



# 系统测试与维护



## 课程内容提要

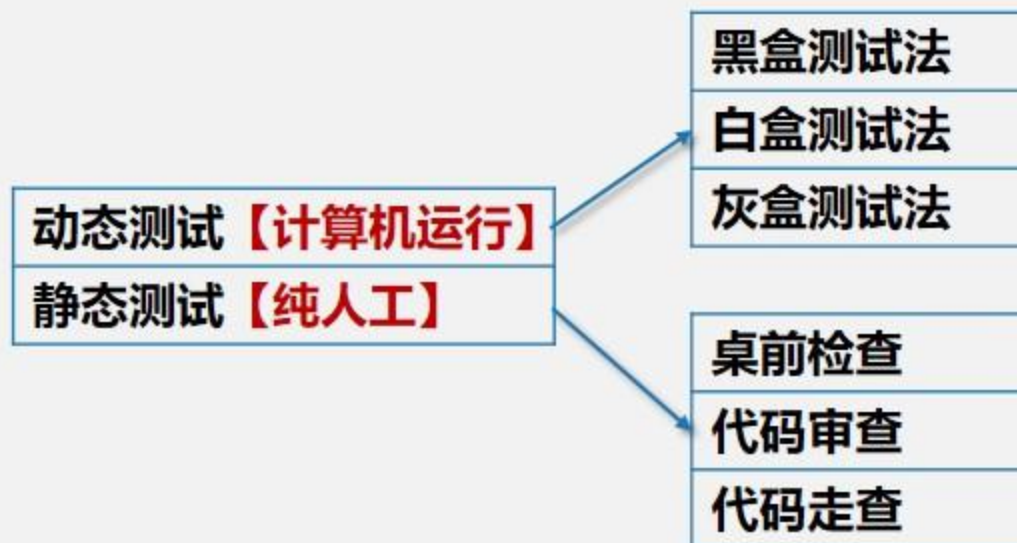


- 软件测试概念 (★★)
- 测试方法及阶段 (★★★★)
- 软件开发环境与工具 (★)
- 可维护性因素 (★★)
- 维护类型 (★★)



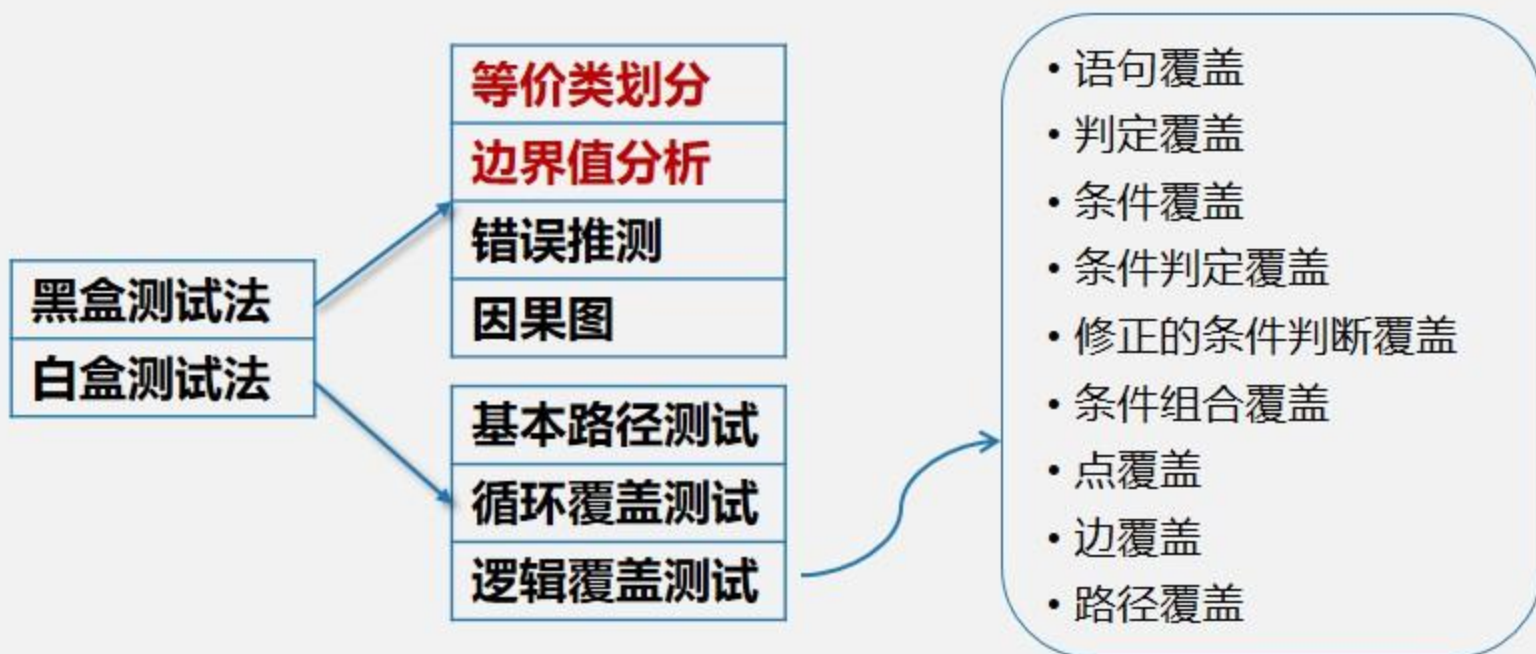
## 软件测试 – 测试类型

- ★ 尽早、不断的进行测试
- ★ 程序员避免测试自己设计的程序
- ★ 既要选择有效、合理的数据，也要选择无效、不合理的数据
- ★ 修改后应进行回归测试
- ★ 尚未发现的错误数量与该程序已发现错误数成正比





## 软件测试 – 测试用例设计







## 软件测试 – 测试类型



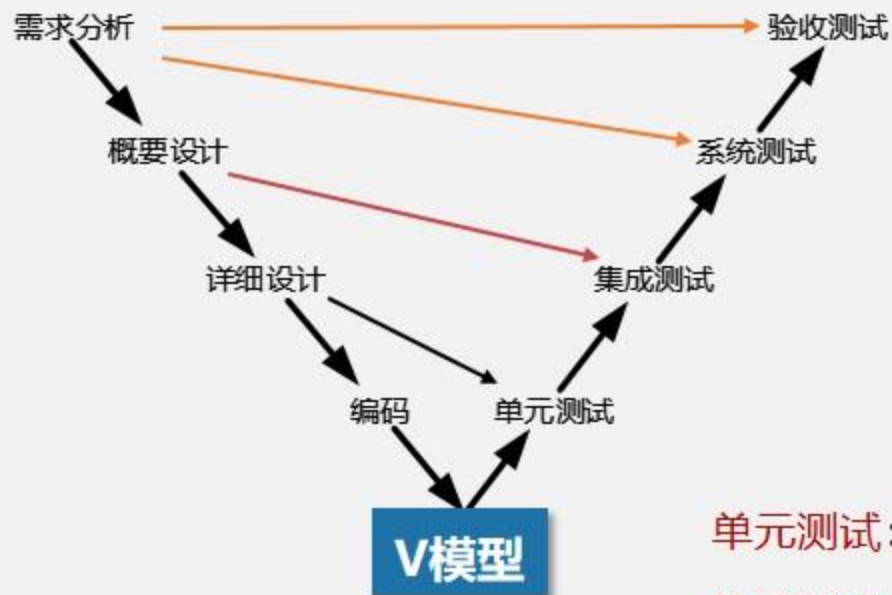
软件测试一般分为两个大类：动态测试和静态测试。前者通过运行程序发现错误，包括（ ）等方法；后者采用人工和计算机辅助静态分析的手段对程序进行检测，包括（ ）等方法。

- A 边界值分析、逻辑覆盖、基本路径
- B 桌面检查、逻辑覆盖、错误推测
- C 桌面检查、代码审查、代码走查
- D 错误推测、代码审查、基本路径

- A 边界值分析、逻辑覆盖、基本路径
- B 桌面检查、逻辑覆盖、错误推测
- C 桌面检查、代码审查、代码走查
- D 错误推测、代码审查、基本路径



## 软件测试 – 测试阶段



**单元测试：**模块测试，模块功能、性能、接口等

**集成测试：**模块间的接口

**系统测试：**真实环境下，验证完整的软件配置项能否和系统正确连接。

**确认测试：**验证软件与需求的一致性。内部确认测试、Alpha测试、Beta测试，验收测试

**回归测试：**测试软件**变更之后**，变更部分的正确性对变更需求的符合性。



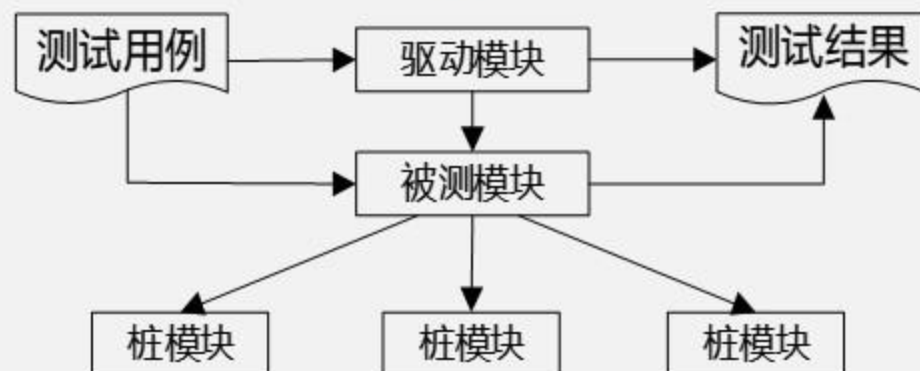
## 软件测试 – 测试阶段



### 集成测试策略

- 一次性组装【风险高】
- 增量式组装【测试全面】

- ✓ 自顶向下【需要桩模块】
- ✓ 自底向上【需要驱动模块】
- ✓ 混合式【都需要】





## 软件测试 – 测试阶段



### 系统测试

- 功能测试
- 性能测试
- 健壮性测试
- 用户界面测试
- 安全性测试
- 安装与反安装测试



- ✓ 负载测试：各种工作负载下系统的性能
- ✓ 压力测试：系统的瓶颈或不能接受的性能点
- ✓ 强度测试：系统资源特别低的情况下运行
- ✓ 容量测试【并发测试】：同时在线的最大用户数
- ✓ 可靠性测试：MTTF之类的参数





## 软件测试 – 面向对象的测试



- ★ 算法层（**单元测试**）：包括等价类划分测试、组合功能测试（基于判定表的测试）、递归函数测试和多态消息测试
- ★ 类层（**模块测试**）：包括不变式边界测试、模态类测试和非模态类测试
- ★ 模板层/类树层（**集成测试**）：包括多态服务测试和展平测试
- ★ 系统层（系统测试）



## 软件测试 – 测试阶段



在单元测试中，（ ）。

- A 驱动模块用来调用被测模块，自顶向下的单元测试中不需要另外编写驱动模块
- B 桩模块用来模拟被测模块所调用的子模块，自顶向下的单元测试中不需要另外编写桩模块
- C 驱动模块用来模拟被测模块所调用的子模块，自底向上的单元测试中不需要另外编写驱动模块
- D 桩模块用来调用被测模块，自底向上的单元测试中不需要另外编写桩模块



## 软件测试 – 测试阶段



软件性能测试有多种不同类型的测试方法，其中，（ ）用于测试在限定的系统下考查软件系统极限运行的情况，（ ）可用于测试系统同时处理的在线最大用户数量。

A 强度测试

B 负载测试

C 压力测试

D 容量测试

A 强度测试

B 负载测试

C 压力测试

D 容量测试



## 软件测试 – 测试阶段



用边界值分析法，假定 $10 \leq X \leq 30$ ，那么X在测试中应取的边界值是（ ）。

A X=11, X=29

B X=9, X=10, X=30, X=31

C X=10, X=30

D X=9, X=31



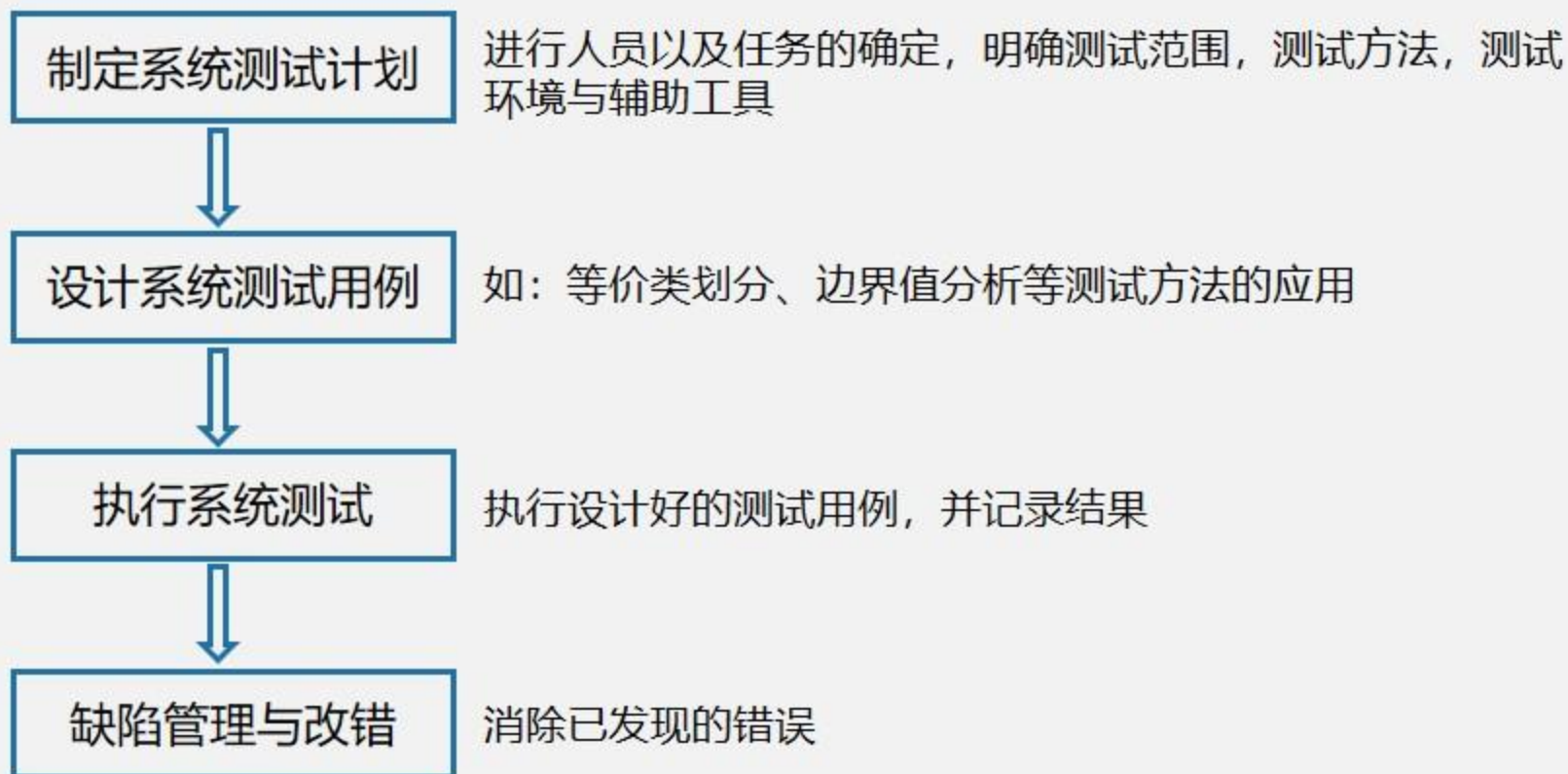




## 软件测试 – 系统测试步骤



### 系统测试活动与步骤





## 软件调试



### ★ 软件调试方法

- 蛮力法：主要思想是“通计算机找错”，低效，耗时
- 回溯法：从出错处人工沿控制流程往回追踪，直至发现出错的根源。复杂程序由于回溯路径多，难以实施
- 原因排除法：主要思想是演绎和归纳，用二分法实现



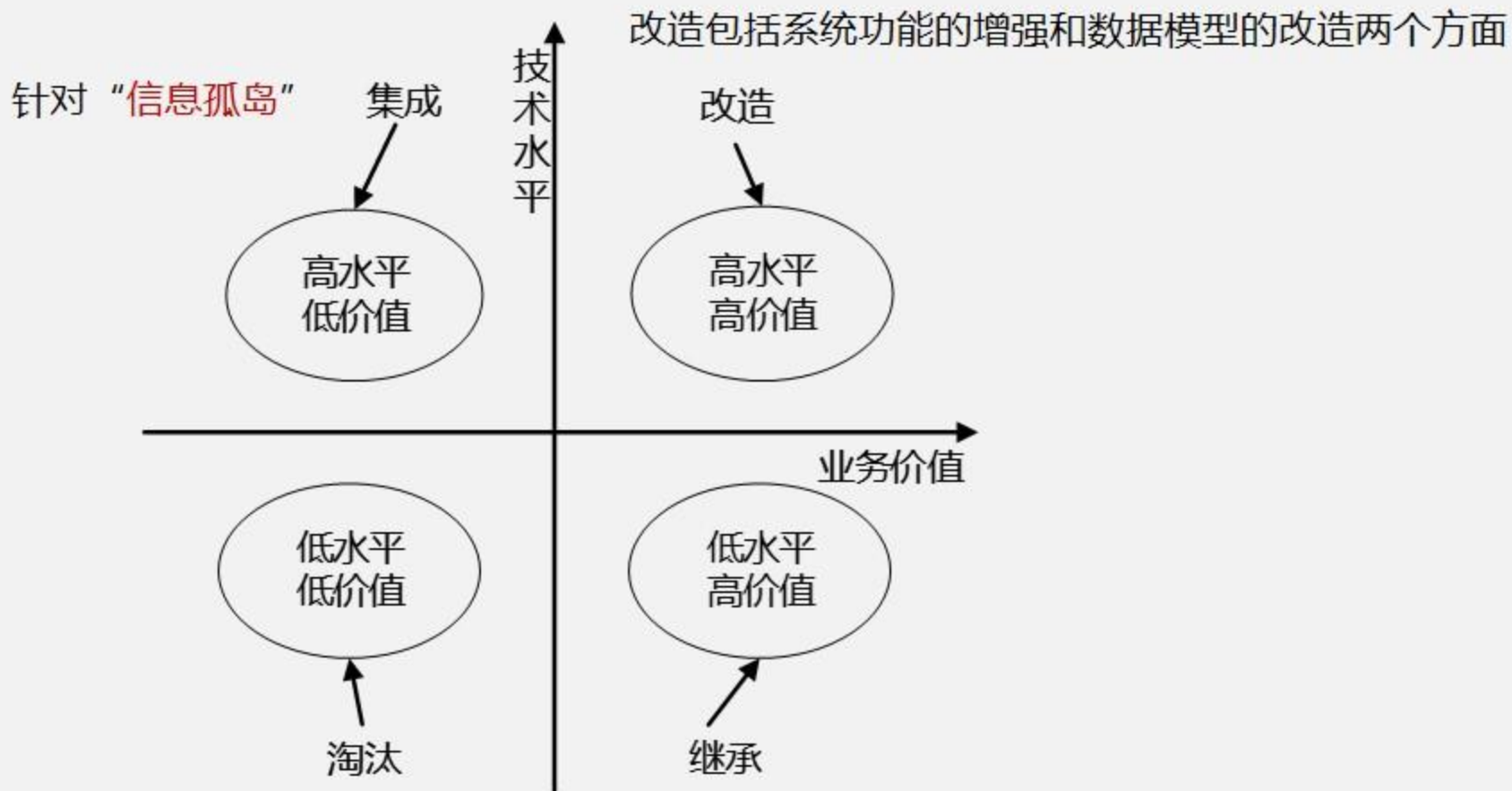
## 软件调试



软件测试	软件调试
目的是找出存在的错误	目的是定位错误并修改程序以修正错误
从一个已知的条件开始，使用预先定义的过程，有预知的结果	从一个未知的条件开始，结束的过程不可预计
测试过程可以事先设计，进度可以事先确定	调试不能描述过程或持续时间



## 系统转换计划 – 遗留系统演化策略



开发新系统时，需要完全兼容遗留系统的功能模型和数据模型





## 系统转换计划 – 遗留系统演化策略



遗留系统的演化可以采用淘汰、继承、改造和集成四种策略。若企业中的遗留系统技术含量较高，业务价值较低，在局部领域中工作良好，形成了一个信息孤岛时，适合于采用（ ）演化策略。

A 淘汰

B 继承

C 改造

D 集成



## 系统转换计划 – 新旧系统的转换策略



直接转换策略



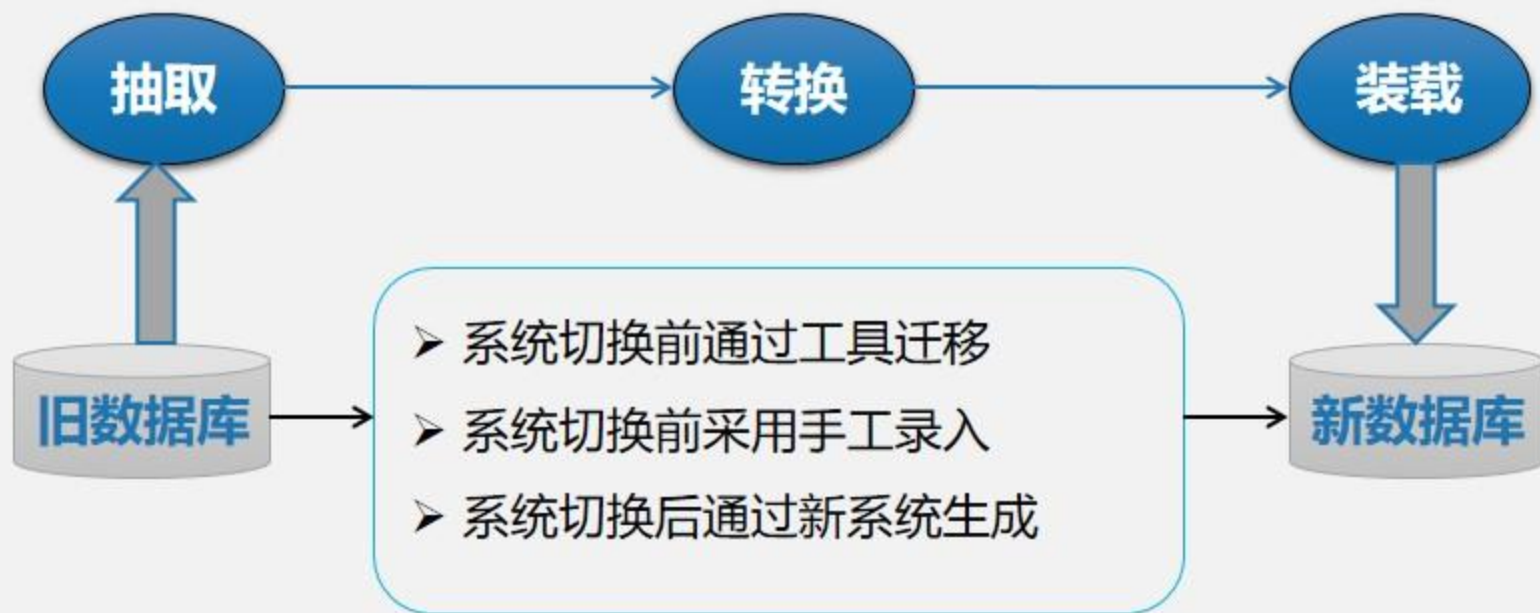
并行转换策略



分段转换策略



## 系统转换计划 - 数据转换与迁移





## 系统运行与维护



- ★ **正确性维护**：指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误。
- ★ **适应性维护**：指使应用软件适应环境变化【外部环境、数据环境】而进行的修改。
- ★ **完善性维护**：扩充功能和改善性能而进行的修改。
- ★ **预防性维护**：为了适应未来的软硬件环境的变化，应主动增加预防性的新的功能，以使系统适应各类变化而不被淘汰。如将专用报表功能改成通用报表生成功能，以适应将来报表格式的变化。





## 例题讲解



软件的维护并不只是修正错误。为了满足用户提出的增加新功能、修改现有功能以及一般性的改进要求和  
建议，需要进行（ ），它是软件维护工作的主要部分；软件测试不可能揭露旧系统中所有潜在的错误，  
所以这些程序在使用过程中还可能发生错误，诊断和更正这些错误的过程称为（ ）；为了改进软件未来  
的可维护性或可靠性，或者为了给未来的改进提供更好的基础而对软件进行修改，这类活动称为（ ）。

A 完善性维护      B 适应性维护      C 预防性维护      D 改正性维护

A 完善性维护      B 适应性维护      C 预防性维护      D 改正性维护

A 完善性维护      B 适应性维护      C 预防性维护      D 改正性维护