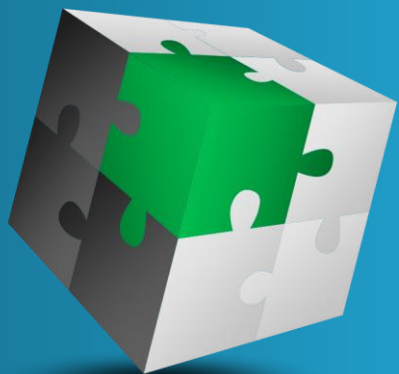


软件工程

Software Engineering



第六章 软件测试与质量管理

穆海伦

计算机学院 智能与软件研究所

E-mail: helen_se@163.com QQ: 1055874556 PH: 13750802617(612617)



软件工程

1

软件测试的基本概念

2

测试用例设计方法

3

软件质量管理



软件工程

1

软件测试的基本概念

2

测试用例设计方法

3

软件质量管理



1.1 软件测试的概念

是为了发现错误而执行程序的过程，即根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例，并利用这些测试用例去运行程序，以发现程序错误的过程。





根据G.J. Myers观点--软件测试的目的：

- ❑ 软件测试是为了发现错误而执行程序的过程
- ❑ 一个好的测试能够在第一时间发现程序中存在的错误
- ❑ 一个好的测试是发现了至今尚未发现的错误的测试。

软件测试是质量控制的重要手段，保证客户拿到或用户使用高质量的软件产品。





1.2 软件测试的误区和原则

软件测试的误区：

- ❑ 误区一：如果发布出去的软件有质量问题，都是软件测试人员的错
- ❑ 误区二：软件测试技术要求不高，至少比编程容易多了
- ❑ 误区三：有时间就多测试一些，来不及就少测试一些
- ❑ 误区四：软件测试是测试人员的事，与开发人员无关
- ❑ 误区五：根据软件开发瀑布模型，软件测试是开发后期的一个阶段





软件测试的原则：

- ❑ 所有测试的标准都是建立在用户需求之上。
- ❑ 软件测试必须基于“质量第一”的思想去开展各项工作，当时间和质量冲突时，时间要服从质量。
- ❑ 事先定义好产品的质量标准，只有有了质量标准，才能根据测试的结果，对产品的质量进行分析和评估。
- ❑ 软件项目一启动，软件测试也就是开始，而不是等程序写完，才开始进行测试。
- ❑ 穷举测试是不可能的。甚至一个大小适度的程序，其路径排列的数量也非常大，因此，在测试中不可能运行路径的每一种组合。
- ❑ 第三方进行测试会更客观，更有效。
- ❑ 软件测试计划是做好软件测试工作的前提。
- ❑ 测试用例是设计出来的，不是写出来的，所以要根据测试的目的，采用相应的方法去设计测试用例，从而提高测试的效率，更多地发现错误，提高程序的可靠性。





软件测试的原则：

- ❑ 对发现错误较多的程序段，应进行更深入的测试。一般来说，一段程序中已发现的错误数越多，其中存在的错误概率也就越大。
- ❑ 重视文档，妥善保存一切测试过程文档（测试计划、测试用例、测试报告等）
- ❑ 应当把“尽早和不断地测试”作为测试人员的座右铭
- ❑ 回归测试的关联性一定要引起充分的注意，修改一个错误而引起更多错误出现的现象并不少见
- ❑ 测试应从“小规模”开始，逐步转向“大规模”。
- ❑ 不可将测试用例置之度外，排除随意性。
- ❑ 必须彻底检查每一个测试结果。
- ❑ 一定要注意测试中的错误集中发生现象，这和程序员的编程水平和习惯有很大的关系
- ❑ 对测试错误结果一定要有一个确认的过程。



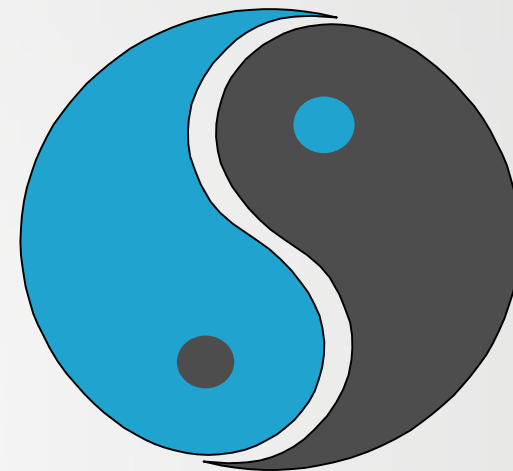


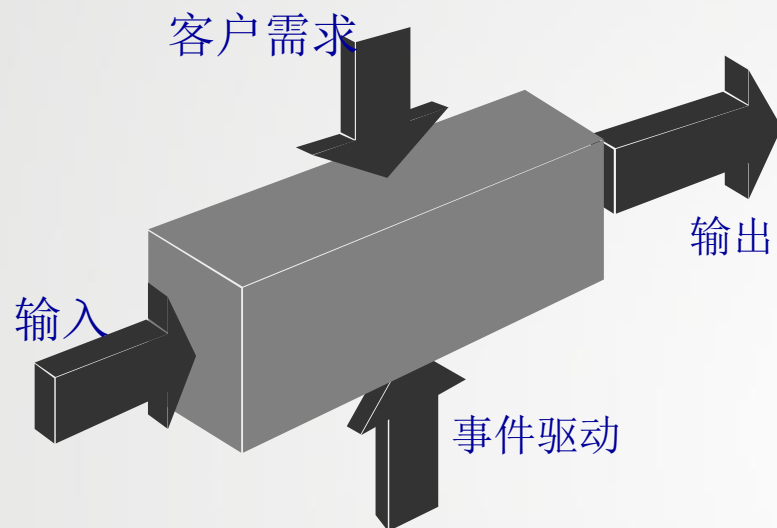
1.3 软件测试的方法

9

软件测试的方法：

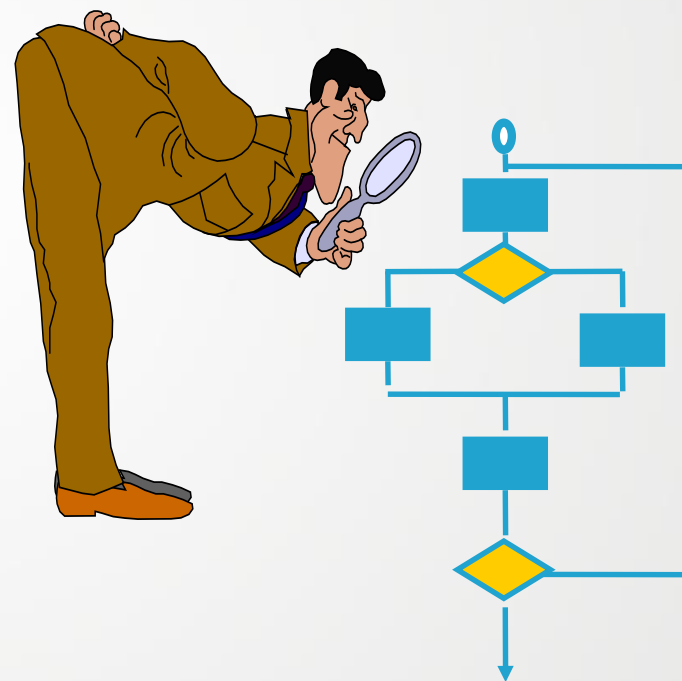
- ☐ 黑盒子和白盒子
- ☐ 静态的和动态的
- ☐ 文档、代码审查
- ☐ 数据输入边界条件法
- ☐ 等价划分、数据流程图
- ☐ 状态变换图
- ☐ 逻辑路径法

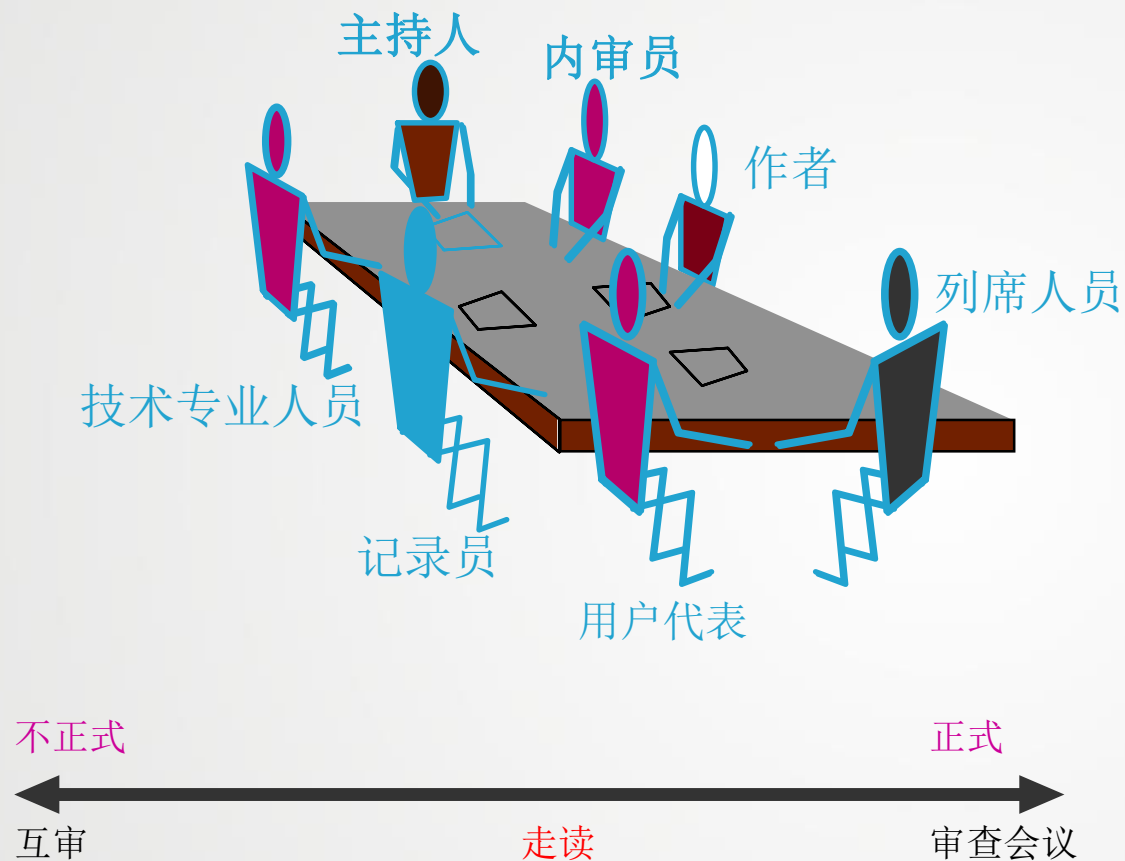




功能测试
数据驱动测试

结构测试
逻辑驱动测试





运行程序





手工模拟用户
操作





Verification: Are we building the product right?

- ❑ 是否正确地构造了软件？即是否正确地做事，验证开发过程是否遵守已定义好的内容。验证产品满足规格设计说明书的一致性。

Validation: Are we building the right product?

- ❑ 是否构造了正是用户所需要的软件？即是否正在做正确的事。验证产品所实现的功能是否满足用户的需求。





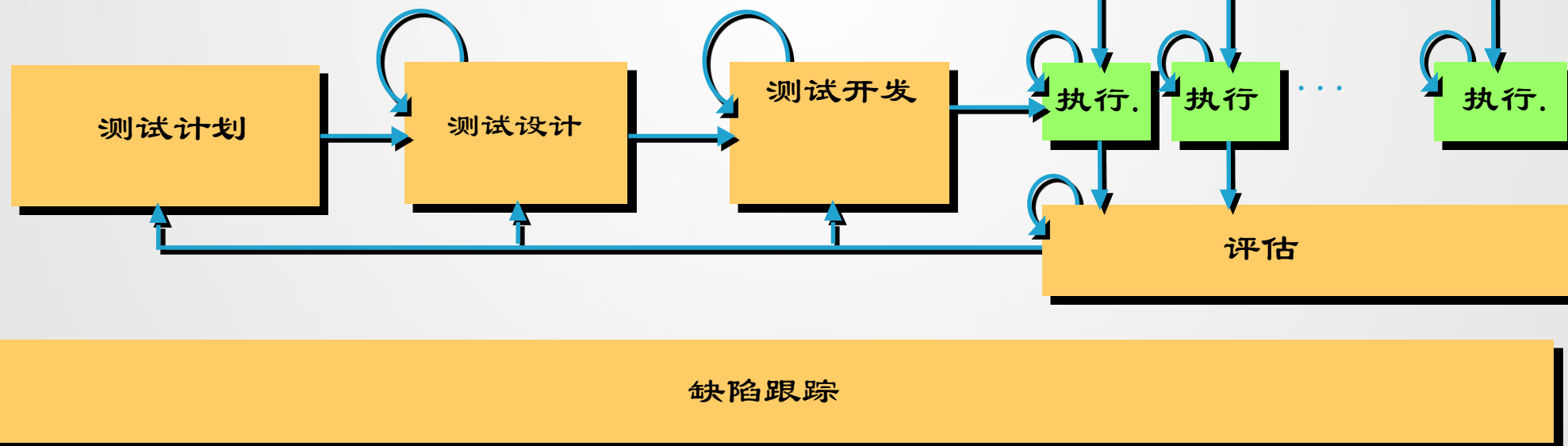
1.4 软件测试的分类和阶段

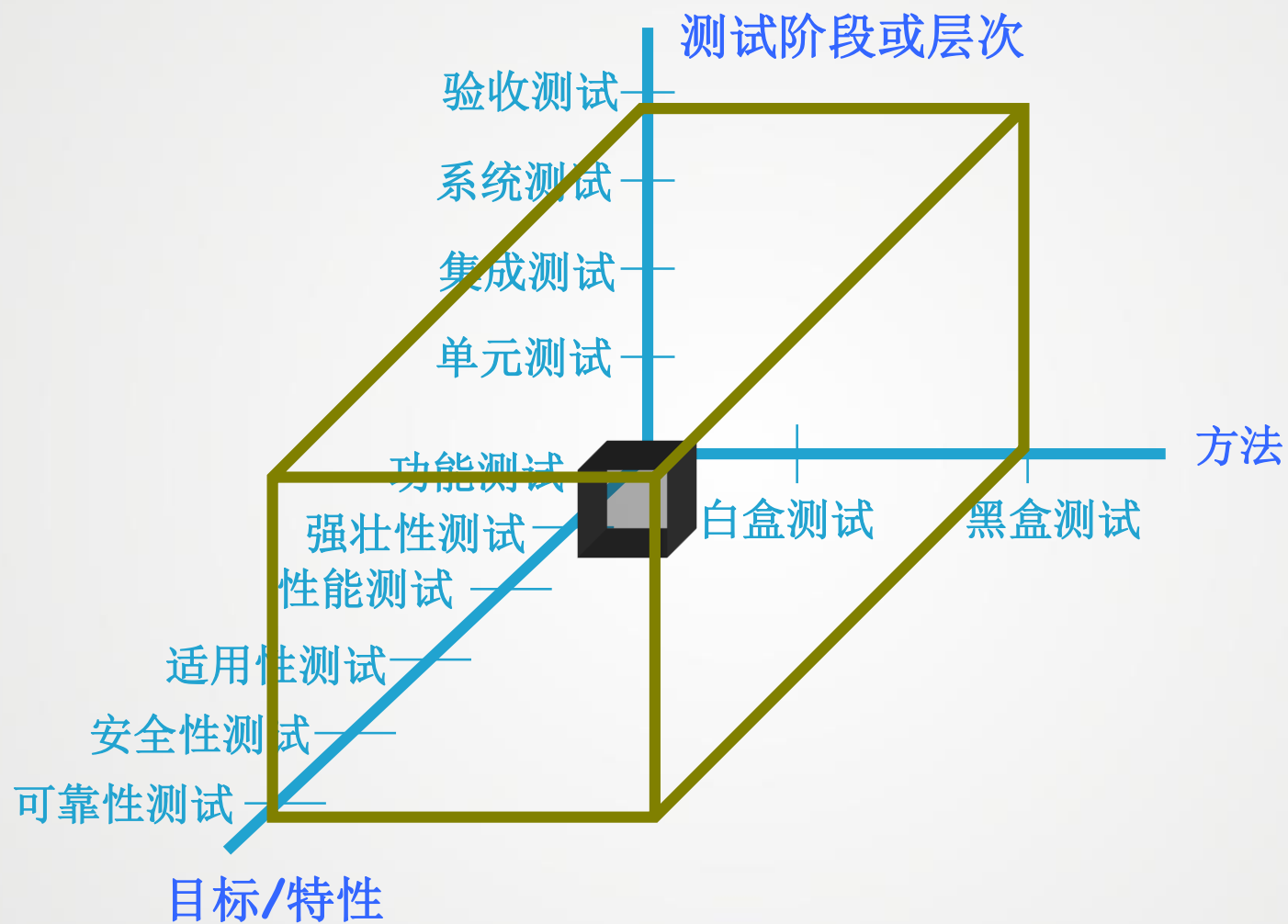
14

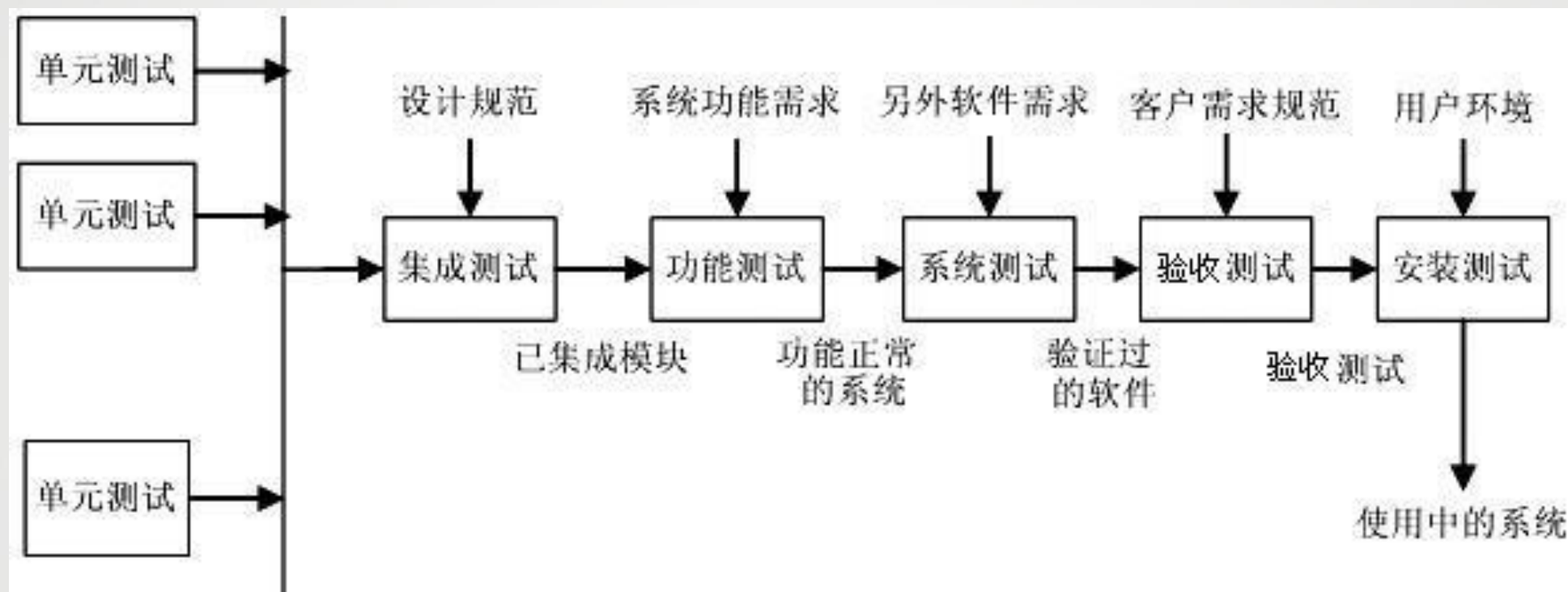
开发生命周期



测试生命周期









1) 单元测试

17

单元测试的对象是程序系统中的最小单元---模块或组件上，在编码阶段进行，针对每个模块进行测试，主要通过白盒测试方法，从程序的内部结构出发设计测试用例，检查程序模块或组件的已实现的功能与定义的功能是否一致、以及编码中是否存在错误。多个模块可以平行地、对立地测试，通常要编写驱动模块和桩模块。

单元测试一般由编程人员和测试人员共同完成。

单元测试的重点：

- ☐ 模块的接口
- ☐ 局部数据结构
- ☐ 重要的执行路径
- ☐ 出错处理路径
- ☐ 影响以上多项的边界条件





2) 集成测试

集成测试，也称组装测试、联合测试、子系统测试，在单元测试的基础上，将模块按照设计要求组装起来同时进行测试，主要目标是发现与接口有关的模块之间问题。

两种集成方式：一次性集成方式和增殖式集成方式。

集成测试的重点：

- ☐ 不正确的接口。
- ☐ 因存取全局（公用）数据引起的块间干扰。
- ☐ 不一致的文件与数据结构。
- ☐ 不适合的模块调用顺序。
- ☐ 出错处理上的错误。





集成测试常采用黑盒测试技术，其实施策略分为非渐增式和渐增式两种：

- ❑ 非渐增式测试：一次就把通过了单元测试的所有模块组装起来，进行全程序的测试，出了问题很难进行错误定位。
- ❑ 自顶向下测试：（渐增式），测试时从顶层模块开始，沿被测程序的结构图逐步下移，每次只增加一个新的模块。

特点：

- 1)能较早的显示出程序的轮廓
- 2)由顶向下的组装顺序，保证任何模块加进程前，其上级模块已先它装入，所以模块的驱动可以利用真实模块，只须编写桩模块供测试之用。
- 3)上层模块得到更多的测试机会，使被测程序获得更为彻底的检验。

- ❑ 自底向上测试：（渐增式），模块组装顺序采取由下向上的路线。

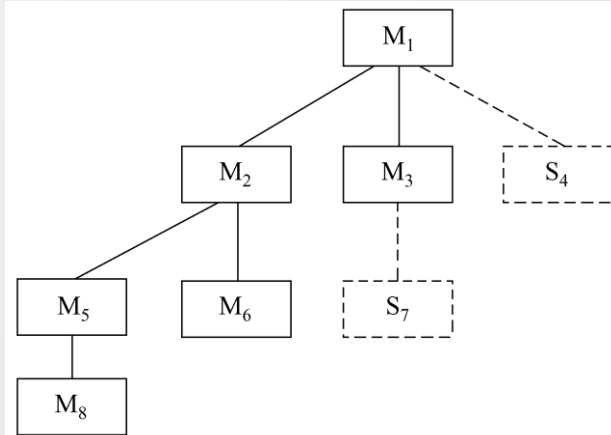
测试步骤：

- 1)从程序的较低层中找一个叶模块，由下向上地逐步增加新模块，组成程序的一个子程序或具有某一功能的模块“群”。
- 2)从另一子系统或群中选择另一个模块，仿照1)组成又一个子系统。
- 3)重复第2)步，得出所有子系统，然后组装成程序。

特点：

- 1)不能在测试的早期显示出程序的轮廓。
- 2)测试软件只需要驱动模块，不需要桩模块。

- ❑ 混合测试：是自顶向下与自底向下测试方法的结合。





3) 功能测试

功能测试一般须在完成集成测试后进行，而且是针对应用系统进行测试。功能测试是基于产品功能说明书，是在已知产品所应具有的功能，从用户角度来进行功能验证，以确认每个功能是否都能正常使用。



系统测试是将软件放在整个计算机环境下，包括软硬件平台、某些支持软件、数据和人员等，在实际运行环境下进行一系列的测试，包括恢复测试、安全测试、强度测试和性能测试等。





5) 验收测试 & 安装测试

验收测试的目的是向未来的用户表明系统能够像预定要求那样工作，验证软件的功能和性能如同用户所合理期待的那样。

安装测试是指按照软件产品安装手册或相应的文档，在一个和用户使用该产品完全一样的环境中或相当于用户使用环境中，进行一步一步的安装操作性的测试。





软件工程

1

软件测试的基本概念

2

测试用例设计方法

3

软件质量管理



2.1 黑盒测试方法

黑盒测试方法：

- ☐ 等价分类法。
- ☐ 边界值分析法。
- ☐ 错误推测法。
- ☐ 因果图法。







白盒测试方法：

□ 逻辑覆盖法：

- 语句覆盖：使被测试程序的每条语句至少执行一次。
- 判定覆盖：使被测试程序的每一分支都至少执行一次。
- 条件覆盖：要求判定中的每个条件都按“真”“假”两种结果至少执行一次。
- 判定/条件覆盖：要求判定中的每个条件都取到各种可能的值，而且每个判定表达式也都要取到各种可能的结果。
- 条件组合覆盖：要求判定中每个条件的各种可能组合都至少出现一次。

□ 路径测试法：

- 结点覆盖：程序的测试路径至少经过程序图中的每个结点一次。
- 边覆盖：程序的测试路径至少经过程序图中每条边一次。
- 路径覆盖：要求程序图中每条路径都至少经过一次。







2.3 黑盒和白盒测试方法的区别

黑盒测试着重测试软件功能。黑盒测试并不能取代白盒测试，它是与白盒测试互补的测试方法，它很可能发现白盒测试不易发现的其他类型的错误。

黑盒测试力图发现下述类型的错误：①功能不正确或遗漏了功能；②界面错误；③数据结构错误或外部数据库访问错误；④性能错误；⑤初始化和终止错误。

白盒测试在测试过程的早期阶段进行，而黑盒测试主要用于测试过程的后期。

设计黑盒测试方案时，应该考虑下述问题：

- ☐ 怎样测试功能的有效性？
- ☐ 哪些类型的输入可构成好测试用例？
- ☐ 系统是否对特定的输入值特别敏感？
- ☐ 怎样划定数据类的边界？
- ☐ 系统能够承受什么样的数据率和数据量？
- ☐ 数据的特定组合将对系统运行产生什么影响？

应用黑盒测试技术，能够设计出满足下述标准的测试用例集：

- ☐ 所设计出的测试用例能够减少为达到合理测试所需要设计的测试用例的总数；
- ☐ 所设计出的测试用例能够告诉我们，是否存在某些类型的错误，而不是仅仅指出与特定测试相关的错误是否存在。





2.4 课堂练习：软件测试方法

29

设有一程序，如图所示，能够对“姓名”、“年龄”、“性别”信息进行输入，且根据下面的规则判断有效性，并将判断的结果进行显示：

- ❑ “姓名”：2-8位字母；
- ❑ “年龄”：0-150的整数；
- ❑ “性别”：数字0或1；

如果输入的信息有效，则显示“信息输入有效”；

如果输入的信息无效，则判断无效的原因，并进行显示，如：“年龄信息输入无效”。

试用等价分类法为该程序设计相应的测试用例。

信息输入程序

姓名：

年龄：

性别：

信息输入判定结果：

年龄信息输入无效

确定





- 测试用例
- 测试分析报告



本章结束

***ANY
QUESTION***

