



# 软件工程



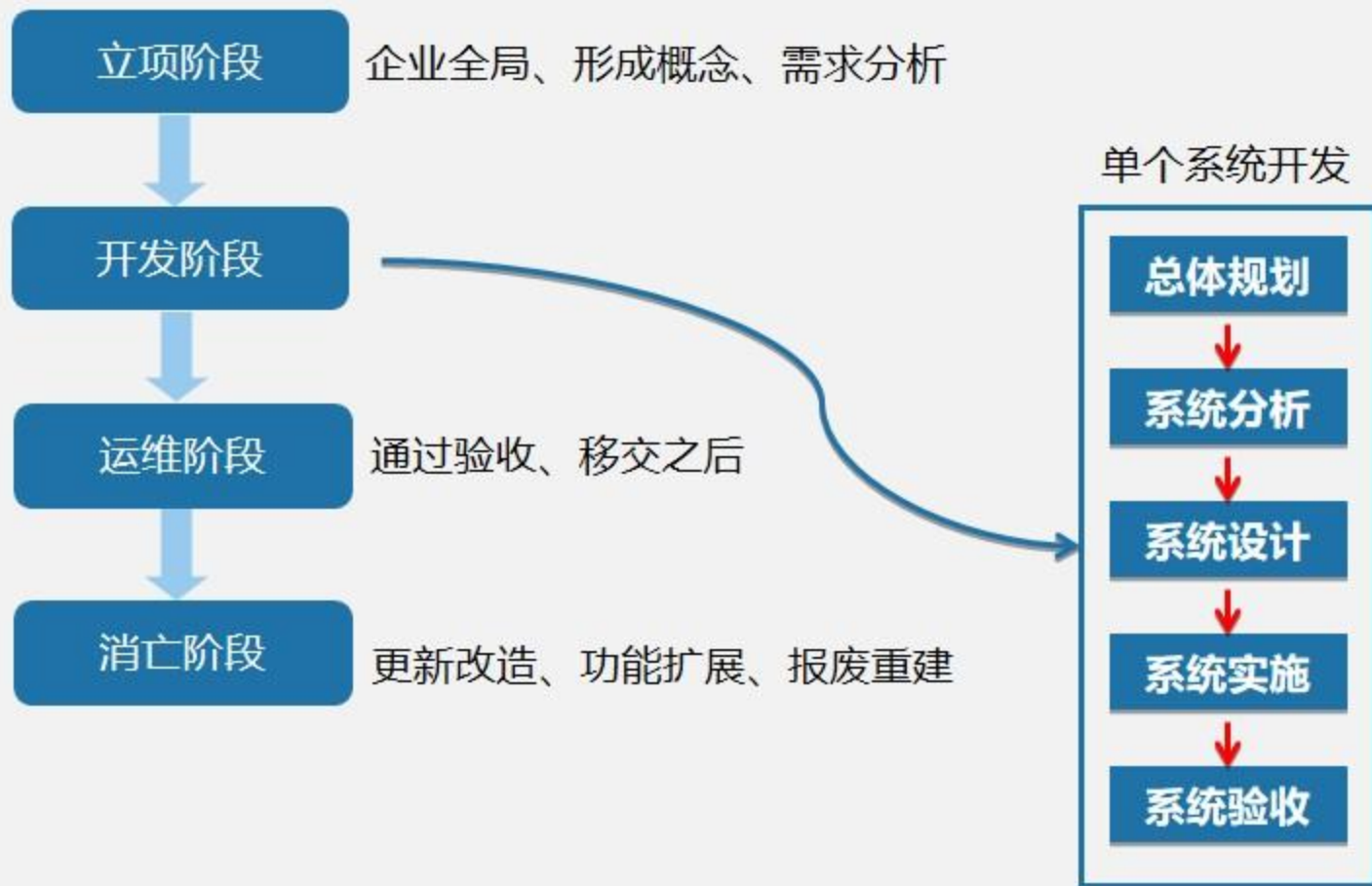
## 课程内容提要



- 信息系统的生命周期 (★★)
- 软件开发方法 (★★★)
- 软件开发模型 (★★★★)
- 逆向工程 (★)



## 信息系统的生命周期





## 信息系统的生命周期







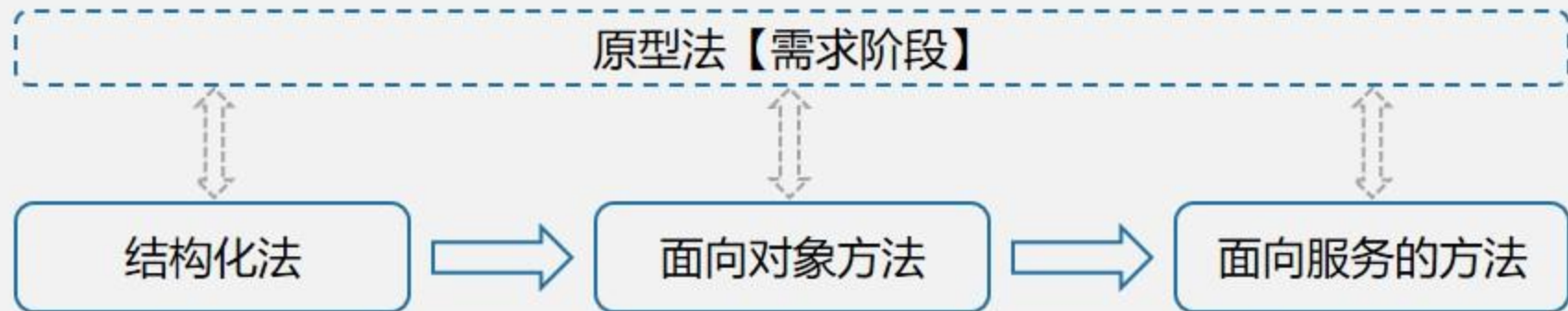
## 软件开发方法



针对需求不明确

按功能分：水平原型（界面）、垂直原型（复杂算法）

按最终结果分：抛弃式原型、演化式原型



自顶向下，逐步分解求精  
严格分阶段，阶段产出标准化  
应变能力差

自底向上  
阶段界限不明  
更好应变、更好复用  
符合人们的思维习惯

粗粒度、松耦合  
标准化和构件化  
抽象级别：  
操作【低】 -> 服务【中】 -> 业务流程【高】



## 软件开发方法



### 其他软件开发方法

形式化方法

净室软件工程【受控污染级别的环境】  
数学模型化  
所有东西均可证明/验证，而不是测试

统一过程方法【UP】

敏捷方法



## 软件开发方法



软件方法学是以软件开发方法为研究对象的学科。其中，（ ）是先对最高层次中的问题进行定义、设计、编程和测试，而将其中未解决的问题作为一个子任务放到下一层次中去解决。（ ）是根据系统功能要求，从具体的器件、逻辑部件或者相似系统开始，通过对其进行相互连接、修改和扩大，构成所要求的系统。（ ）是建立在严格数学基础上的软件开发方法。

A 面向对象开发方法

B 形式化开发方法

C 非形式化开发方法

D 自顶向下开发方法

A 自底向上开发方法

B 形式化开发方法

C 非形式化开发方法

D 原型开发方法

A 自底向上开发方法

B 形式化开发方法

C 非形式化开发方法

D 自顶向下开发方法



## 软件开发方法 – 软件开发模型

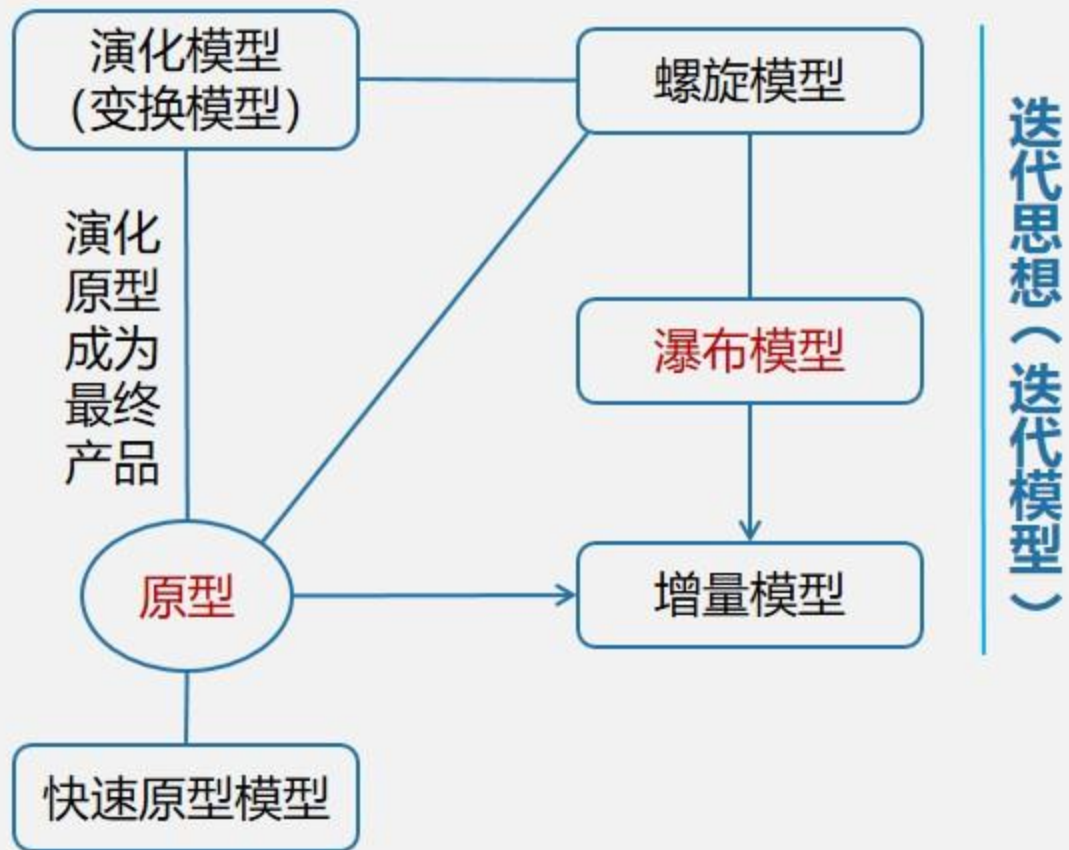


- ✓ **瀑布模型**
- ✓ 演化模型
- ✓ 增量模型
- ✓ **螺旋模型**
- ✓ **原型模型**
- ✓ 喷泉模型
- ✓ V模型
- ✓ 迭代模型/迭代开发方法
- ✓ 快速应用开发
- ✓ 构件组装模型/基于构件的开发方法
- ✓ **统一过程/统一开发方法**
- ✓ **敏捷开发方法**
- ✓ 模型驱动的开发方法
- ✓ **基于架构的开发方法**



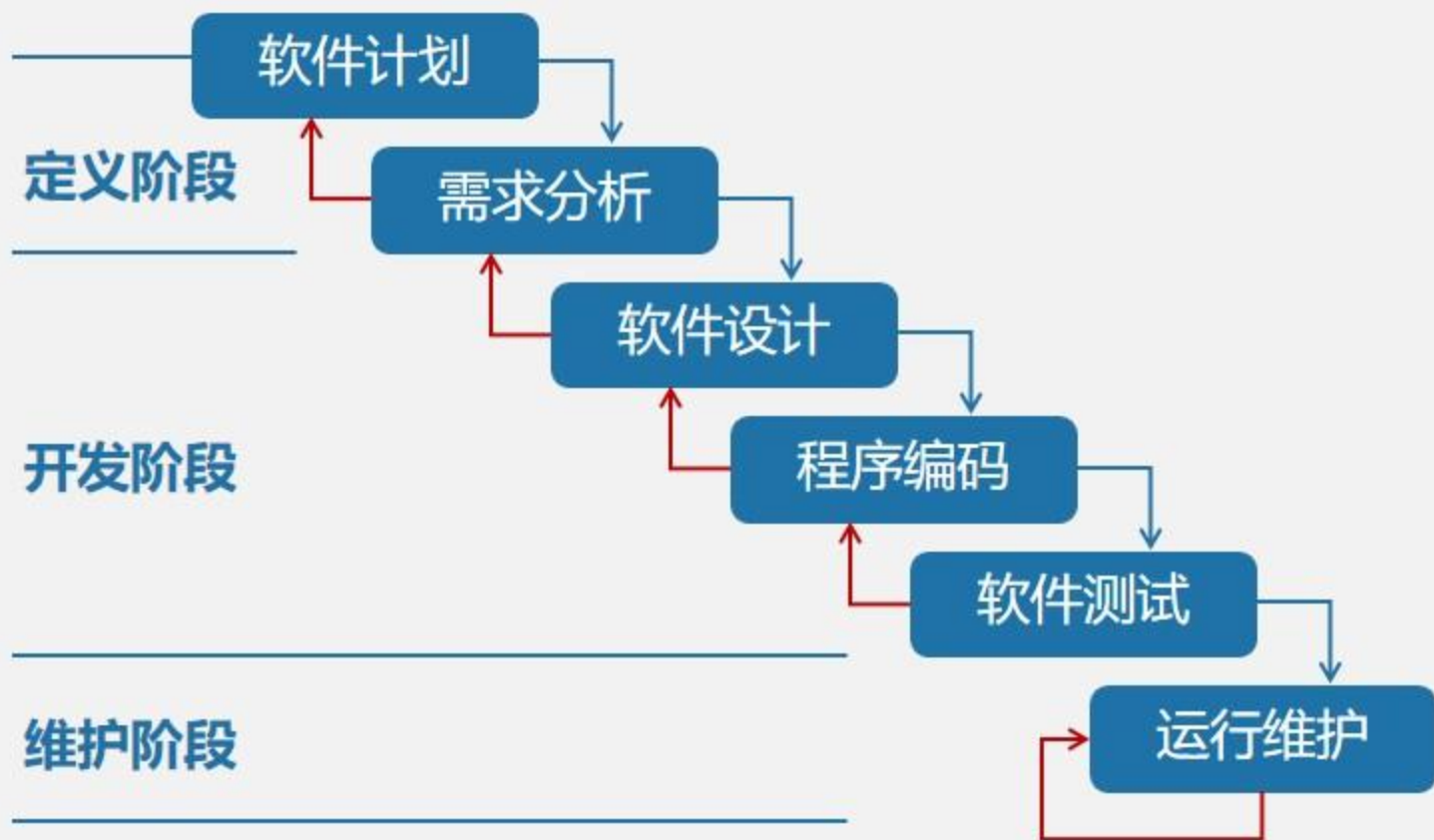


## 软件开发方法 – 软件开发模型 – 其他经典模型



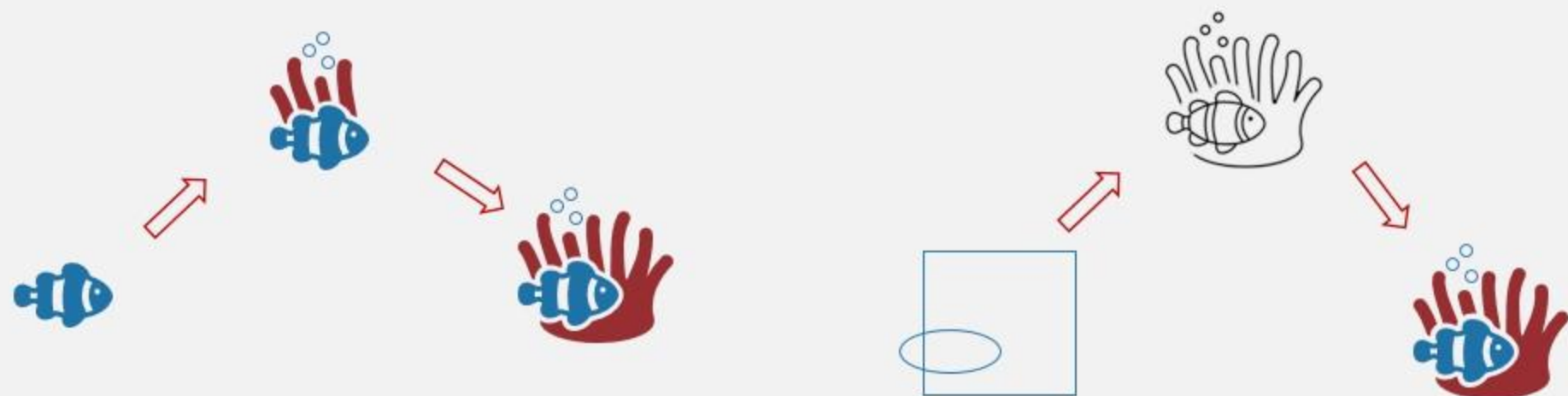


## 软件开发方法 – 软件开发模型 – 瀑布模型





## 软件开发方法 – 软件开发模型 – 迭代与增量



增量型

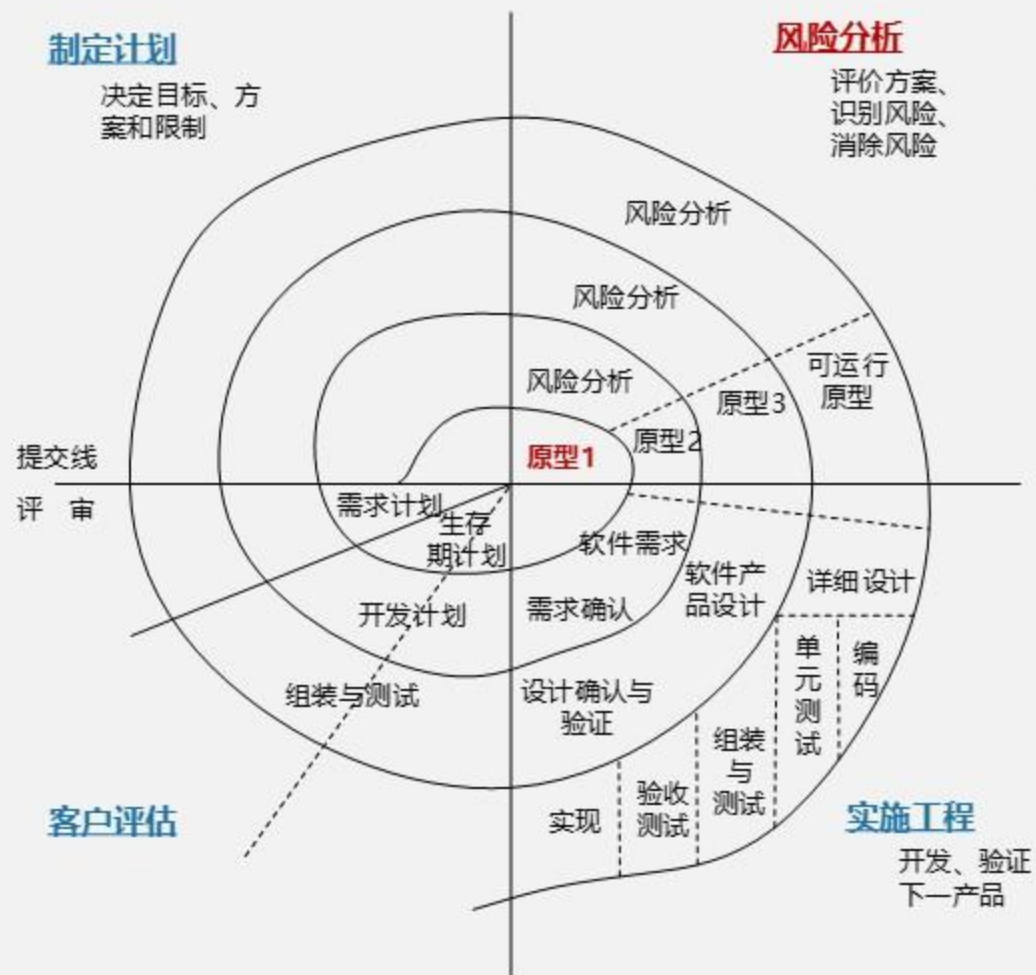
迭代型

增量与迭代

【UP / 敏捷】



## 软件开发模型 - 增量模型与螺旋模型



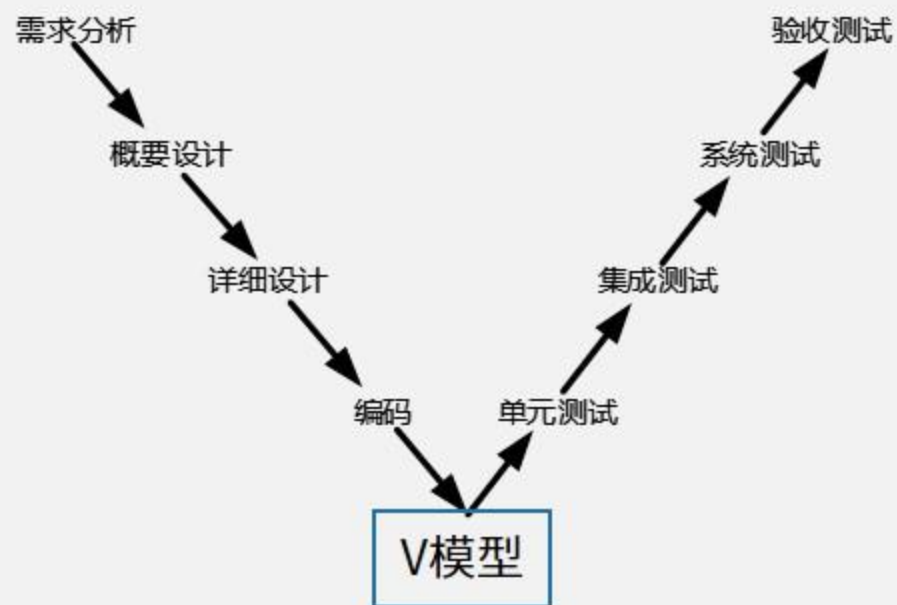
以原型为基础+瀑布模型

考虑了风险问题

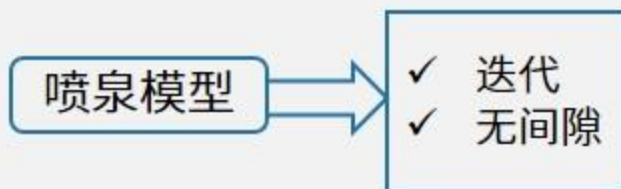




## 软件开发方法 – 软件开发模型 – 其他模型



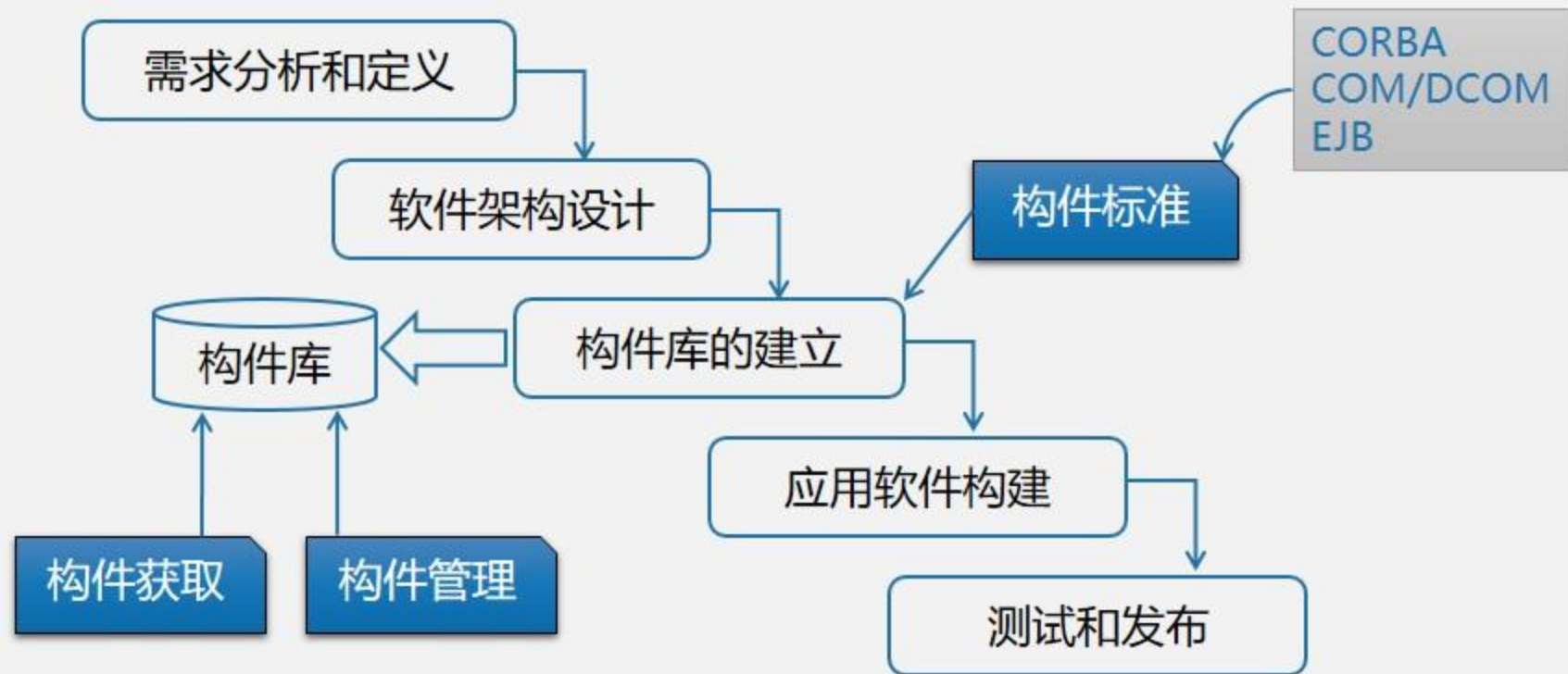
测试贯穿于始终



早期著名的面向对象模型

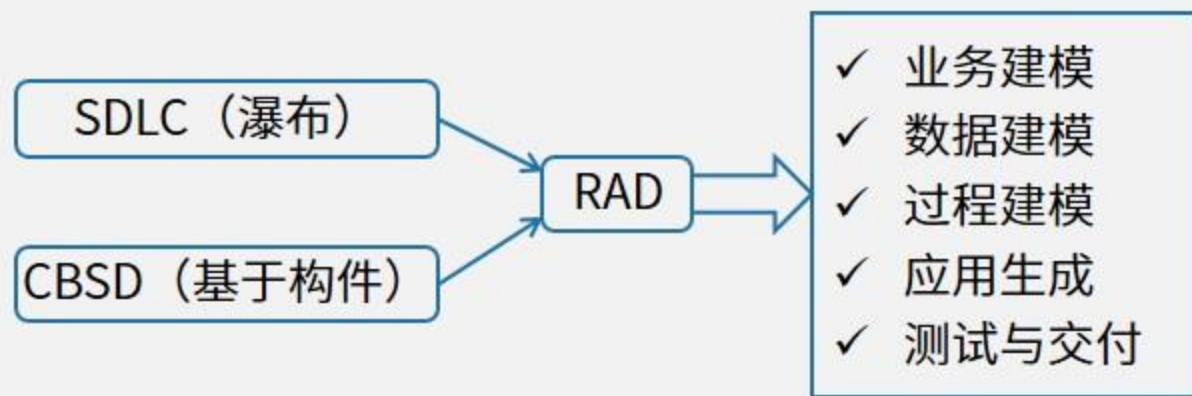


## 软件开发方法 – 软件开发模型 – 构件组装模型



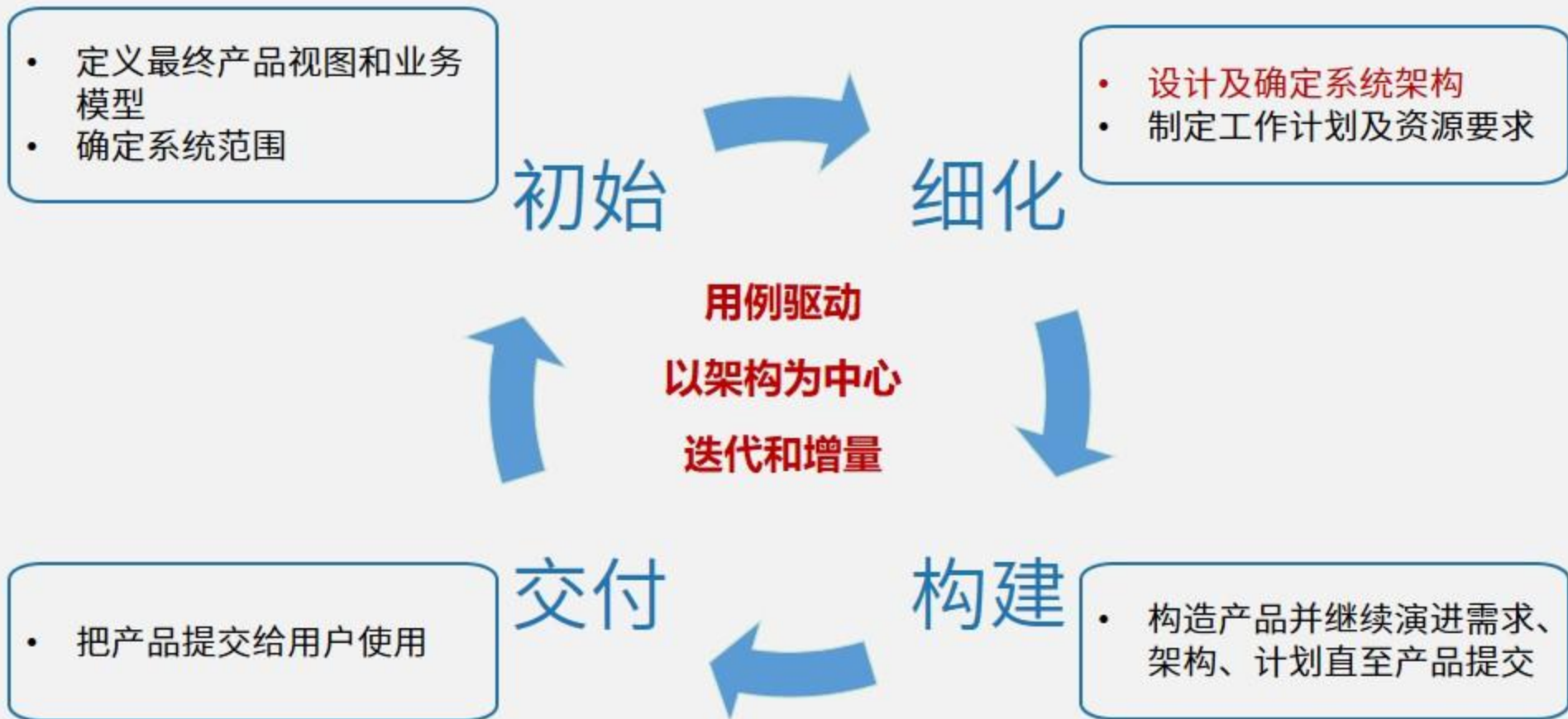


## 软件开发模型 – 快速应用开发模型 (RAD)





## 软件开发方法 – 软件开发模型 – 统一过程







## 软件开发方法 – 软件开发模型



( ) 把整个软件开发流程分成多个阶段，每一个阶段都由目标设定、风险分析、开发和有效性验证以及评审构成。

- A 原型模型
- B 瀑布模型
- C 螺旋模型
- D V模型



## 软件开发方法 – 软件开发模型



基于RUP的软件过程是一个迭代过程。一个开发周期包括初始、细化、构建和移交四个阶段，每次通过这四个阶段就会产生一代软件，其中建立完善的架构是（ ）阶段的任务。采用迭代式开发，（ ）。

- A 初始
- B 细化
- C 构建
- D 移交

- A 在每一轮迭代中都要进行测试与集成
- B 每一轮迭代的重点是对特定的用例进行部分实现
- C 在后续迭代中强调用户的主动参与
- D 通常以功能分解为基础



## 软件开发方法 – 软件开发模型 – 敏捷方法



### 敏捷宣言

- ✓ 个体和交互胜过过程和工具
- ✓ 可工作的软件胜过大量的文档
- ✓ 客户合作胜过合同谈判
- ✓ 响应变化胜过遵循计划

如：结构化方法

无软件开发方法

无序，不可控

传统软件开发方法

敏捷方法

传统软件开发方法	敏捷方法
预设性的	适应性的
以开发过程为本	以人为本
整体分阶段	增量迭代，小步快跑
	适合小型项目



## 软件开发模型 – 敏捷开发方法



下列关于敏捷方法的叙述中，正确的是（ ）。

- A 敏捷方法强调软件过程与工具胜过个体和交互
- B 敏捷方法的思想是预设性，而不是适应性
- C 敏捷方法尤其适合于开发团队比较大的项目
- D 敏捷方法强调响应变化胜过遵循计划





## 软件开发方法 – 软件开发模型 – 敏捷方法



### 敏捷开发方法 - XP

#### 4大价值观

沟通【加强面对面沟通】

简单【不过度设计】

反馈【及时反馈】

勇气【接受变更的勇气】

#### 12条过程实践规则

简单设计

测试驱动

代码重构

结对编程

持续集成

现场客户

发行版本小型化

系统隐喻

代码集体所有制

规划策略

规范代码

40小时工作机制



## 软件开发模型 – 敏捷开发方法



- ◆ 极限编程 (XP) : 沟通、简明、反馈、勇气, 测试先行。
- ◆ SCRUM: 增量迭代、发布产品的重要性高于一切。
- ◆ 水晶方法: 不同类型项目需要不同方法。
- ◆ 开放式源码: 程序开发人员在地域上分布很广【其它方法强调集中办公】。
- ◆ 特性驱动开发 (FDD) : 编程开发人员分成两类: 首席程序员和“类”程序员。
- ◆ 自适应软件开发 (ASD) : 其核心是三个非线性的、重叠的开发阶段: 猜测、合作与学习。
- ◆ 动态系统开发方法 (DSDM) : 倡导以业务为核心。



## 软件开发模型 - 敏捷开发方法

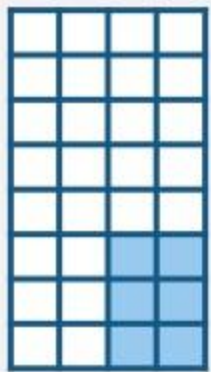


### 敏捷开发方法 - SCRUM

从相关方处获得需求



产品负责人 (PO)



产品待办列表



团队

团队选择该迭代  
承诺完成多少工作

迭代计划会议



迭代待办事项

下个迭代版本  
产品待办列表整理



敏捷教练  
(Scrum Master)



迭代  
(1-4周)

在一个迭代版本中  
目标没有发生变化



每日站会



迭代评审会议



潜在可发布的产品增量



迭代回顾会议



## 软件开发模型 – 敏捷开发方法



( ) 适用于程序开发人员在地域上分布很广的开发团队。( ) 中，编程开发人员分成首席程序员和“类”程序员。

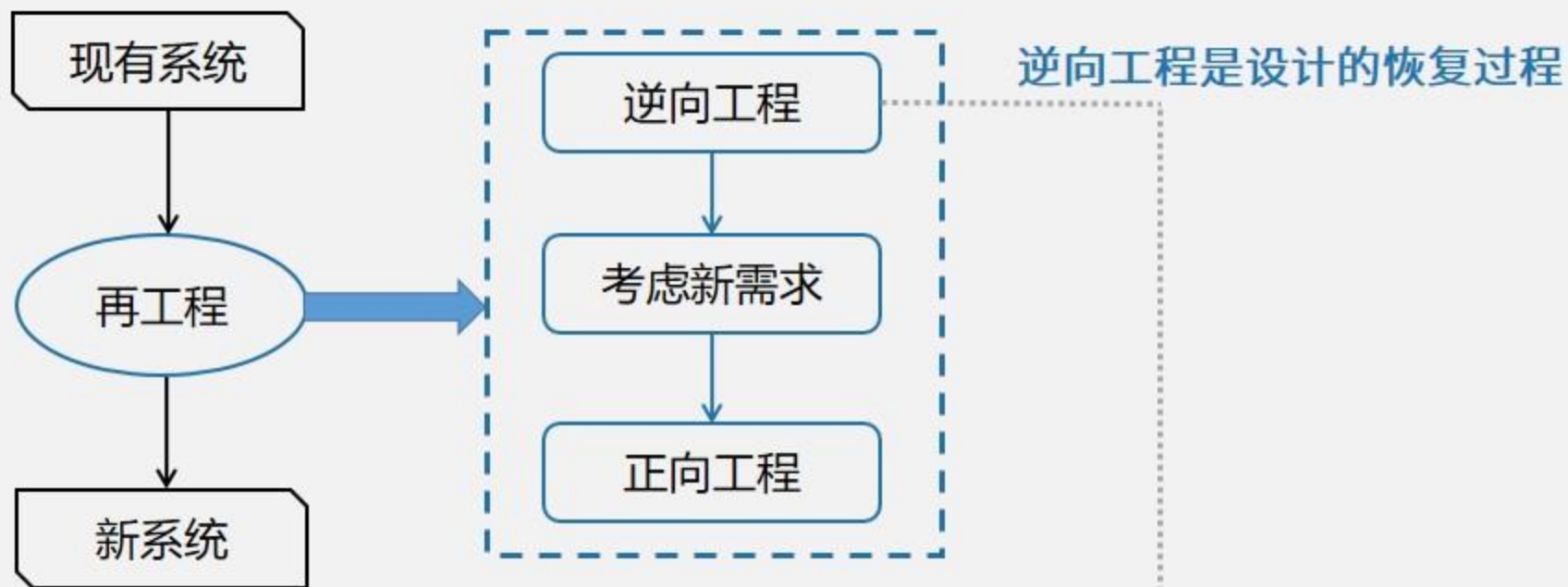
- A 水晶系列 (Crystal) 开发方法
- B 开放式源码 (Open source) 开发方法
- C SCRUM开发方法
- D 功用驱动开发方法 (FDD)

- A 自适应软件开发 (ASD)
- B 极限编程 (XP) 开发方法
- C 开放统一过程开发方法 (OpenUP)
- D 功用驱动开发方法 (FDD)





## 软件开发方法 – 逆向工程



实现级：包括程序的抽象语法树、符号表、过程的设计表示

结构级：包括反映程序分量之间相互依赖关系的信息，例如调用图、结构图、程序和数据结构

功能级：包括反映程序段功能及程序段之间关系的信息，例如数据和控制流模型

领域级：包括反映程序分量或程序诸实体与应用领域概念之间对应关系的信息，例如实体关系模型



## 软件开发方法 – 逆向工程



软件逆向工程就是分析已有的程序，寻求比源代码更高级的抽象表现形式。在逆向工程导出信息的四个抽象层次中，（ ）包括反映程序各部分之间相互依赖关系的信息；（ ）包括反映程序段功能及程序段之间关系的信息。

A 实现级

B 结构级

C 功能级

D 领域级

A 实现级

B 结构级

C 功能级

D 领域级