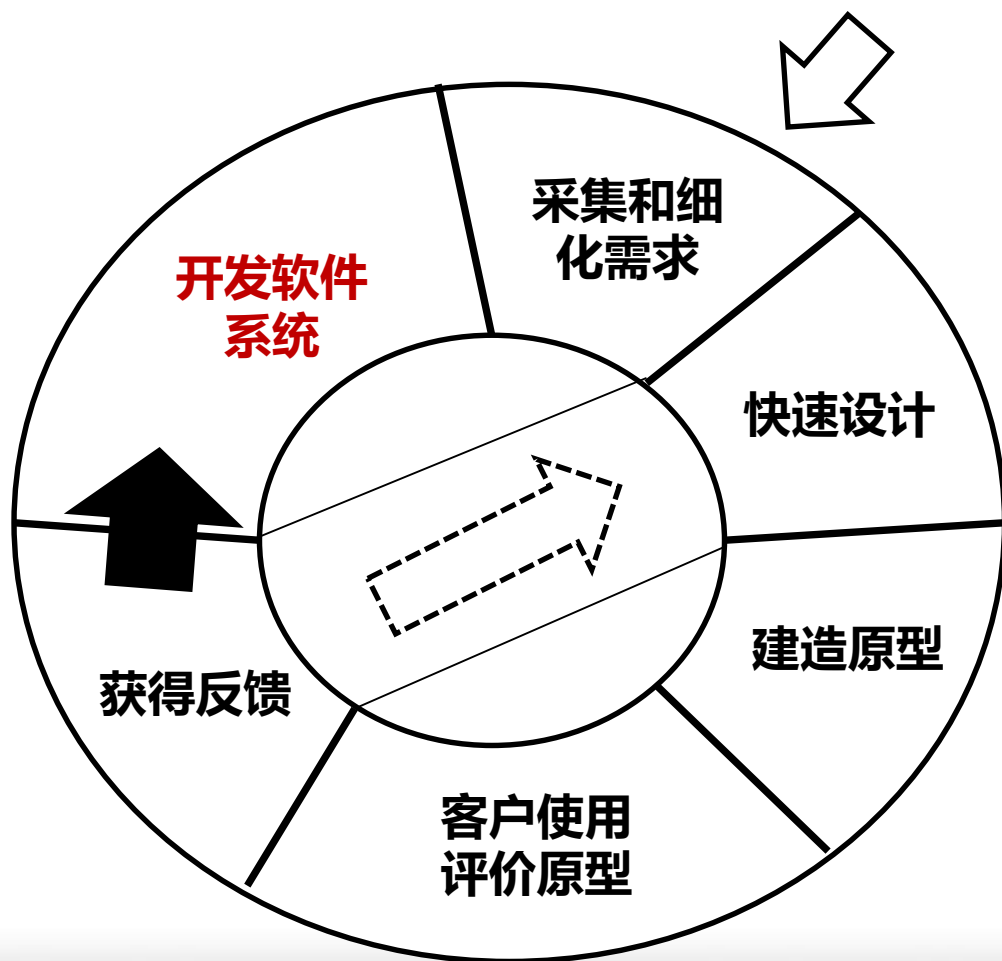
迭代模型

2.6 原型模型(Prototype Model)

□ **原型**就是产品早期的雏形或仿真

□ **何为软件原型?**

- ✓ 用户界面
- ✓ 执行流程



□ **原型的主要作用**

- ✓ 尽早验证需求
- ✓ 尽早明确不确定性的因素
- ✓ 作为媒介，便于与用户沟通
- ✓ 为后续界面设计提供基础

2.6 原型模型(Prototype Model)

➤ 适合于需求**难导出、难以描述**或确定的软件

采购入库

入库单号

入库类型：采购入库

仓库名称 请选择

采购单号

入库时间

新增1

编辑

删除

审核6

打印

<input type="checkbox"/>	入库单号	仓库名称	制单人	入库类型	入库时间	审核状态	审核人	审核时间	入库说明
<input type="checkbox"/>	RK-1-2-3	福暉仓库	祥子	采购入库	2017-12-13	待审核	XXXXXX	XXXXX	XXXXXXXX

新增入库单

*入库编号

RK201712081418515772395

制单人员

XXX

*选择仓库

华北仓

入库时间

2017/12/08 14:18:51

采购订单5

不选采购订单也可以入库

备注

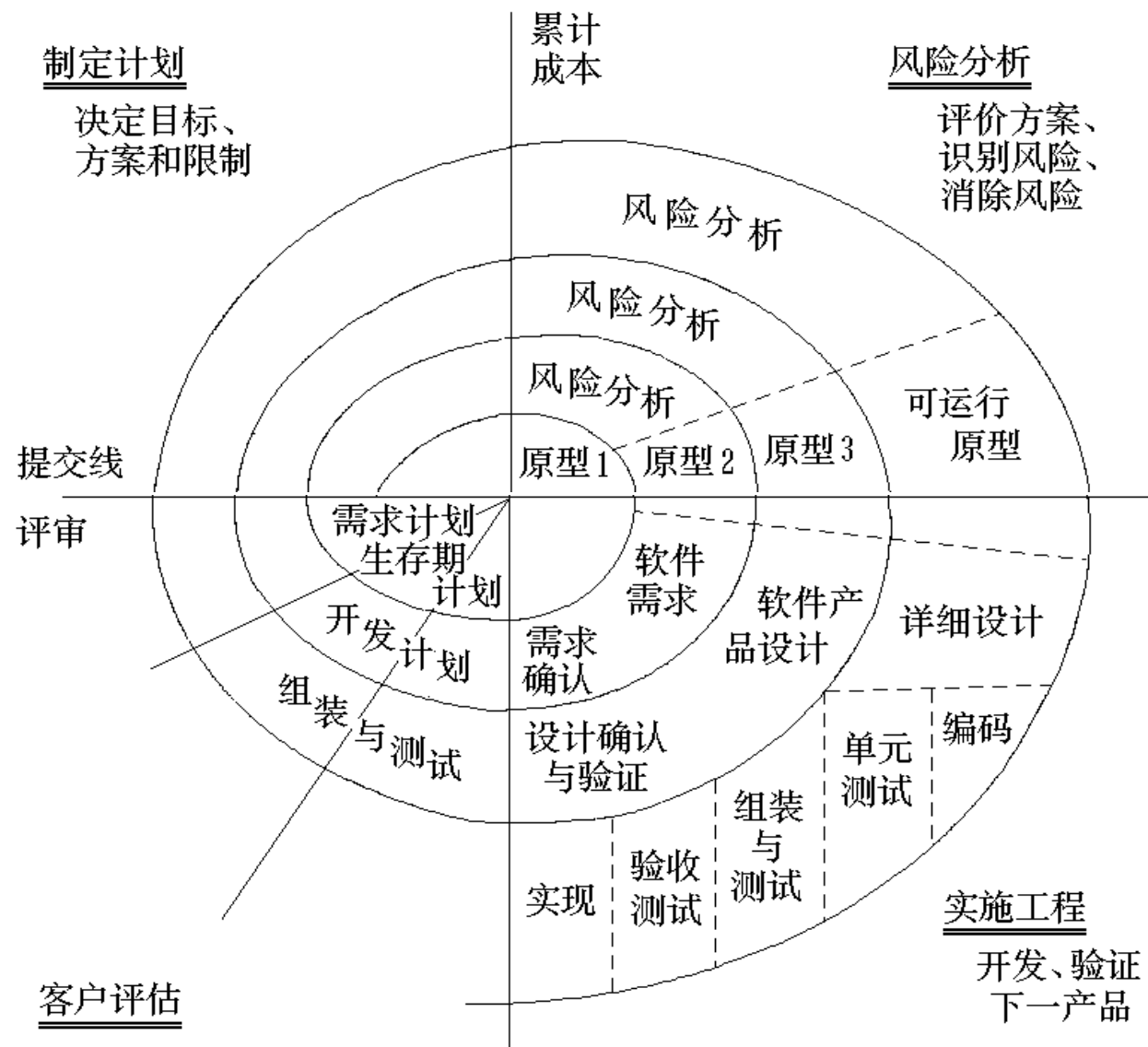
选择商品

商品导入

删除行

<input type="checkbox"/>	商品编码	商品名称	规格名称	条形码	入库数量	采购数量	已入库	商品成本	小计
					5				

2.7 螺旋模型(Spiral Model)



- 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
- 每个迭代均包括四个阶段
- 适合于需求不明确、开发风险高、系统规模大的软件项目
- 不足：管理复杂

第1圈：开发出产品的规格说明；

第2圈：开发产品的原型系统；

第3~n圈：不断的迭代，开发不同的软件版本；

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



3.1 软件过程模型的选择

□考虑软件项目的特点

- ✓尤其是待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
- ✓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

- 大量工作用于**撰写软件文档**，而非去编写程序代码
- 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以及其它文档，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

小结

□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

问题和讨论

