1. 软件测试概述

1. 缺陷出现的原因：

一、**产品说明书(需求) (主要原因)** 随意、易变、沟通不足

二、**设计 (次要原因)** 随意、易变、沟通不足

三、**编码** 软件复杂度，进度压力，低级错误

四、**其它** 理解错误，测试错误

2. 危机原因：缺乏规范化工程约束 ->缺陷的不断积累与放大效应

3. Glenford J. Myers:**测试是尽可能多地发现软件错误**

Myers的软件测试定义：测试是为发现错误而执行一个程序或系统的过程

Myers的**三个重要观点**：1. 测试是为了**证明程序有错**，而不是证明程序无错误；

2.一个**好的测试用例**是在于它能**发现至今未发现的错误**；

3.一个**成功的测试**是**发现了至今未发现的错误的测试**；

4. 测试与调试：**测试：发现错误、贯穿整个开发过程。人员一般不是开发人员。**

**调试：排除错误、解决问题、编码活动的一部分。人员一般是开发人员。**

**5. 软件测试的目的：1、确保软件质量**

**2、确保软件开发过程方向的正确性。**

6. 测试原理/原则（案例题）：**1、用户需求至上**

**2、测试时有计划的活动**

**3、缺陷有集群性**

**4、测试应从小规模走向大规模**

**5、穷尽测试不可能**

**6、有效的测试应由第三方独立进行**

**7、测试无法揭示所有缺陷**

**8、测试的杀虫剂悖论（潜在缺陷对已进行的测试具有免疫力）**

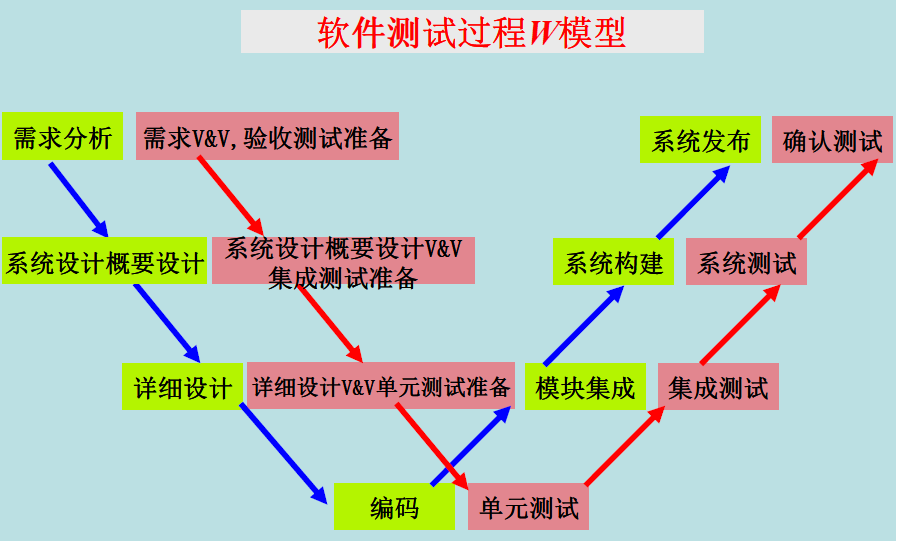
**9、测试是有风险的行为**

**10、并非所有软件缺陷都得修复。**

7. 测试用例三要素：**输入，执行条件，期望输出**

**8. 软件测试W模型：**在V模型的基础上，增加与开发阶段的同步测试，形成W模型；

**测试与开发同步进行**，有利于尽早的发现问题



1. **白盒测试**
2. **实施者**：单元测试阶段：一般由开发人员进行。

集成测试阶段：一般由测试人员和开发人员共同完成

**2. 静态白盒测试方法：1.桌面检查2.代码检查/走查3.代码评审** (规范化程度，弱🡪强)

**代码审查小组**：1.主持人2.作者3.评论员4.记录员

**代码审查步骤**：1**.计划**（主持人做计划）2.**概述**（描述技术背景）3.**准备**（评论员审查代码**核对表**）4.**审查会议**（阅读代码讨论提问，记录错误：类型、严重级别，讨论速度，**不讨论解决方案**）5.**审查报告**（缺陷列表：类型、严重级别）6.**返工**（分配缺陷并修复）7.**跟进**（监督，审查修复部分）

1. **动态白盒测试**
2. **基于控制流的测试方法**

**1.语句覆盖:保证程序中每条语句都执行一遍**，但可能会遗漏边。

一、100%的语句覆盖很困难

1)处理错误的代码片段2）小概率事件3)不可达代码（逻辑短路）

二、脆弱的100%语句覆盖

无法发现某些严重问题，原因：遗漏某些不存在语句的边

**2.判定覆盖（分支覆盖）：保证每个判断取真假值至少一次（判定的每个分支至少经过一次）**

**（判定覆盖包含语句覆盖）**

优点：简单，避免了语句覆盖的问题

缺点：忽略了表达式内的条件，不能发现每个条件的错误

**3.条件覆盖：保证每个判断中的每个条件去真假值至少一次**（很弱的覆盖准则）

局限：1.条件覆盖不稳定，无法知道对分支到底覆盖了多少

2.不能保证程序所有分支都被执行，即同时可能遗漏语句和分支

**4.判定条件覆盖：保证每个判断和每个条件取真假值至少一次**（较强）

错误屏蔽现象：100%覆盖，但不能检测出测试用例错误

**5.条件组合覆盖：保证每个判断中的所有条件的取值组合至少出现一次**

缺陷：1.可能会遗漏路径

2.代价昂贵：2n组合数目，n为一个判定中的原子条件数

3.某些条件组合式不可能的

条件组合测试用例的**约简**：利用**短路效应**寻找最小测试用例集，先做短路分析，再做测试用例

**6.路径覆盖：保证每条可能执行到的路径都至少经过一次（如果有环路，则每条环路至少经过一次）**

本质是判定的组合

优点：较为彻底的一种测试

局限：1)路径分支指数级增加2x,x为相互独立的分指数

2）不可达路径存在

3）不能替代涉及**条件测试**的方法

路径覆盖是一种比较强的覆盖标准，但**不能**替代条件覆盖、判定条件覆盖和条件组合覆盖

1. **基于数据流的测试方法**

步骤：1.流图

2.分析变量的定义节点和使用节点

3.可能的路径数=use\*def

4.列出DU路径

5.约简DU路径

6.设计测试用例

**三、黑盒测试**

**1. 黑盒vs白盒**：1.基于规格说明书；2.覆盖尽可能多的上层行为

**目的：测试 功能的正确和遗漏；界面错误；数据访问；性能；初始化和终止（PPT3-4 P53）**

**为什么需要黑盒测试？**1.无法获得源代码；2.尽早黑盒测试的益处——发现软件功能缺陷；3.弥补遗漏的逻辑缺陷；4.适用于各个测试阶段（单元测试、集成测试、系统测试、回归测试）

**2. 实施者： 专门的软件测试部门；有经验的测试人员。**

**3. 进入退出条件：进入条件：**

**编码开始：设计测试数据并执行测试。**

**退出条件：**

**1、完成测试计划**

**2、发现并修正了错误**

**3、预算和开发时间**

**4. 边界值分析vs等价类划分**

1.等价类划分是，往往要先确定边界值

2.边界值分析是等价类划分方法的补充

3.测试中需要将两者结合起来使用

**5.随机测试：**特点： 1、均匀选取可能的输入值

2、避免选择偏见

3、对所有输入一视同仁

缺点：

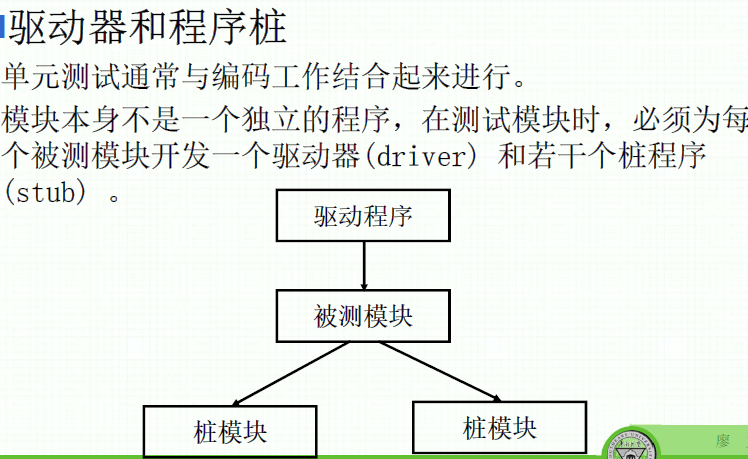
程序缺陷分配并不均匀

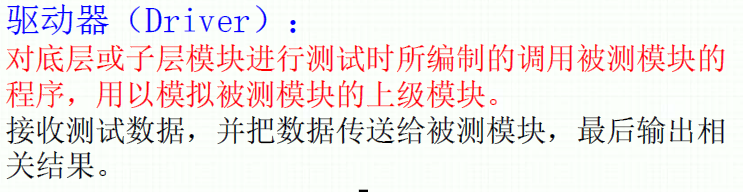
随机测试很难取到特定值

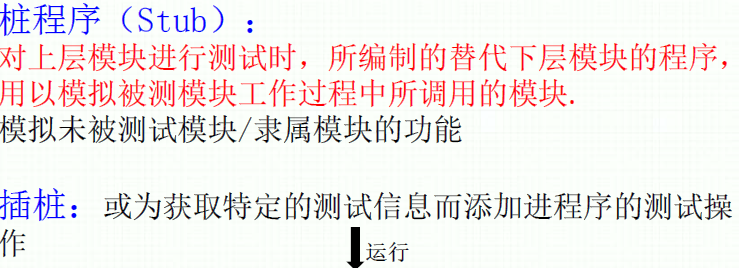
**6.蜕变测试：**对测试期望输出难以预测的程序，基于已知的期望输出以及相关领域知识和实现建立的用例之间、输出之间的关系来通过已知用例生成新的用例。通过多次生成的用例集来对程序进行测试就是蜕变测试。

1. **单元测试，集成测试**

**1．单元测试规程**

****

****

****

#### **2.** 瞬时集成测试：

**当所有构建都通过单元测试，就把他们组合成一个最终系统，并观察它是否正常运转**

**缺陷：1、无休止的错误、错误很多、修复困难**

**2、模块一次性结合，难以找出错误原因**

**3、接口错误和其他错误容易混淆**

**适用领域：小型软件开发**

#### **增量集成测试**

**将程序分成小的部分进行构造和测试**

**优点：1、错误易分离和修正**

**2、接口容易进行彻底测试**

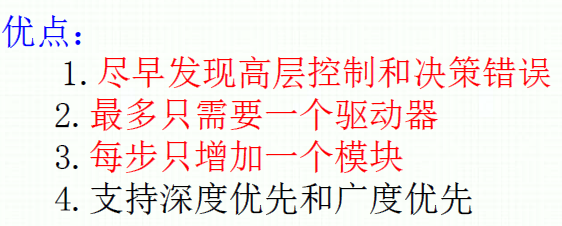
**缺点：会有额外开销、但能大大减少发现和修正错误的时间**

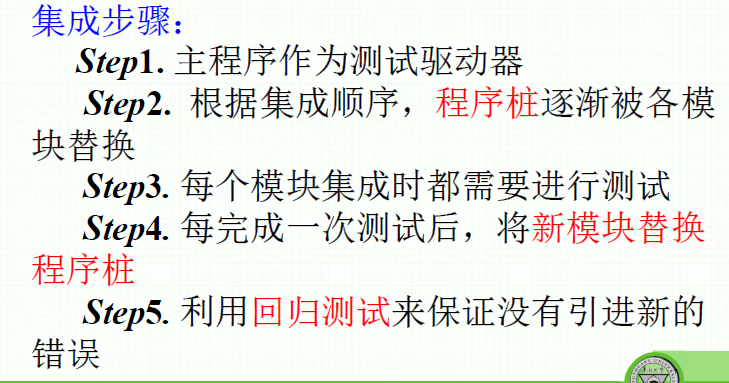
#### **自顶向下：**

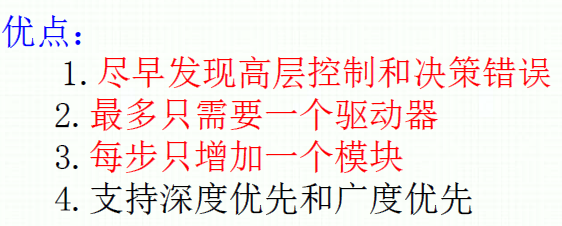
**首先集成主模块，然后按照控制层次向下集成**

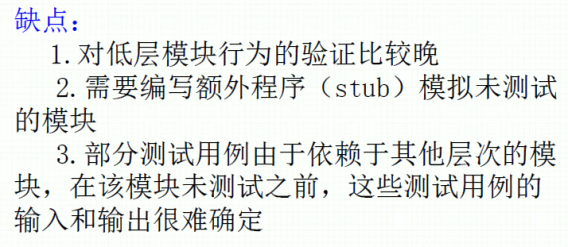
**方式：**

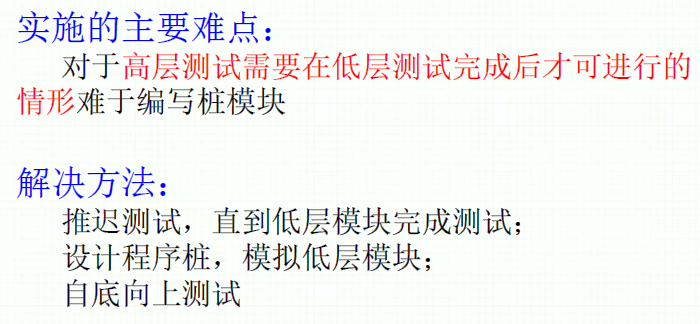
**深度优先、广度优先**

****

****

****

****

****

#### 自底向上集成

从原子模块，即程序最底层模块开始就构造并进行集成测试

