



第五章 软件测试

软件测试方法
软件测试类型
测试步骤
测试用例的设计
纠错技术

软件测试目标与测试过程模型

- 测试目标
 - 发现错误
 - 预防错误
 - 使用人工或自动的手段，运行或测定某个系统的过程。其目的是检验它是否满足规定的要求，或是清楚了解预期结果与实际结果之间的差异。
- 测试原则

■ 软件错误类型

- 结构错误
- 数据错误
- 编程错误
- 接口错误

■ 软件测试的方法

- 白盒测试
- 黑盒测试

■ 软件测试的生命周期

- 测试计划
 - 测试目的
 - 测试任务
 - 测试环境
 - 测试类型
 - 测试进度及人员安排
- 测试设计

- 测试开发
- 测试执行
- 测试评估分析
- 缺陷跟踪

■ 测试类型

- 数据和数据库完整性测试
- 功能测试
- 用户界面测试
- 性能测试
- 强度测试
- 容量测试
- 可靠性测试
- 配置测试
- 安装测试

■ 回归测试

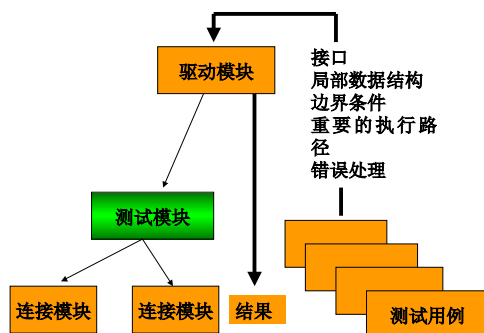
■ 其它测试

包括软件的安全性、兼容性、可用性、健壮性、容灾性、恢复性以及文档的完整性与精确性。

软件测试步骤

■ 单元测试

- 单元测试的内容
 - 模块接口
 - 局部数据结构
 - 重要的执行路径
 - 错误处理
 - 边界测试
- 单元测试的方法
 - 驱动程序
 - 连接程序



■ 组装测试

- 自顶向下结合
 - 先深度后宽度
 - 先宽度后深度
 - 特点
- 自底向上结合
- 折中的方法
 - 上层：自顶向下
 - 下层：自底向上

■ 确认测试

- 测试方法：黑盒测试
- 测试后的处理
- α 测试和 β 测试

■ 系统测试

- 恢复测试
- 安全测试
- 强度测试
- 性能测试

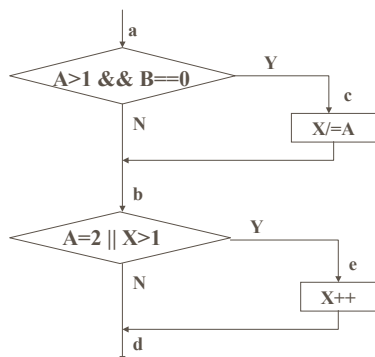
软件测试技术

■ 测试用例设计

■ 逻辑覆盖

例: void fun(float A,float B,float X)

```
{ if ( A>1 && B==0 ) Then X /=A;
  if ( A==2 || X>1 )      then X++;
}
```



■ 语句覆盖

测试用例: 路径: a-c-e

输入数据:	$\begin{cases} A=2 \\ B=0 \\ X=4 \end{cases}$	预期结果:	$\begin{cases} A=2 \\ B=0 \\ X=3 \end{cases}$
-------	---	-------	---

■ 分支覆盖

测试用例1: 路径: a-c-e

输入数据:	$\begin{cases} A=2 \\ B=0 \\ X=4 \end{cases}$	预期结果:	$\begin{cases} A=2 \\ B=0 \\ X=3 \end{cases}$
-------	---	-------	---

测试用例2: 路径: a-b-d

输入数据:	$\begin{cases} A=1 \\ B=0 \\ X=1 \end{cases}$	预期结果:	$\begin{cases} A=1 \\ B=0 \\ X=1 \end{cases}$
-------	---	-------	---

■ 条件覆盖

条件1: A > 1	——T1	} 第一个判定
(A ≤ 1)	——F1	
条件2: B = 0	——T2	
(B ≠ 0)	——F2	
条件3: A = 2	——T3	} 第二个判定
(A ≠ 2)	——F3	
条件4: X > 1	——T4	
(X ≤ 1)	——F4	

测试用例1: 路径: a-c-e

输入数据: $\begin{cases} A=2 \\ B=0 \\ X=1 \end{cases}$ 满足条件: T1, T2, T3, F4
满足判定为“真”的分支。

测试用例2: 路径: a-b-e

输入数据: $\begin{cases} A=1 \\ B=1 \\ X=2 \end{cases}$ 满足条件: F1, F2, F3, T4
未满足判定2为“假”的分支。

■ 分支/条件覆盖

■ 多重覆盖

■ 条件组合

- | | |
|------------------|------------------|
| (1) $A>1, B=0$ | (5) $A=2, X>1$ |
| (2) $A>1, B!=0$ | (6) $A=2, X<=1$ |
| (3) $A<=1, B=0$ | (7) $A!=2, X>1$ |
| (4) $A<=1, B!=0$ | (8) $A!=2, X<=1$ |

■ 测试用例

$A=2, B=0, X=4$	(1, 5)
$A=2, B=1, X=1$	(2, 6)
$A=1, B=0, X=2$	(3, 7)
$A=1, B=1, X=1$	(4, 8)

■ 循环覆盖

- 单循环
- 嵌套循环
- 串联循环

■ 等价类划分

■ 划分等价类

- 输入数据范围和个数
- 一组可能的取值
- 成立的条件

根据	有效	无效
取值范围和个数	1个	2个
例: 1-9999	$1 \leq X \leq 9999$	$X < 1$ $X > 9999$
输入值和操作	1个	1个
例: 菜单选择操作	F1键	F2键
数据类型	多个	多个
例: 整型	+整数、0、-整数	浮点、字符型等

■ 确定测试用例

- 为每个等价类规定一个唯一的编号，建立等价类表。
- 设计一个测试用例，使其尽可能多地覆盖尚未覆盖的有效等价类，直到所有等价类全部被覆盖。
- 设计新的测试用例，使其只覆盖一个无效等价类。

例：电话号码由三部分组成。

区号	+	前缀	+	后缀
□□□		2000		××××
×××			
		9999		
(3位数字/空)		(4位数字)		(4位数字)

输入条件	有效等价类	无效等价类
区号	空白 (1) 3位数字 (2)	非数字字符 (5) 少于3位数字 (6) 多于3位数字 (7)
前缀	2000~9999间的4位数字 (3)	非数字字符 (8) 小于2000 (9) 大于9999 (10) 少于4位数字 (11) 多于4位数字 (12)
后缀	4位数字 (4)	非数字字符 (13) 少于4位数字 (14) 多于4位数字 (15)

设计有效类测试用例。

测试用例	覆盖等价范围	结果
6739-1742	(1) (3) (4)	有效
010-6739-1742	(2) (3) (4)	有效

设计无效类测试用例。

测试用例	覆盖等价范围	结果
01A-6739-8888	(5)	无效
33-6666-1234	(6)	无效
1234-9999-1234	(7)	无效
.....

■ 边界值分析

■ 测试范围

- 边界内附近
- 边界上
- 边界外附近

测试值: { 0.1
0.0
-0.0001
9998.9
9999
9999.1

■ 数据个数

- 最小个数
- 最大个数
- 最小个数-1
- 最大个数+1。

如：输入文件可包括1~255个纪录。

测试值应选: { 0个
1个
255个
256个

■ 图形技术

■ 因果图

- 根据软件规格说明书，分析找出原因和结果，并给每个原因和结果赋予一个标识符。
- 分析软件规格说明书描述的语义，找出原因和结果、原因和原因之间的关系，画出因果图。
- 对于某些由于约束和限制条件不可能出现的组合情况，用记号标明约束和限制条件。
- 把因果图转换成判定表。
- 根据判定表的每一列，设计测试用例。

■ 程序图

■ 纠错技术

- 归纳法
- 演绎法
- 回溯

测试计划和测试分析报告