

软件工程



课程概况

- 教材:《软件工程导论(第6版)》,张海藩编著,清华大学出版社

第1章 软件工程学概述

第2章 可行性研究

第3章 需求分析

第5章 总体设计

第6章 详细设计

第7章 实现

第8章 维护

第9章 面向对象方法学引论

第10章 面向对象分析

第11章 面向对象设计

第12章 面向对象实现



●考核方式

考勤和课堂表现： 15%

实验： 25%

书面考试： 60%

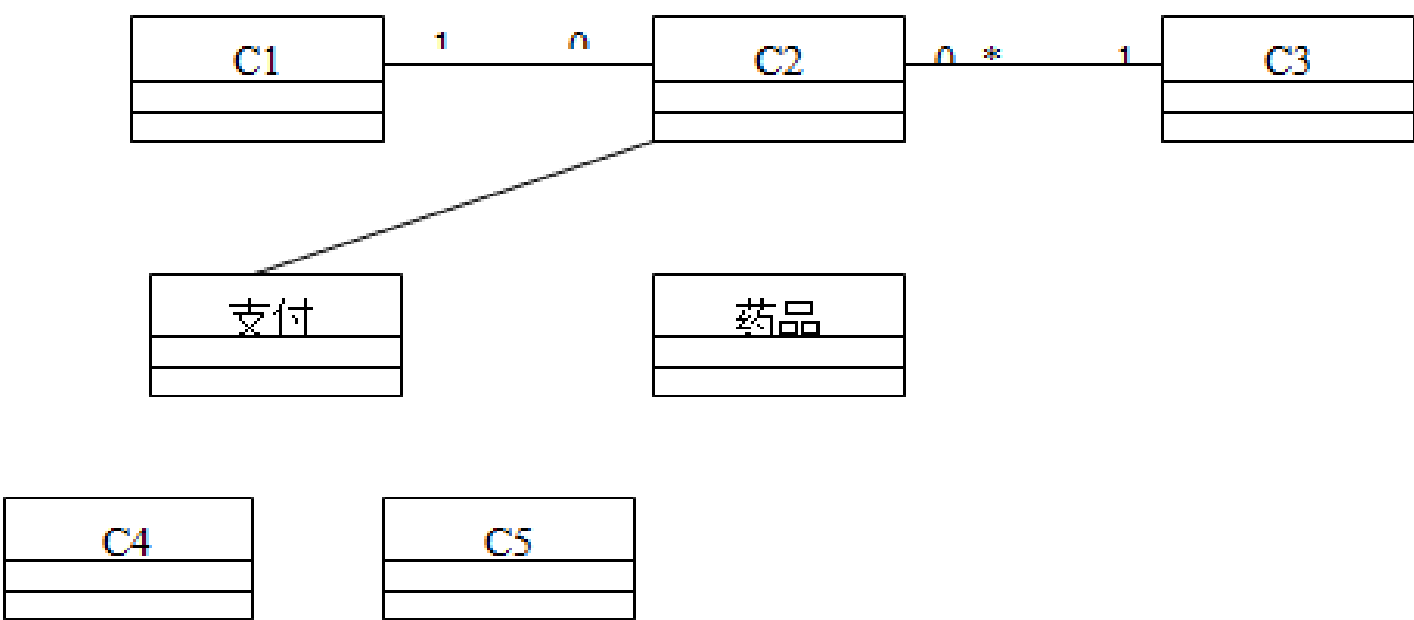


5、阅读说明，回答问题。（8分）

【说明】某网上药店允许顾客凭借医生开具的处方购买药品，该网上药店的基本功能为：

- 1) 顾客必须在网上药店注册，填写顾客资料。
- 2) 注册顾客在网上药店登录成功后，进入处方录入界面填写开具处方的医生信息以及处方上的药品信息，填写完成后提交处方。
- 3) 系统对于已提交的处方进行验证，验证处方中的医生信息是否有效，以及该处方是否为医生所开具等。系统取消未通过验证的处方。
- 4) 对于通过验证的处方，系统计算药品价格。在顾客对所购处方药品进行支付，支付可以使用信用卡或支付宝账户。支付后，邮寄药品给顾客。

下图为网上药店的不完整类图。



软件工程师（Software Engineer）是一个认证考试，具体地说是从事软件职业的人员的一种职业能力的认证，通过它说明具备了工程师的资格。

一般情况下都是分4级：

- 1、软件技术员
- 2、助理软件工程师
- 3、软件工程师
- 4、高级软件工程师



软件工程师



| | | | | |
|--|---------------------|---------|------------|-------|
| <input type="checkbox"/> 互联网软件工程师 | 上海开高软件科技有限公司 | 上海-松江区 | 0.8-2万/月 | 11-24 |
| <input type="checkbox"/> 嵌入式软件工程师 | 上海天庸科技发展有限公司 | 上海-黄浦区 | 20-30万/年 | 11-24 |
| <input type="checkbox"/> 嵌入式软件工程师 | 徕卡测量系统 | 上海 | 1-1.5万/月 | 11-24 |
| <input type="checkbox"/> 全栈软件工程师/Junior Full Stack En... | 数码方块软件开发（上海）有限公司... | 上海-徐汇区 | 1-2万/月 | 11-24 |
| <input type="checkbox"/> 协议软件工程师 | 宏达通讯有限公司（HTC） | 上海-浦东新区 | 1.5-2.2万/月 | 11-24 |
| <input type="checkbox"/> 物流软件工程师 | 米思米（中国）精密机械贸易有限公... | 上海-奉贤区 | 0.8-1.3万/月 | 11-24 |
| <input type="checkbox"/> 软件工程师（视觉事业部） | 开异智能技术（上海）有限公司 | 上海-松江区 | 1.5-2.5万/月 | 11-24 |
| <input type="checkbox"/> 嵌入式软件工程师 | 上海奥世信息科技有限公司 | 上海 | 1-2万/月 | 11-24 |
| <input type="checkbox"/> c++/上位机软件工程师 | 上海信耀电子有限公司 | 上海 | 1-2万/月 | 11-24 |
| <input type="checkbox"/> 中级软件工程师 | 特励丝软件科技（上海）有限公司 | 上海 | 1-1.5万/月 | 11-24 |
| <input type="checkbox"/> 初级软件工程师 | 上海海曙智能科技有限公司 | 上海-浦东新区 | 0.6-1万/月 | 11-24 |
| <input type="checkbox"/> 嵌入式软件工程师 | 格雷博智 上海复光竞成科技有限公司 | 上海-浦东新区 | 0.8-1.8万/月 | 11-24 |
| <input type="checkbox"/> 软件工程师 | 上海复光竞成科技有限公司 | 上海-杨浦区 | 1-2万/月 | 11-24 |
| <input type="checkbox"/> C# winform/.Net软件工程师 | 上海欧融芝实业有限公司 | 上海-静安区 | 0.8-1.5万/月 | 11-24 |
| <input type="checkbox"/> 嵌入式软件工程师 | 上海翌虹信息技术有限公司 | 上海-闵行区 | 1-1.5万/月 | 11-24 |



项目管理师

 编辑

项目管理师是指掌握项目管理原理、技术、方法和工具，参与或领导启动、计划、组织、执行、控制和收尾过程的活动，确保项目能在规定的范围、时间、质量与成本等约束条件下完成既定目标的人员。

| | | | |
|-----|-----------------------------|-----|----------------|
| 中文名 | 项目管理师 | 指 | 掌握项目管理原理、技术、方法 |
| 外文名 | Project Management Division | 时 间 | 20世纪50年代后期 |

| | | | |
|----|--------|--------|--------|
| 目录 | 1 资格等级 | 4 项目背景 | 7 学习目标 |
| | 2 考试介绍 | 5 职业概况 | 8 考试政策 |
| | 3 申报条件 | 6 课程介绍 | 9 注意事项 |

资格等级

 编辑

这是中国项目管理CPMP

项目管理员

助理项目管理师

项目管理师

高级项目管理师



考试方式

学术型硕士研究生（计算机科学与技术、软件工程）：

初试科目：包括英语一、数学一、政治、专业课（计算机组成原理和与数据结构）（专业课为自命题）；

复试：包括笔试和面试两部分，笔试科目为编译原理和C++程序设计，面试主要为英语口语测试以及专业综合能力面试。

全日制专业学位硕士研究生（软件工程）：

初试科目：包括英语二、数学二、政治、专业课（操作系统与数据结构）（专业课为自命题）；

复试：包括笔试和面试两部分，笔试科目为C++程序设计，面试主要为英语口语测试以及专业综合能力面试。

毕业去向

外资企业：如谷歌、IBM、微软、ORACLE、苹果、EMC、SAP等。

高新企业：如腾讯、百度、网易、淘宝、华为、中兴等。

大型国企：如移动、联通、电信、国家电网等。

银行、金融机构、政府机关、事业单位等。

目录

| | |
|------------------------|--|
| 摘要..... | |
| ABSTRACT..... | |
| 第 1 章 绪论..... | |
| 第 2 章 相关技术与工具..... | |
| 第 3 章 XX 系统需求分析..... | |
| §3.1 可行性分析..... | |
| §3.2 需求分析..... | |
| 第 4 章 XX 系统的概要设计..... | |
| §4.1 系统功能模块设计..... | |
| §4.2 数据库设计..... | |
| 第 5 章 XX 系统界面设计..... | |
| §5.1 登录模块..... | |
| §5.2 主页及导航栏介绍..... | |
| 第 6 章 XX 系统功能模块设计..... | |
| §6.1 通用功能..... | |
| §6.2 系统管理员角色功能..... | |
| 第 7 章 XX 系统的系统测试..... | |
| 第 8 章 总结与展望..... | |
| 致谢..... | |
| 参考文献..... | |



第1章 软件工程学概述

- 软件危机
- 软件工程
- 软件生命周期
- 软件过程



1.1 软件危机

软件危机的介绍
产生软件危机的原因
消除软件危机的途径

软件危机指在计算机软件的开发和维护过程中，所遇到的一系列严重问题。



■开发费用和进度难以估算和控制，大大超过预期的资金和规定日期

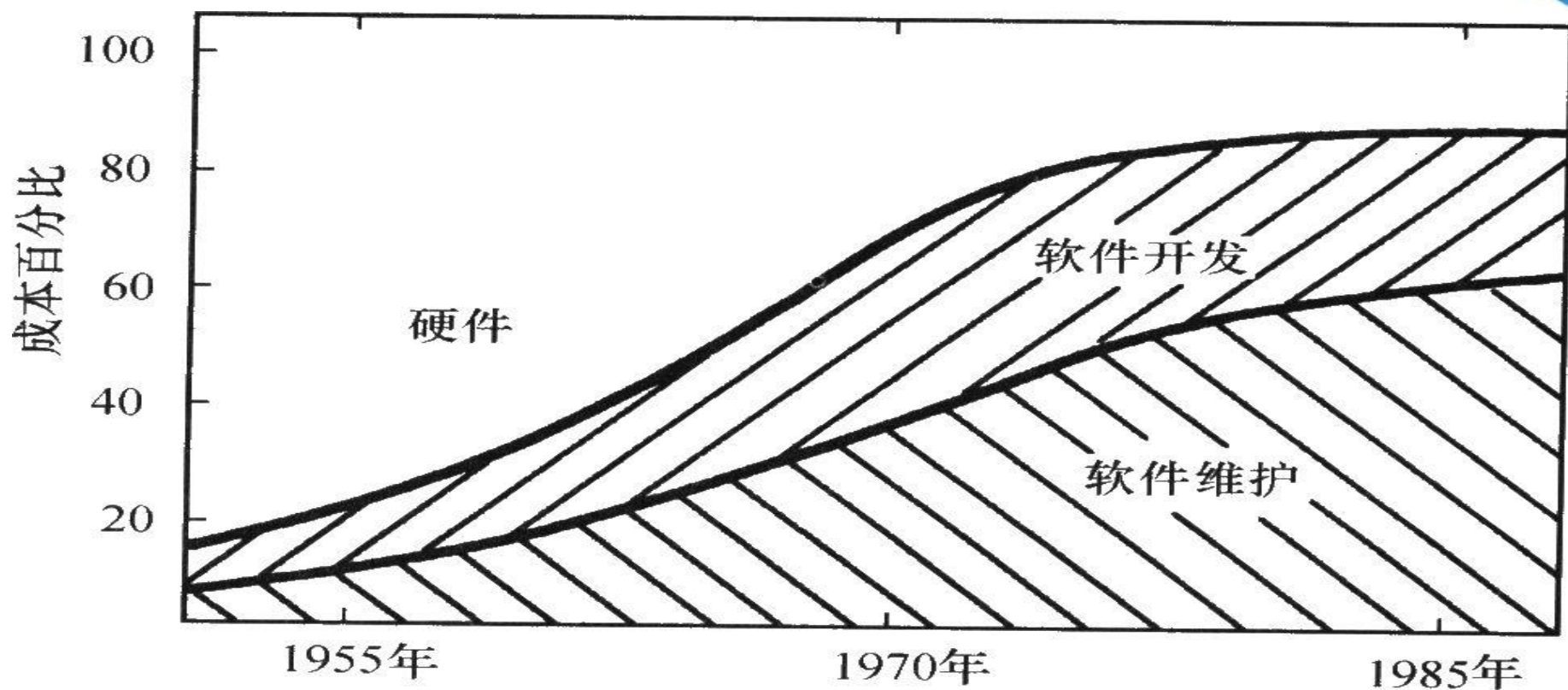
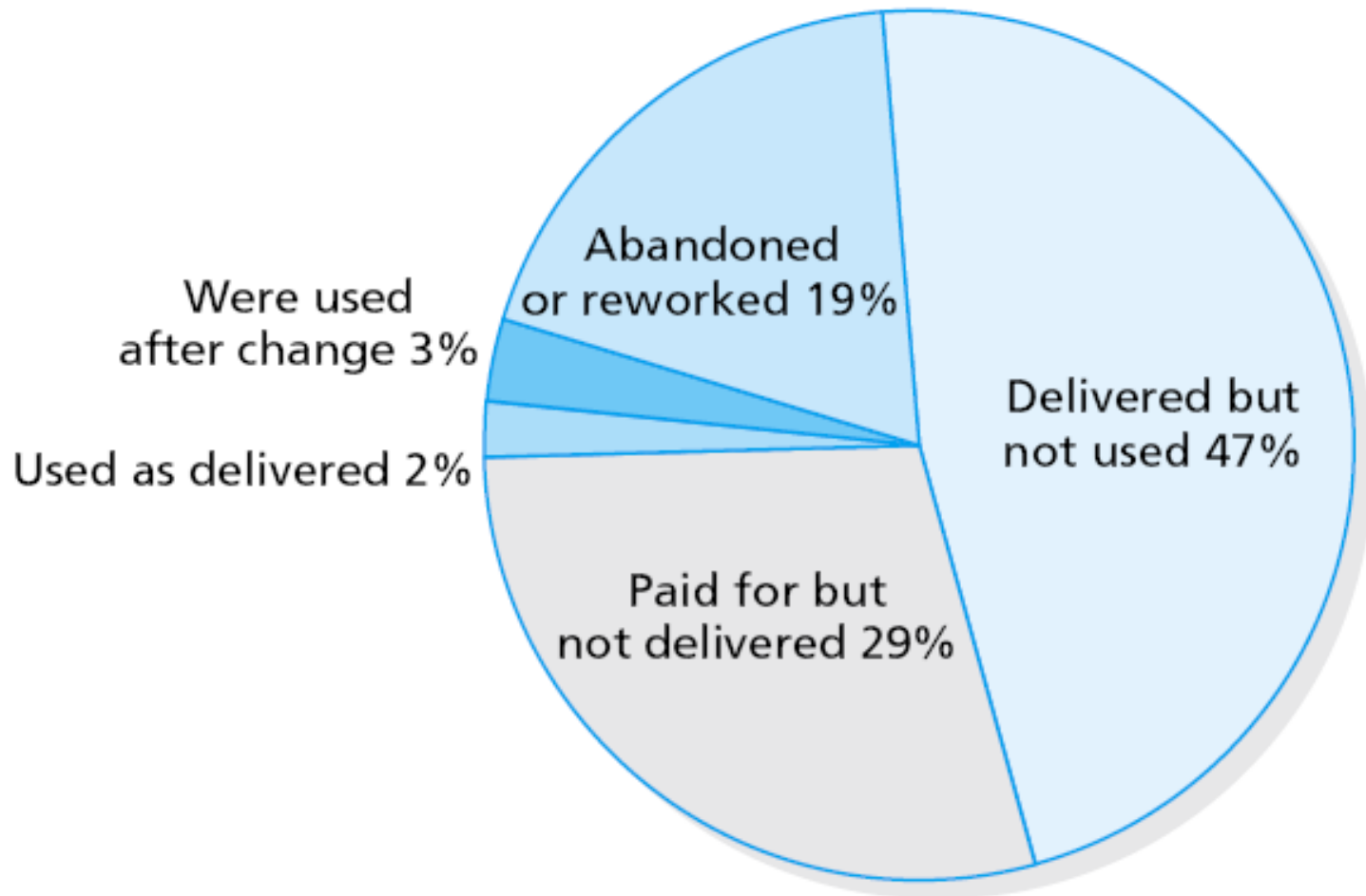


图 1.1 硬件/软件成本变化趋势



■ 软件需求分析不够充分，用户不满意“已经完成”的软件系统



■软件质量难于保证

- ◆1963 年，美国用于控制火星探测器的计算机软件中的一个“,”号被误写为“.”，而致使飞往火星的探测器发生爆炸，造成高达数亿美元的损失。
- ◆1966年，IBM 360 机的操作系统。花费 5000 人一年的工作量，写了近 1 万行代码。错误百出，每次的新版本就是从前一版本中找 1000 个程序错误而修正的结果。

● 千年虫问题

20世纪90年代末，由于许多早期编程中使用了两位数字表示年份（如99代表1999年），导致在跨世纪时（从1999年到2000年），大量软件和系统无法正确识别和处理年份，引发了所谓的“千年虫”问题。

● 波音737MAX软件缺陷

2018年和2019年，两架波音737MAX飞机相继发生坠机事故，导致数百人丧生。调查发现，飞机的自动防失速系统（MCAS）存在软件缺陷。

● 亚马逊Web服务宕机

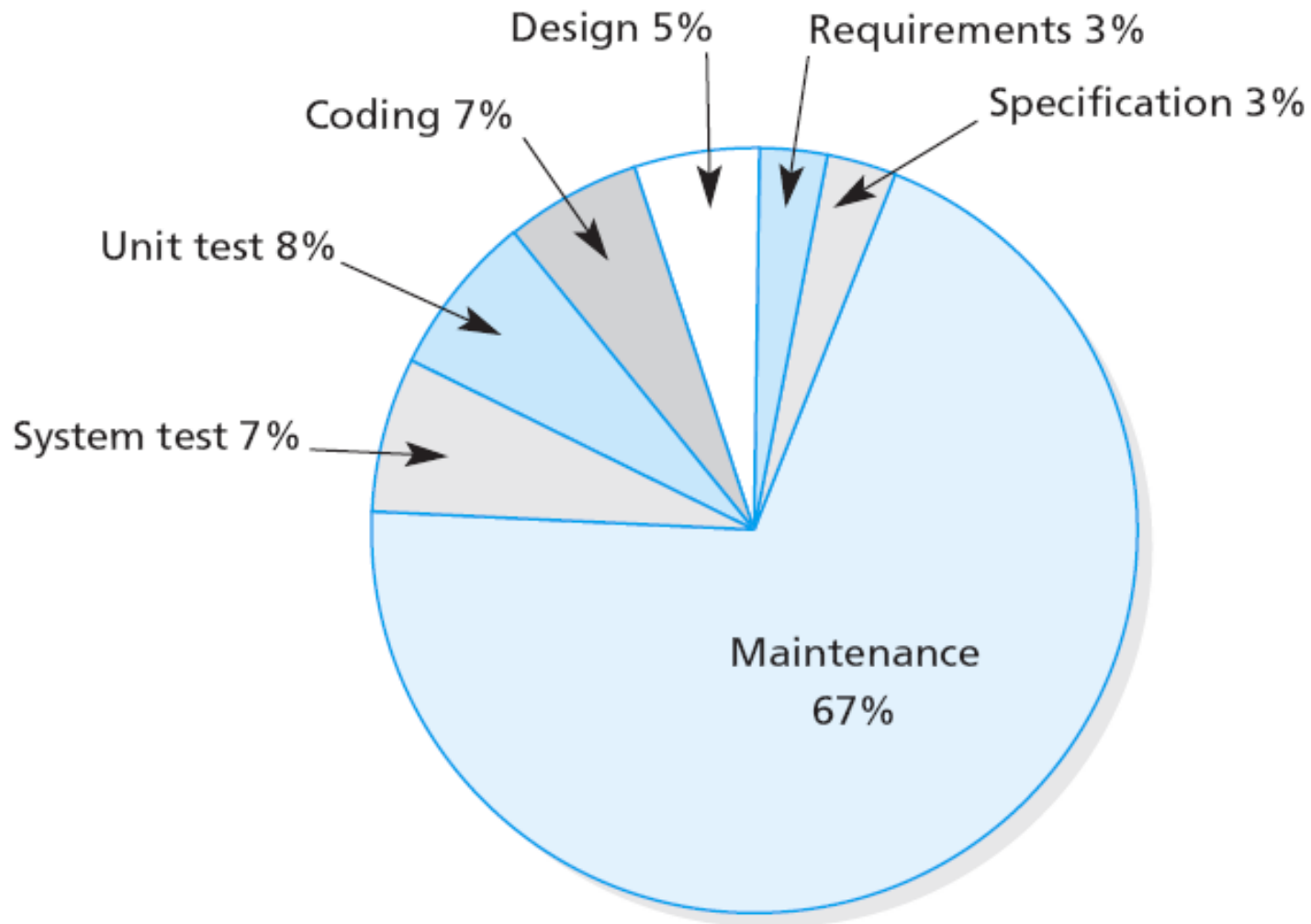
2017年，亚马逊Web服务（AWS）的一个主要数据中心发生宕机，导致许多依赖AWS的网站和服务中断。

● Equifax数据泄露

2017年，信用评级机构Equifax的一个安全漏洞导致其数据库中数百万人的敏感信息被泄露。

■软件维护困难

◆通常没有保留适当的文档资料



产生软件危机的原因

- 与软件本身的特点有关
- 与软件开发与维护的方法不正确有关



● 软件与一般程序不同

- 软件远比一般程序规模庞大，复杂性高。
- 大型软件开发既有技术问题，还有社会问题。
 - ◆ 开发团队成员分工合作
 - ◆ 技术与管理的矛盾
 - ◆ 软件开发人员对软件应用的领域知识的了解

| 类别 | 参加人员数 | 研制期限 | 产品规模（源程序行数） |
|-----|-----------|--------|-------------|
| 微型 | 1 | 1~4 周 | 0.5k |
| 小型 | 1 | 1~6 月 | 1k~2k |
| 中型 | 2~5 | 1~2 年 | 5k~50k |
| 大型 | 5~20 | 2~3 年 | 50k~100k |
| 甚大型 | 100~1000 | 4~5 年 | 1M(=1000k) |
| 极大型 | 2000~5000 | 5~10 年 | 1M~10M |

● 软件开发和维护方法中存在的问题

- 缺乏有效的、系统的技术手段和管理方法

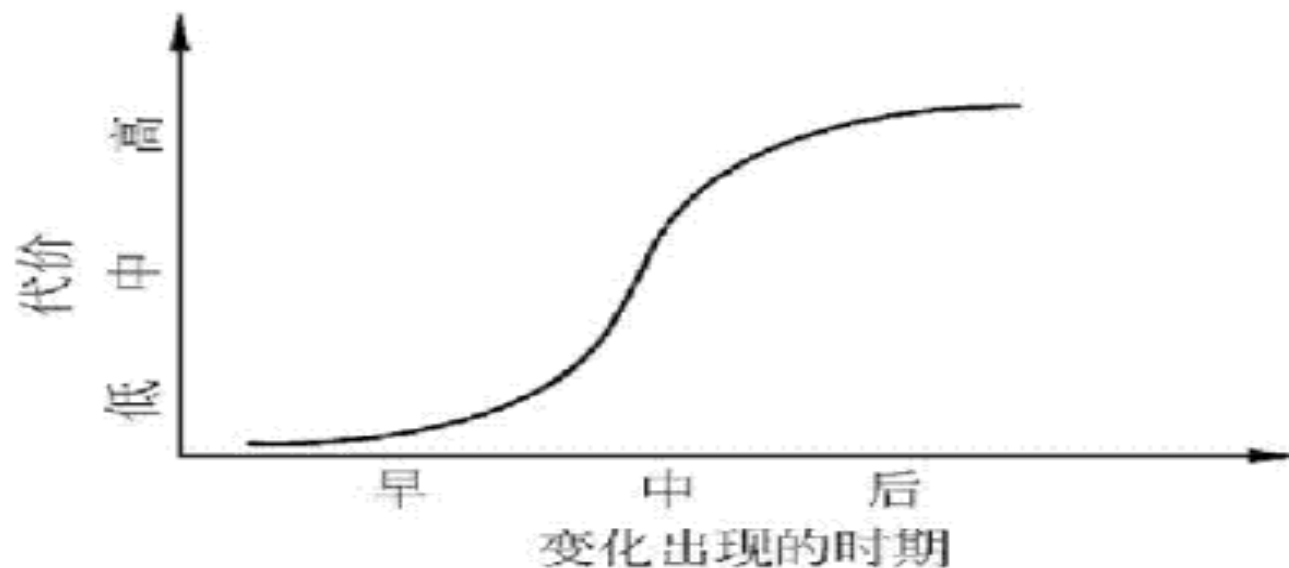
- 用户和软件开发人员的理解鸿沟

- 错误的认识和做法

 - ◆ 忽视软件需求分析的重要性

 - ◆ 认为软件开发就是写程序并设法使之运行

 - ◆ 轻视软件维护等



消除软件危机的途径

- 彻底消除“软件就是程序”的错误观念。
 - 一个软件必须由一个完整的配置组成，事实上，软件是程序、数据及相关文档的完整集合。文档是开发、使用和维护程序所需要的图文资料。
- 充分认识到软件开发是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目，不是个人独立的劳动。
- 推广和使用在实践中总结出来的软件开发的成功技术和方法。
- 开发和使用更好的软件工具



1.2 软件工程

软件工程的介绍
软件工程方法学

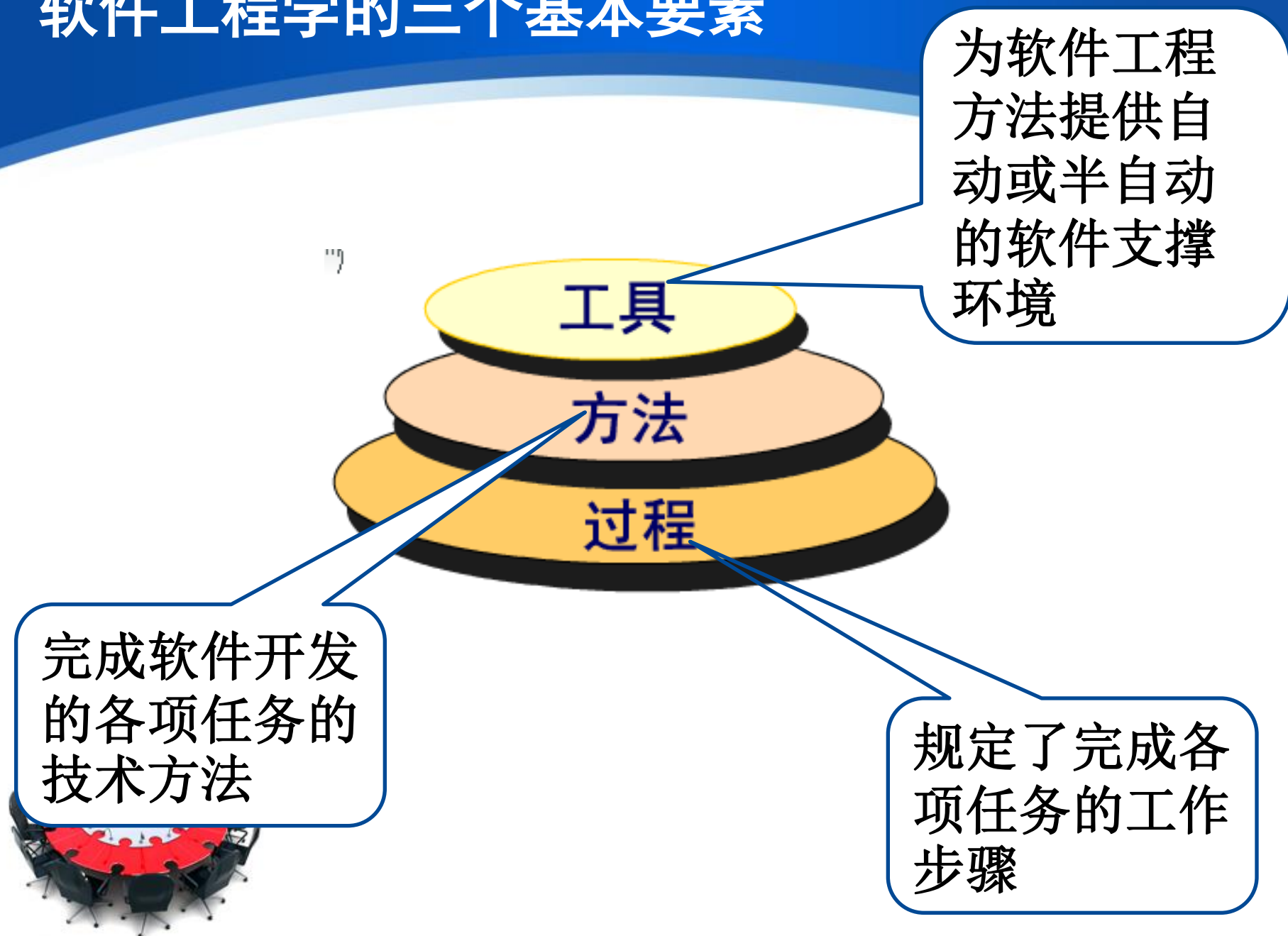


软件工程的定义

- 采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的**管理技术**和当前能够得到的最好的技术方法结合起来，**以经济地开发出高质量的软件并有效地维护它**，这就是软件工程。



软件工程学的三个基本要素



两种使用最广泛的软件工程方法学

- 传统方法学

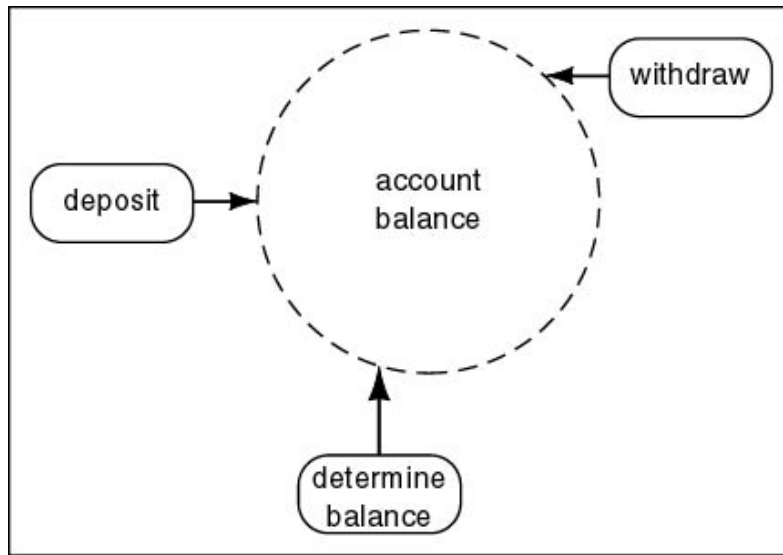
- 也称为生命周期方法学或结构化范型
- 采用结构化技术(结构化分析、设计和实现)
- 结构化范型要么面向行为，要么面向数据

- 面向对象方法学

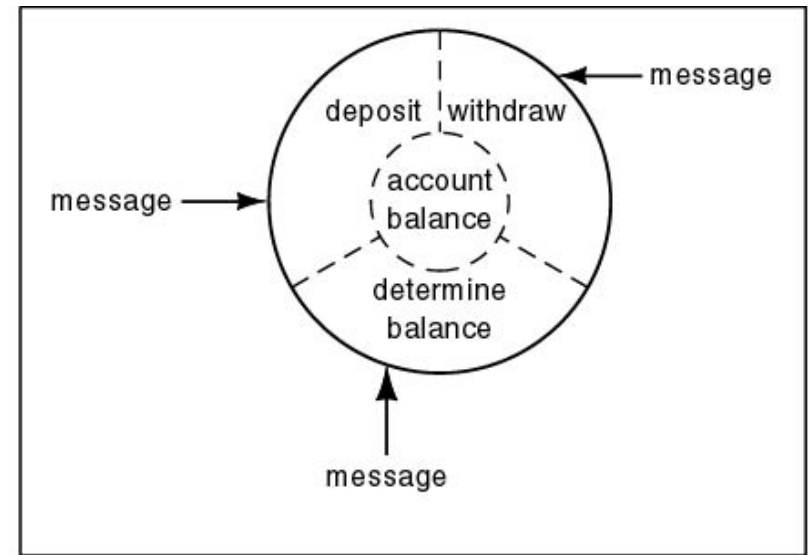
- 把数据和行为看成同等重要，以数据为主线，把数据和对数据的操作紧密地结合
- 面向对象方法=对象+类+继承+用消息通信



软件工程



(a)



(b)

传统方法与面向对象方法比较



1.3 软件生命周期

软件产品或系统一系列相关活动的全周期



软件生命周期的各个阶段

■ 软件定义

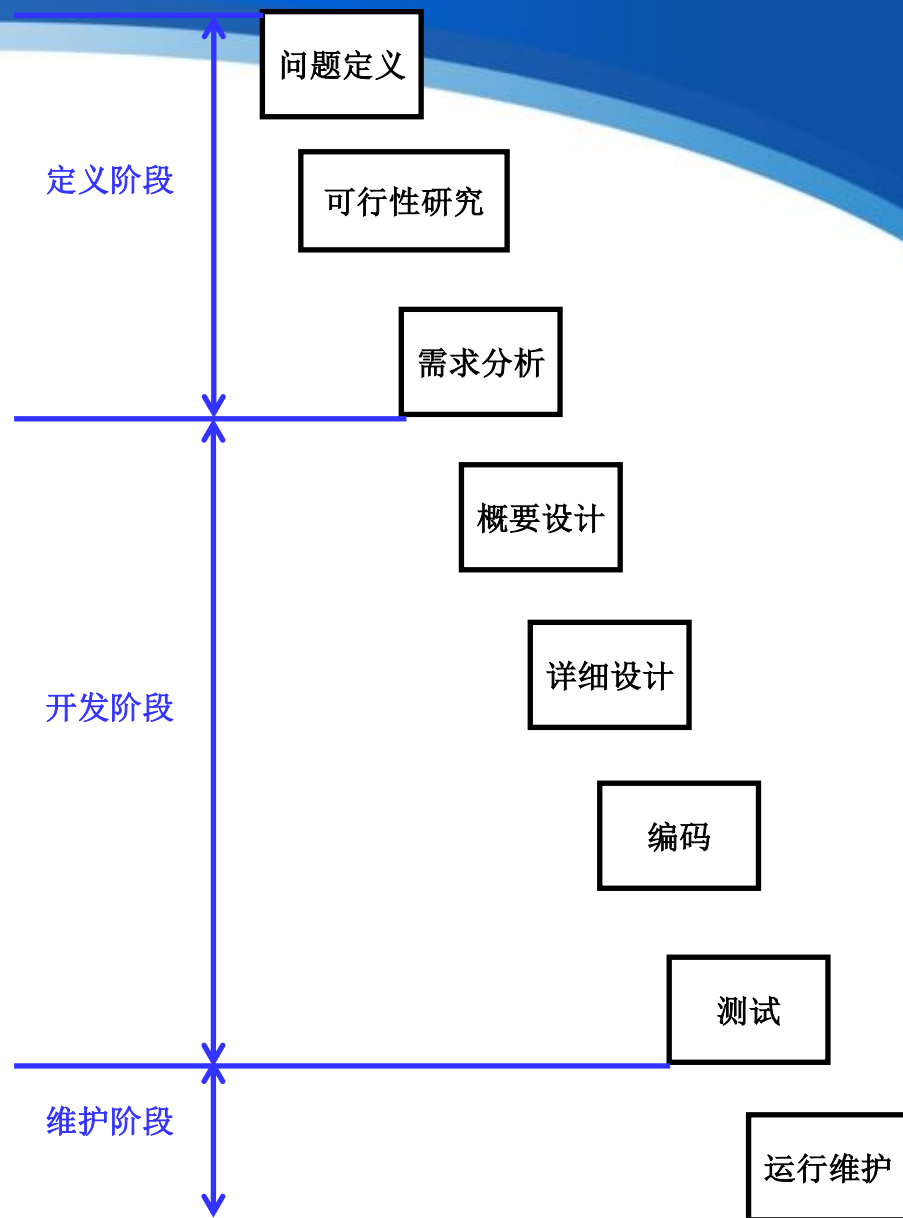
- 总目标、可行性；导出实现策略及系统功能；估计资源和成本，并且制定工程进度表。

■ 软件开发

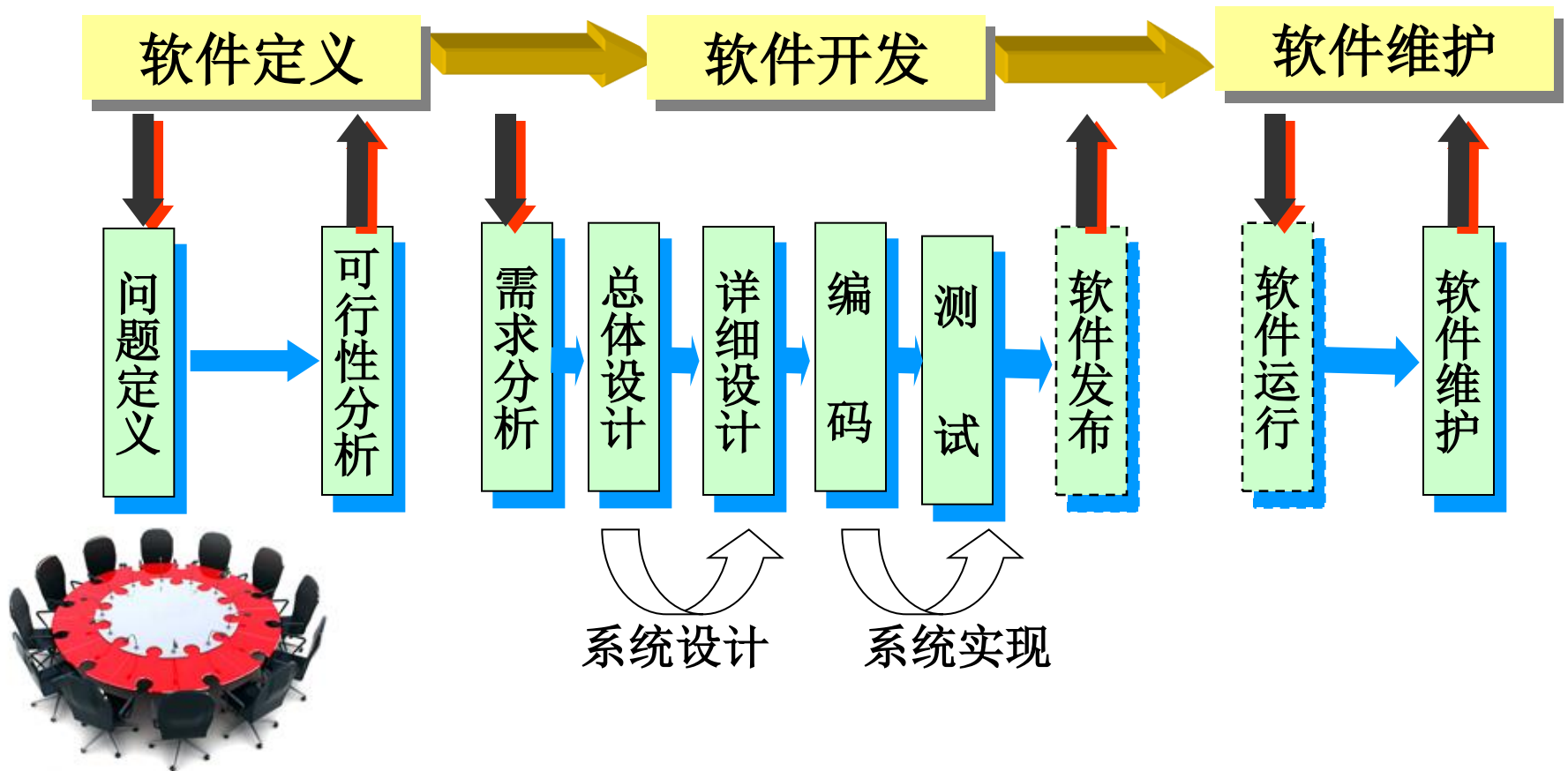
- 具体设计和实现

■ 软件维护

- 使软件持久地满足用户的需要



软件生命周期



软件生命周期 (续1)

1. 问题定义

- “要解决的问题是什么?”
- 确定用户要求解决的性质、工程的目标和规模。

2. 可行性研究

- “对于上一个阶段所确定的问题有行得通的解决 办法吗?”
- 经济可行性、技术可行性、法律可行性、不同的 方案

3. 需求分析

- “为了解决这个问题，目标系统必须做什么”
- 确定系统必须具有的功能和性能，系统要求的运行环境，并且预测系统发展的前景。
- 规格说明书(specification)



软件生命周期 (续2)

4. 总体设计（概要设计）

- “概括地说，应该怎样实现目标系统?”
- 设计出实现目标系统的几种可能的方案。推荐一个最佳方案。

5. 详细设计

- “应该怎样具体地实现这个系统呢?”
- 设计出程序的详细规格说明。

6. 编码和单元测试

- 写出正确的容易理解、容易维护的程序模块
- 仔细测试编写出的每一个模块。



软件生命周期 (续3)

7. 综合测试

- 集成测试和验收测试，现场测试或平行运行

8. 软件维护

- 使系统持久地满足用户的需要。
- 改正性维护，适应性维护，完善性维护，预防性维护。



1.4 软件过程

什么是软件生命周期模型
典型的软件生命周期模型



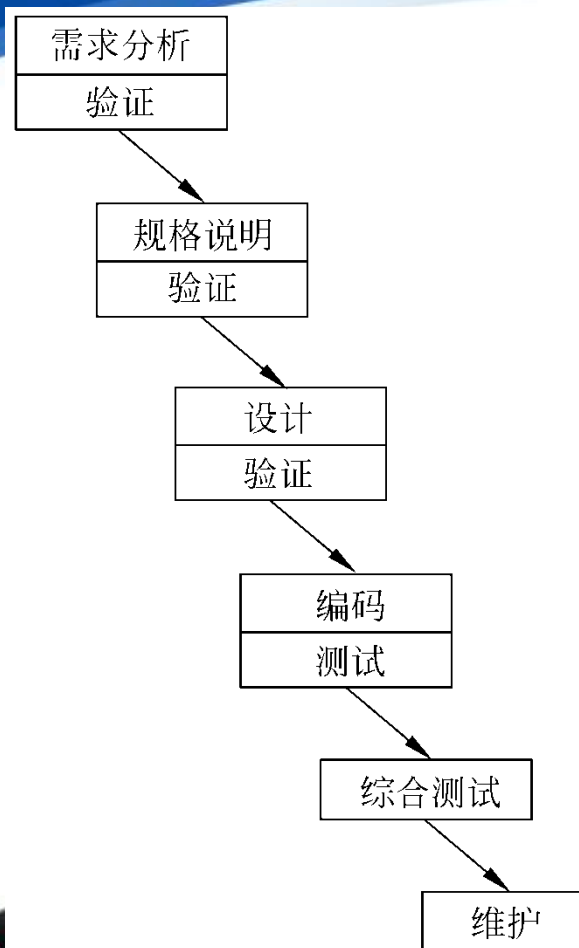
- 生命周期模型规定了把生命周期划分成哪些阶段及各个阶段的执行顺序，因此，也称为过程模型。

- 任务框架，各项任务的工作步骤
- “什么人”、“什么时候”、“做什么”以及“怎样做”

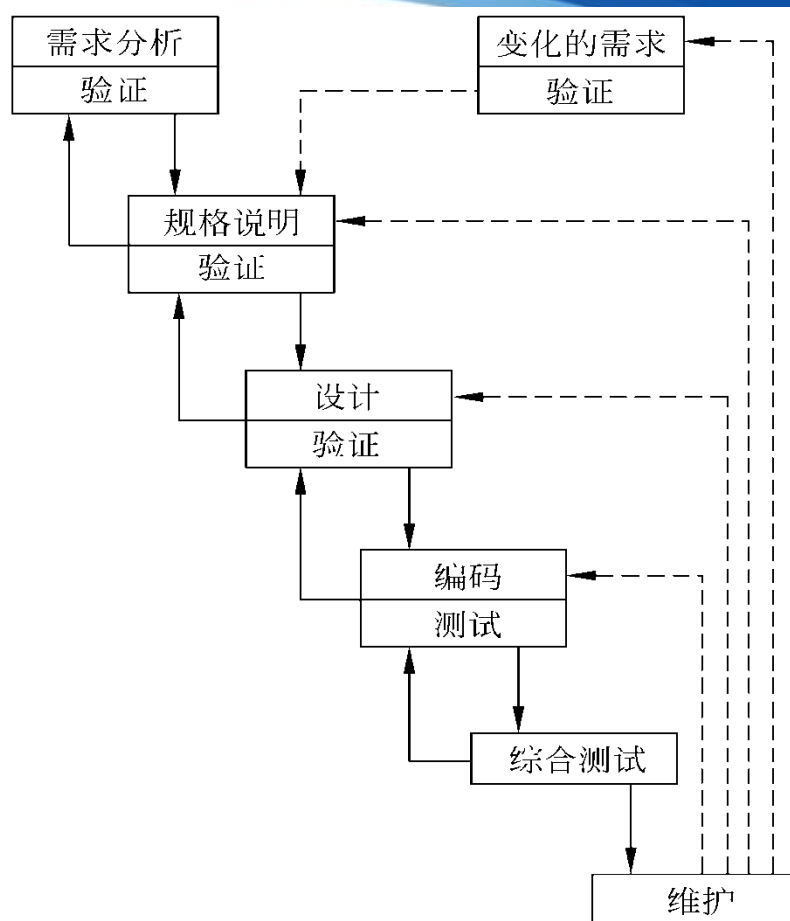
- 典型的过程模型

- 瀑布模型(Waterfall model)
 - 快速原型开发模型(Rapid Prototyping model)
 - 增量模型(Incremental model)
 - 螺旋模型(Spiral model)
 - 喷泉模型(Fountain model)
 - 极限编程 (XP, eXtreme Programming)
 - RUP (Rational Unified Process)

软件过程：瀑布模型



理想的瀑布模型



实际的瀑布模型



● 瀑布模型的特点

- 阶段间具有顺序性和依赖性

- 推迟实现的观点

- 质量保证的观点(文档驱动)

 - ◆ 每个阶段都必须完成规定的文档

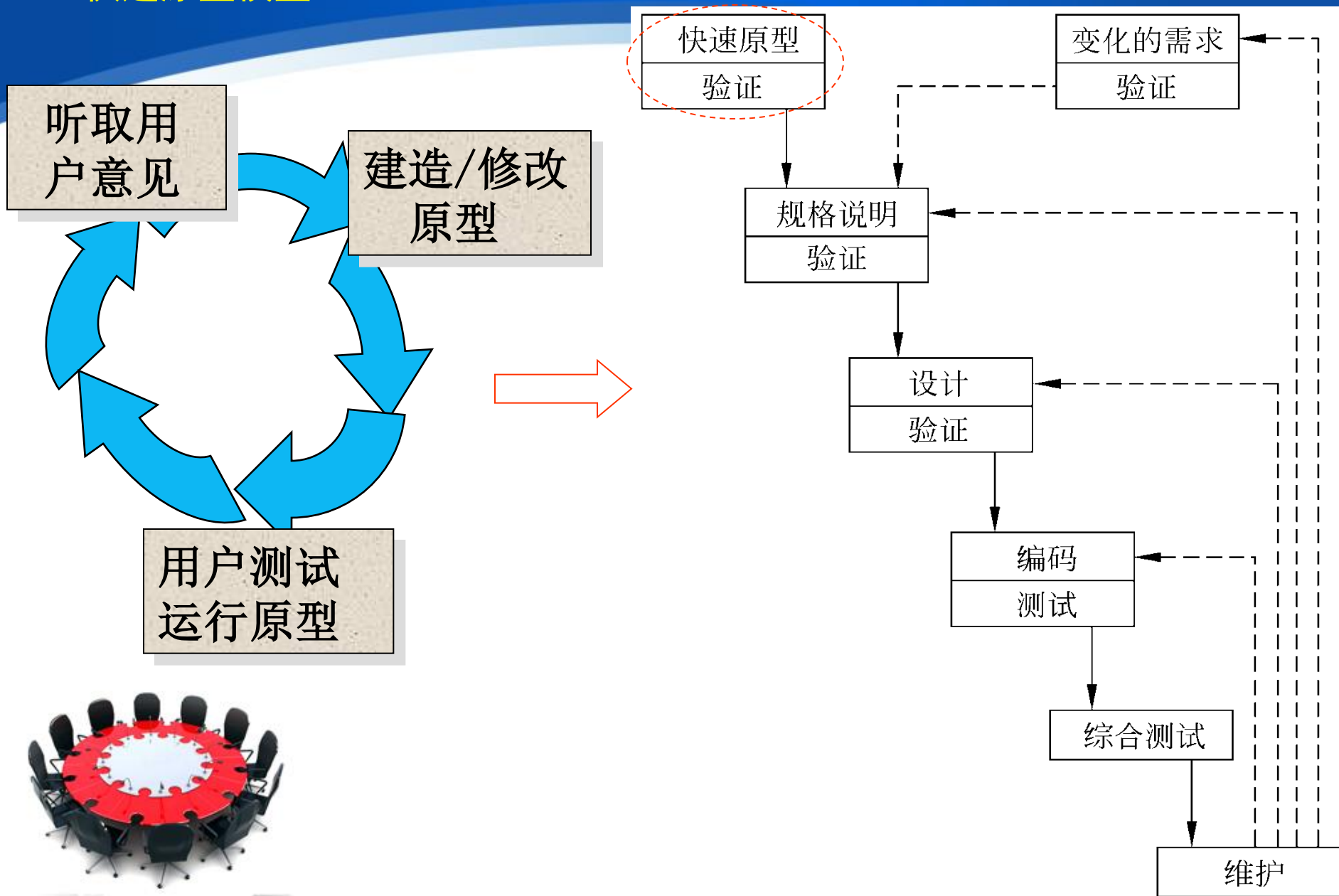
 - ◆ 每个阶段结束前都要对所完成的文档进行评审

● 瀑布模型的缺点

- 开发过程一般不能逆转，否则代价太大。软件的实际情况必须到项目开发后期才能看到。

- 无法解决软件需求不明确或不准确的问题；可能导致最终开发的产品不能真正满足用户需要。

...快速原型模型

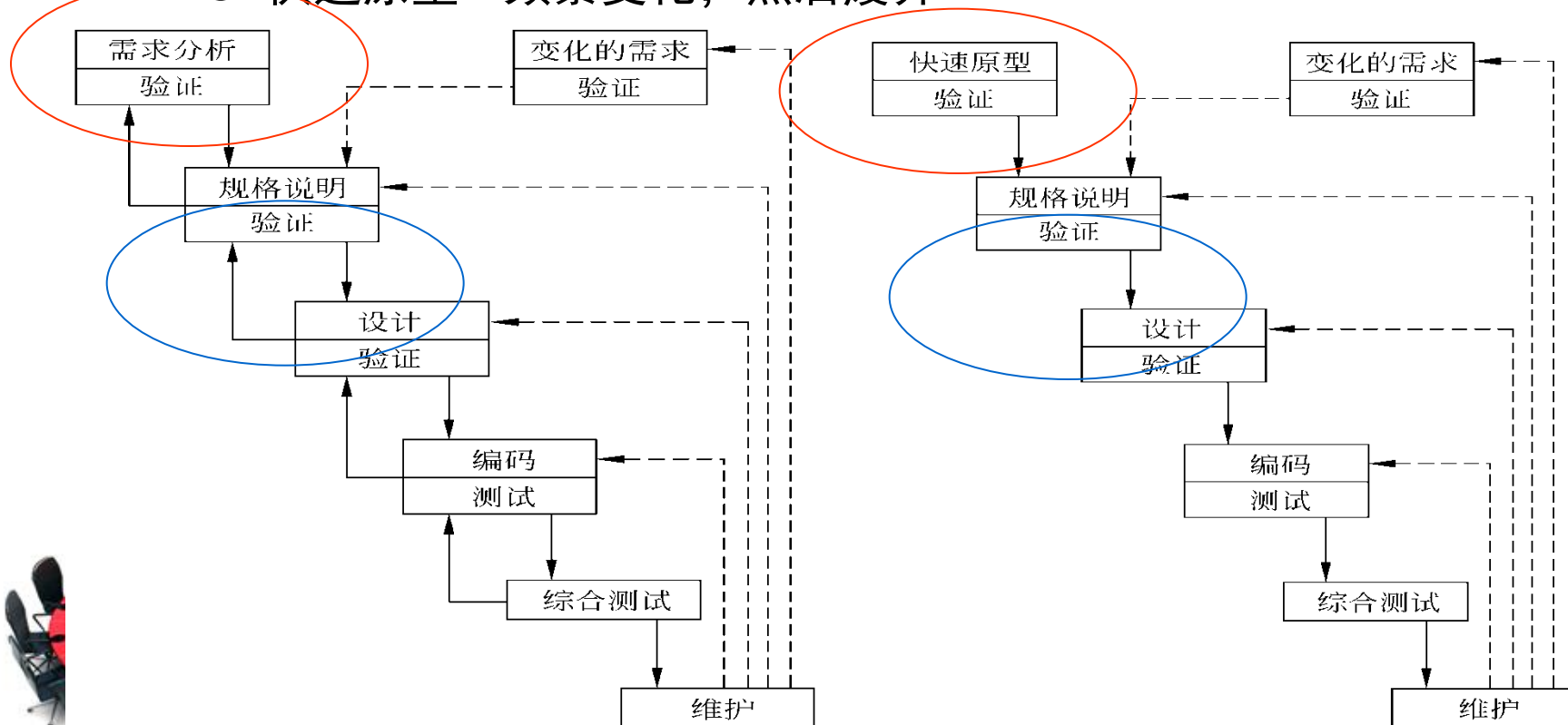


软件过程：快速原型模型

■ 比较

⑩ 瀑布模型—试图一次就获得正确的产品

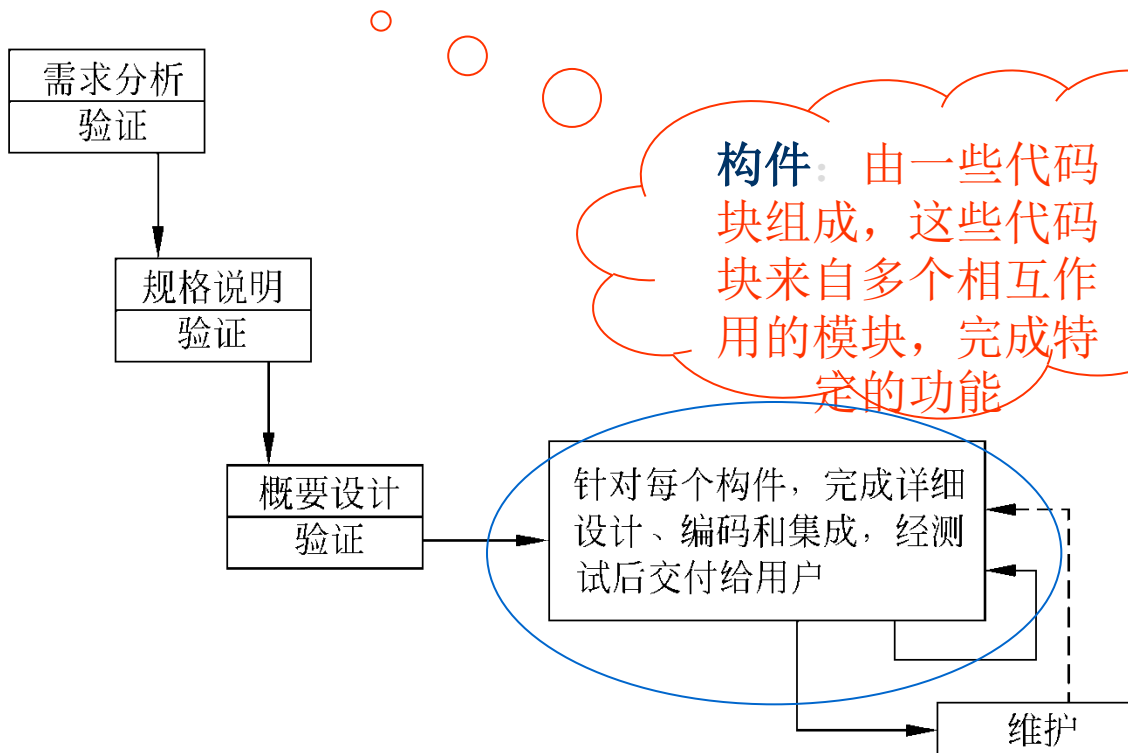
⑩ 快速原型—频繁变化，然后废弃



软件过程：增量模型

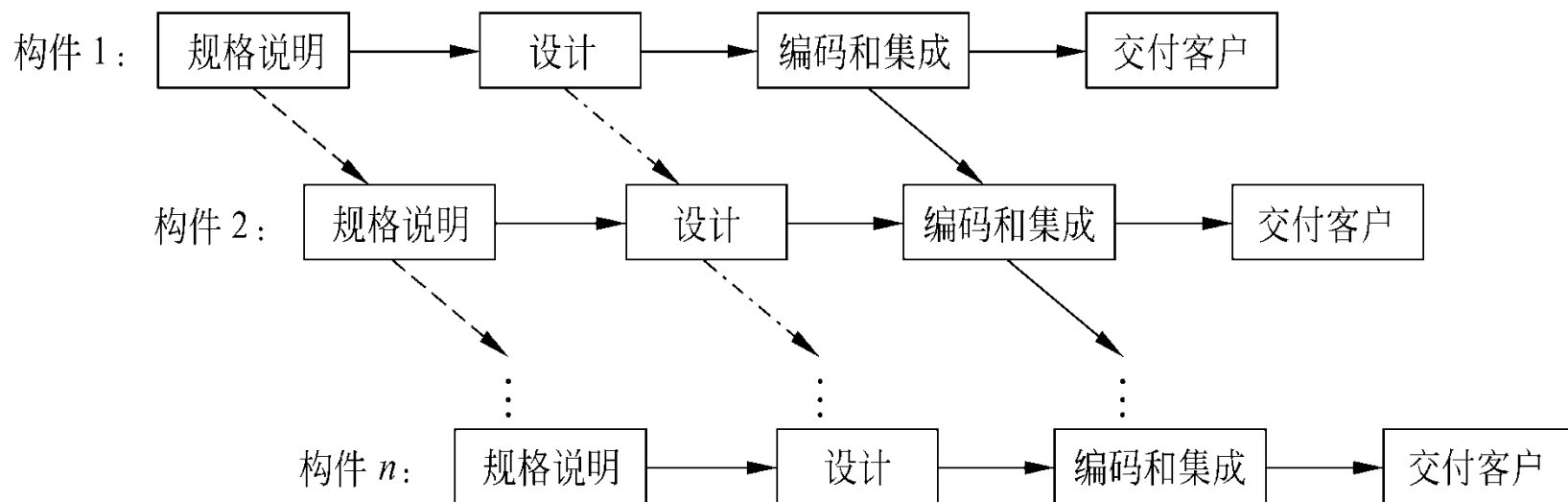
也称：渐增模型

把软件产品作为一系列增量构件来设计、编码、集成和测试。



软件过程：增量模型

风险更大的增量模型



软件过程：螺旋模型

1988年，Barry Boehm正式发表了软件系统开发的“螺旋模型”，它将瀑布模型和快速原型模型结合起来，强调了其他模型所忽视的风险分析

- 简化版本：瀑布模型+风险分析

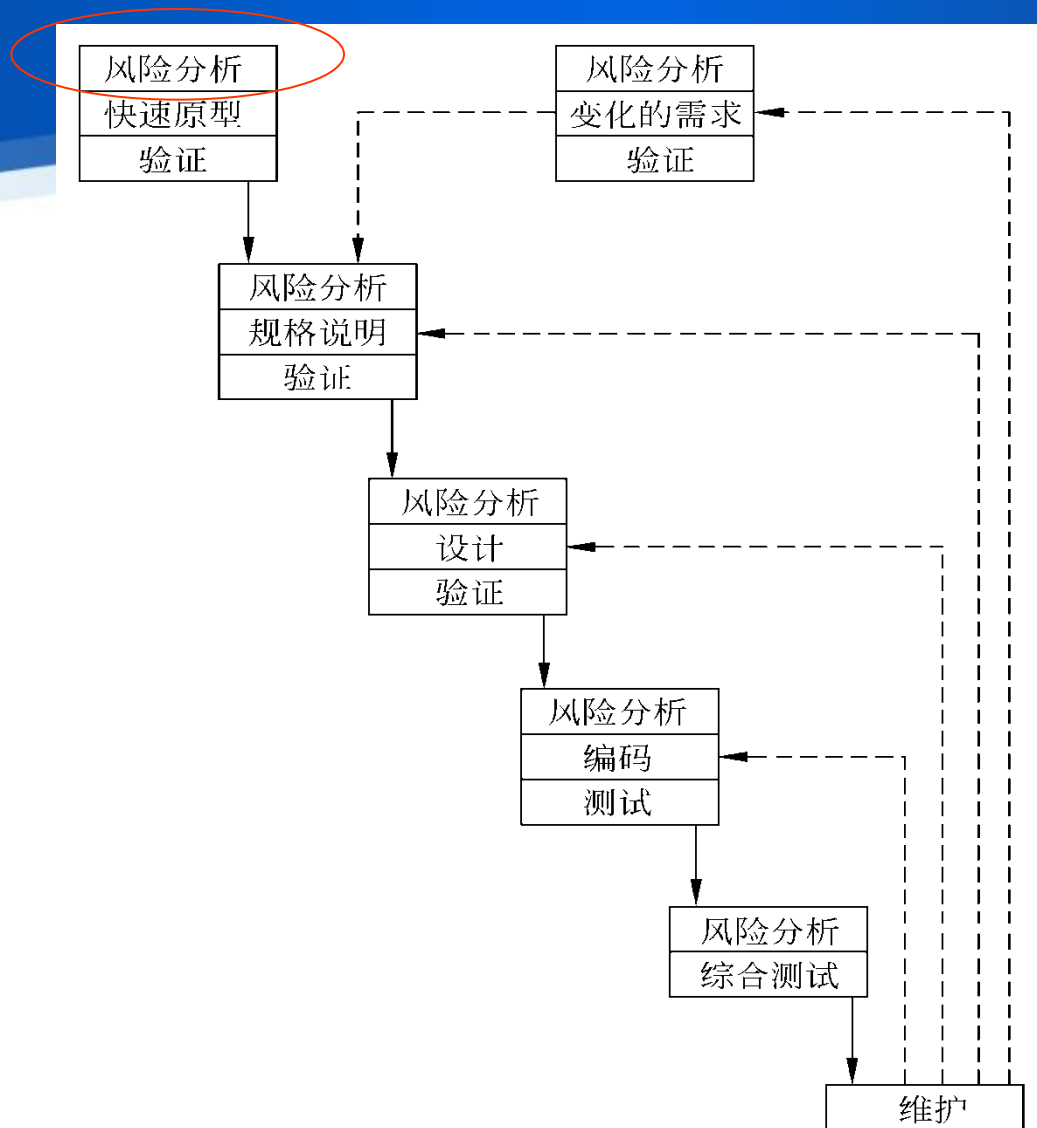
- ⑩ 每个阶段之前

- 确定目标，可供选择的办法及其限制条件
 - 风险分析

- ⑩ 每个阶段之后

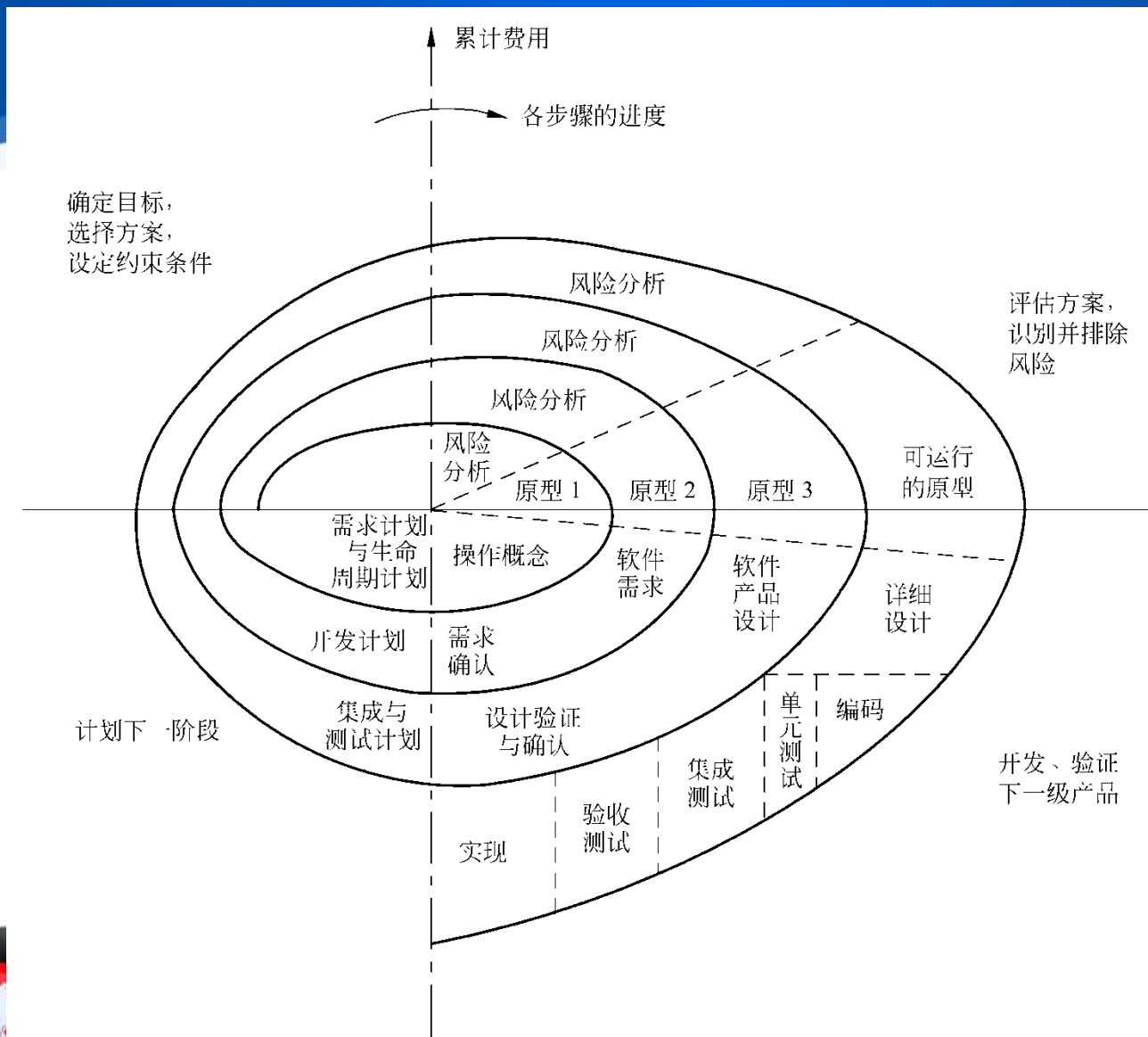
- 评估
 - 计划下一阶段





简化的螺旋模型





完整的螺旋模型

