

CHAPTER 7

Software Testing

Outline

- basic concept (why, goal)**
- two testing methods**
- testing steps**
- reliability analysis /stop testing conditions**
- debug**

Software Testing



Why software testing ?

➤ 简单举例

- 某飞行器控制软件在飞行高度达到500米时，开始对航偏和高度进行修正。

```
if (height == 500)
{
    .... //修正航偏和高度
}
```

单点相等判断结合具体运行环境，导致执行分支不正确。

Why software testing ?

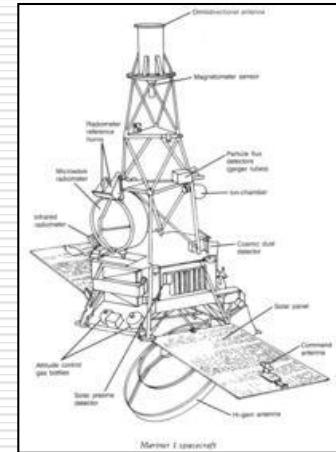
➤ 简单举例

```
int i;  
  
for (i=0; i<=200000; i++)  
  
{  
  
    .....  
  
}
```

变量类型使用不当，导致死循环。

低级错误，重大事故

1962年，美国金星探测器水手1号的运载火箭中，控制飞行姿态的软件发生故障，导致偏离轨道，地面人员不得以发出自毁指令，导致美国首次太阳系飞行计划失败。



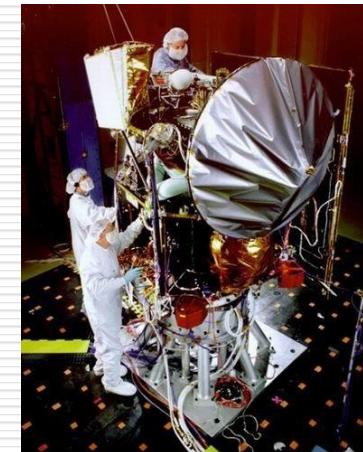
稿纸上的 $\bar{x} \rightarrow$ 程序中的 x

速度平均值 \rightarrow 原始速度



低级错误，重大事故

1998年，为火星着陆计划选择合适地点的美国火星轨道探测器MCO没有进入预定轨道，陷入火星大气层坠毁。



英制单位“磅” → 公制单位“牛顿”



洛克希德·马丁公司
提供的数据单位



NASA喷气动力实验室
编写的导航软件采用
的数据单位

许多重大软件事故

- 1994–1995, 迪斯尼的狮子王系统不支持问题。
- 1996年6月4日，欧洲阿丽亚娜5型火箭在发射40秒后爆炸，原因是惯性参考系统软件的数据转换异常造成软件失效。
- Intel的pentium处理器
1994年浮点除法缺陷。
200年8月28日，1.13 MHZ处理器一个可能导致运行程序被挂起的执行指令问题。
- 1999年12月3日, 美国航天局火星极地登陆飞船失踪。
- 1991年爱国者导弹防御系统系统时钟错误积累，造成跟踪系统失去精确度。
- 千年虫, 世界各地解决2000年错误超过数亿美元。

许多重大软件事故

- 2003年8月14日，分布式计算机系统试图同时访问同一资源引起软件失效，造成美国东北部大面积停电。
- 2004年，洛杉矶机场400多架飞机与机场失去联系，原因是空管软件时钟缺陷。
- 2006年3次故障，中航信离港系统，近百机场登机系统瘫痪
 - 2006年1月11日
 - 2006年7月26日
 - 2006年10月9日
-

从计算机出现至今，软件质量一直是人们关注的重大问题，软件问题在国民经济、生命安全、国防等方面导致的灾难和事故屡见不鲜。

思考

问题1：为什么软件会导致这么多灾难性后果？

问题2：为什么不能在软件交付使用前把所有的错误消除？

问题3：采用什么方法能够在软件交付使用前尽可能多地发现错误？

软件测试是目前工程中使用最广泛的一种方法

测试在软件开发中非常重视

- **The biggest cost in software development**
- 软件产品最大的成本是检测软件错误、修正软件错误的成本。
- 在整个软件开发中，测试工作量一般占 $30\% \sim 40\%$ ，甚至 $\geq 50\%$ 。
- 在人命关天的软件（如飞机控制、核反应堆等）测试所花费的时间往往是其它软件工程活动时间之和的三到五倍。

举例

“很多人都认为微软是一家软件开发公司，而事实上，我们是一家软件**测试**公司”

——比尔·盖茨

例：Windows95有1000万行代码，Windows2000有5000万行代码，3000多个工程师，几百个小团队。Exchange2000和Windows2000开发人员结构。

	Exchange2000	Windows2000
项目经理	25人	约250人
开发人员	140人	约1700人
测试人员	350人	约3200人

举例

国内软件行业 测试/开发人员比例调查情况表

	1:1	1:2	1:3	1:4	1:5	1:6	1:7	1:7以上
2009年	9%	14%	20%	14%	13%	5%	5%	20%
2008年	11%	14%	20%	11%	12%	5%	5%	22%

参考：国外软件行业平均测试/开发比例1:1

Why software testing ?—其他原因

➤ 难以说得清楚的软件问题

- ✓ 缺点(defect)
- ✓ 异常(anomaly)
- ✓ 谬误(fault)
- ✓ 偏差(variance)
- ✓ 问题(problem)
- ✓ 失败(failure)
- ✓ 错误(error)
- ✓ 缺陷(bug)

Why software testing ?

“如果软件中的问题没有人发现，那么它算不算软件缺陷？”

古谚：“一片树叶飘落在森林中，没有人听见，谁能说它发出了声音？”

由于不能报告没有看见的问题，因此，没有看见就不能说存在软件缺陷？

只有看到了，才能断言软件缺陷，尚未发现的软件缺陷只能说是未知软件缺陷。

眼
见
为
实

实践，testing !

Why software testing ?

➤ Software quality assurance



Software testing 定义

➤ **Definition:**

“Testing is the process of executing a program with the intent of finding errors.”

Glen Myers

It is about

- ✓ **Finding errors**
- ✓ **Exercising code across range of inputs and conditions**

软件测试的定义

- 软件测试是为了发现错误而运行一个软件的过程。
- 什么是软件错误？
 - 功能不正确
 - 性能不能满足要求
 - 界面不舒适
 -

Test narrow view (unit level)

- ✓ **Testing** is the process of finding fault in program.
- ✓ **A good test case** is one that has a high probability of finding an as-yet-undiscovered error.
- ✓ **A successful test** is one that uncovers an as-yet-undiscovered error.
- ✓ “**No errors found**”: not a good test!
- ✓ **Testing can’t prove program to be right.**

软件测试的定义或目标(狭义)

- ✓ 测试是为了发现程序中的错误而执行程序的过程
- ✓ 一个好的测试用例是发现了至今未发现错误的用例
- ✓ 一次成功的测试是发现了至今未发现错误的测试
- ✓ 程序测试能证明错误的存在，但不能证明错误不存在。
- ✓ 测试的目的是发现程序中的错误，为了证明程序有错，而不是证明程序无错。

其它说法

- ✓ 把证明程序无错当作测试目的不仅是不正确的，完全做不到的，而且对做好测试没有任何益处，甚至是十分有害的。
- ✓ 软件测试要设法使软件发生故障，暴露软件错误。
- ✓ 测试的“成功”与“失败”
- ✓ 能够发现错误的测试是成功的测试，否则是失败的测试。

其它说法

“测试的目的是说明程序正确地执行它应有的功能”
这种说法正确吗？

举例：程序Triangle，输入三个整数，表示一个三角形的三个边长，该程序产生一个结果，指出该三角形是等边三角形、等腰三角形、还是不等边三角形。

为说明其能正确执行它的功能，可使用“测试用例”(3, 4, 5), (5, 5, 6), (6, 6, 6)，程序都能给出正确结果，是否就可认为程序是正确的？

Broad view (广义)(acceptance level)

- ✓ The process used to ensure that the software conforms to its specification and meets the user requirements
- ✓ **Validation:** “Are we building the right product?”
- ✓ **Verification:** “Are we building the product right?”
- ✓ Takes place at all stages of Software Engineering



Verification vs. validation

Validation: (有效)

- "Are we building the right product"
- The software should do what the users really require
- Objective

Verification: (验证)

- "Are we building the product right"
- The software should conform to its specification
- Subjective

V & V must be

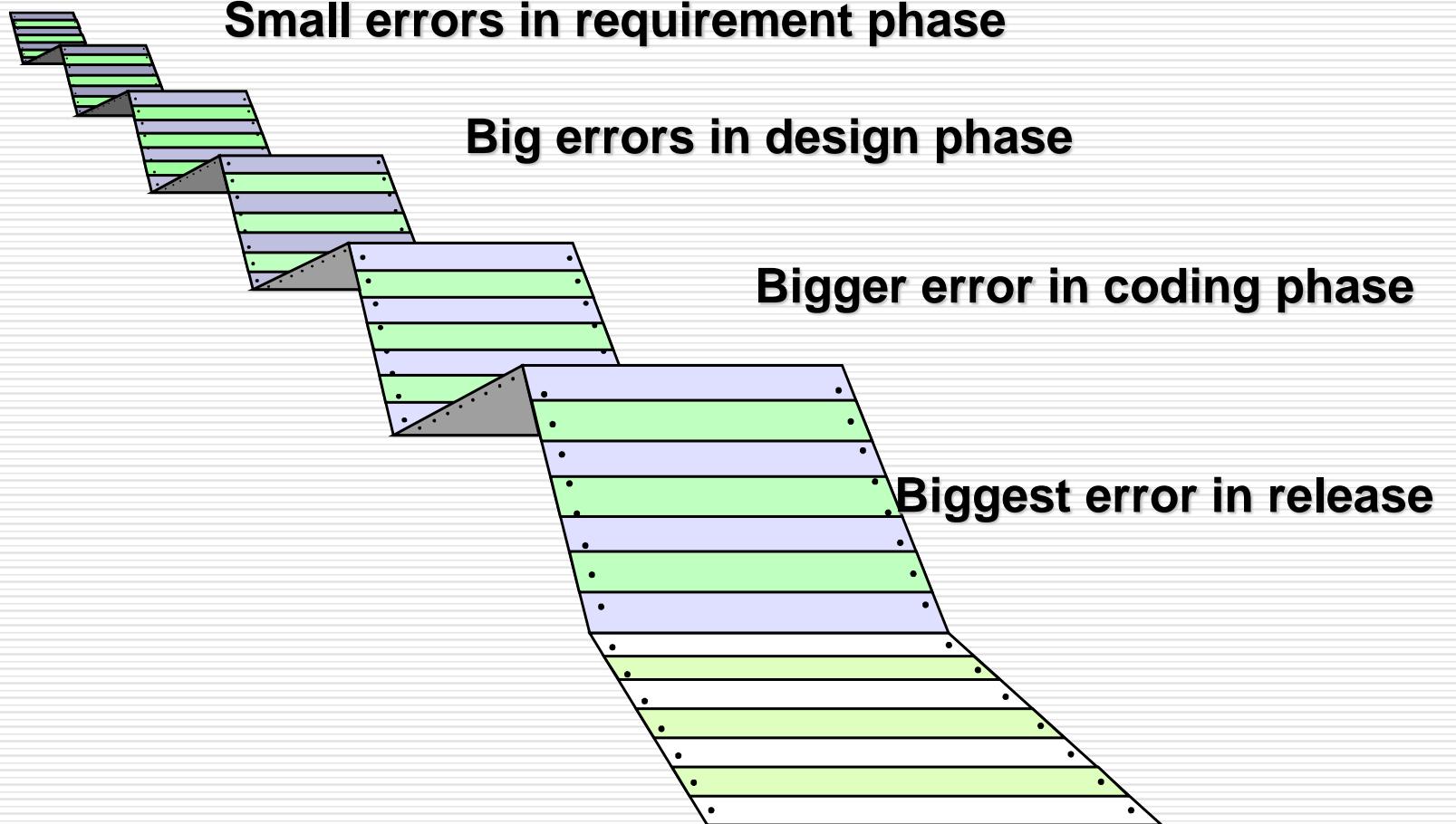
applied at each stage in the software process.

程序测试 VS. 软件测试

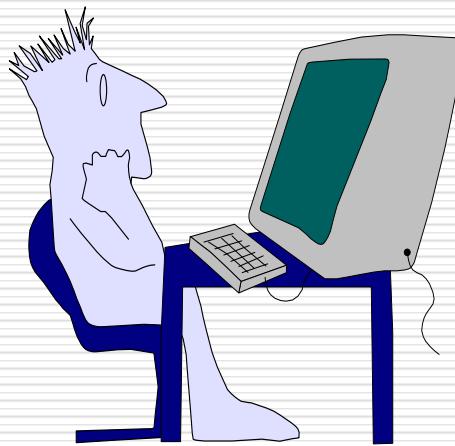
- 软件测试不同于程序测试；
- 软件测试应贯穿于软件定义与开发的整个期间；
- 据美国一家公司统计，查出的软件错误中，属于需求分析和软件设计的错误约占 64%，属于程序编写
的错误仅占 36%。程序编写的许多错误是“先天的”。

如何测试， 测试原则

- ✓ Tests should be planned long before testing begins
- ✓ All tests should be traceable to customer requirements
- ✓ Testing as early as possible
- ✓ To be most effective, testing should be conducted by an independent third party
- ✓ The developer of software should do no testing at all.
- ✓ Test cases should include not only input data, but also output data

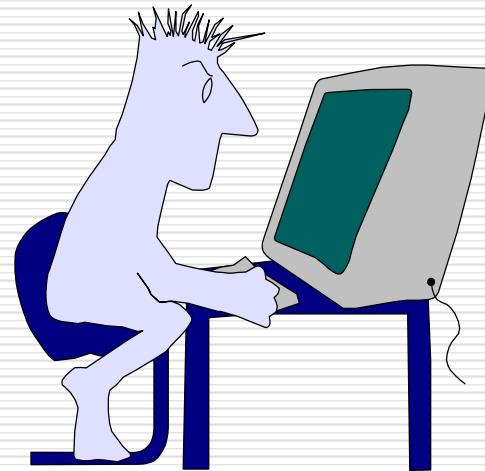


Who Tests the Software?



developer

**Understands the system
but will test “gently” and
is driven by delivery**



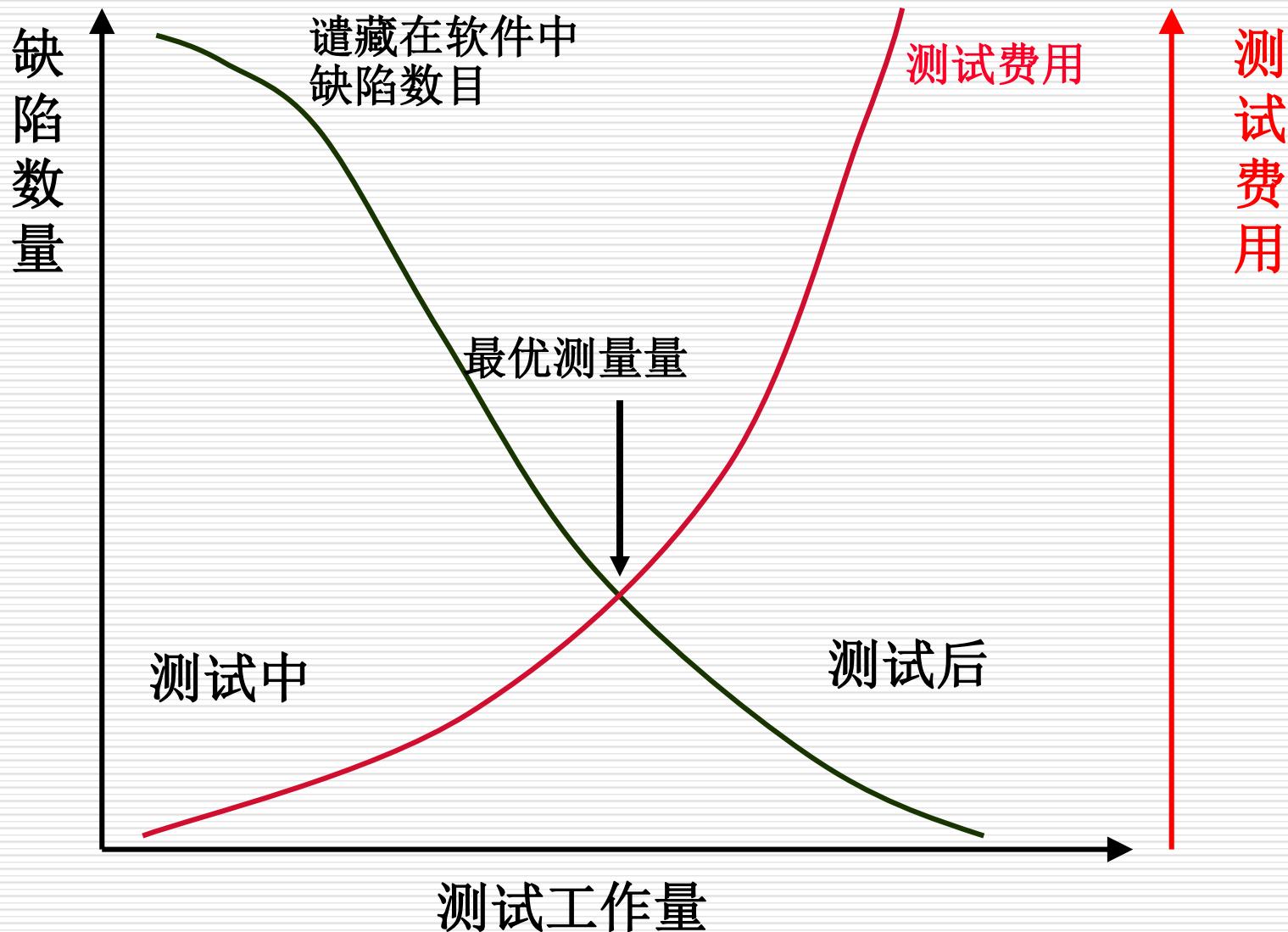
independent tester

**Must learn about the
system but will attempt
to break it and is
driven by quality**

测试原则(cont.)

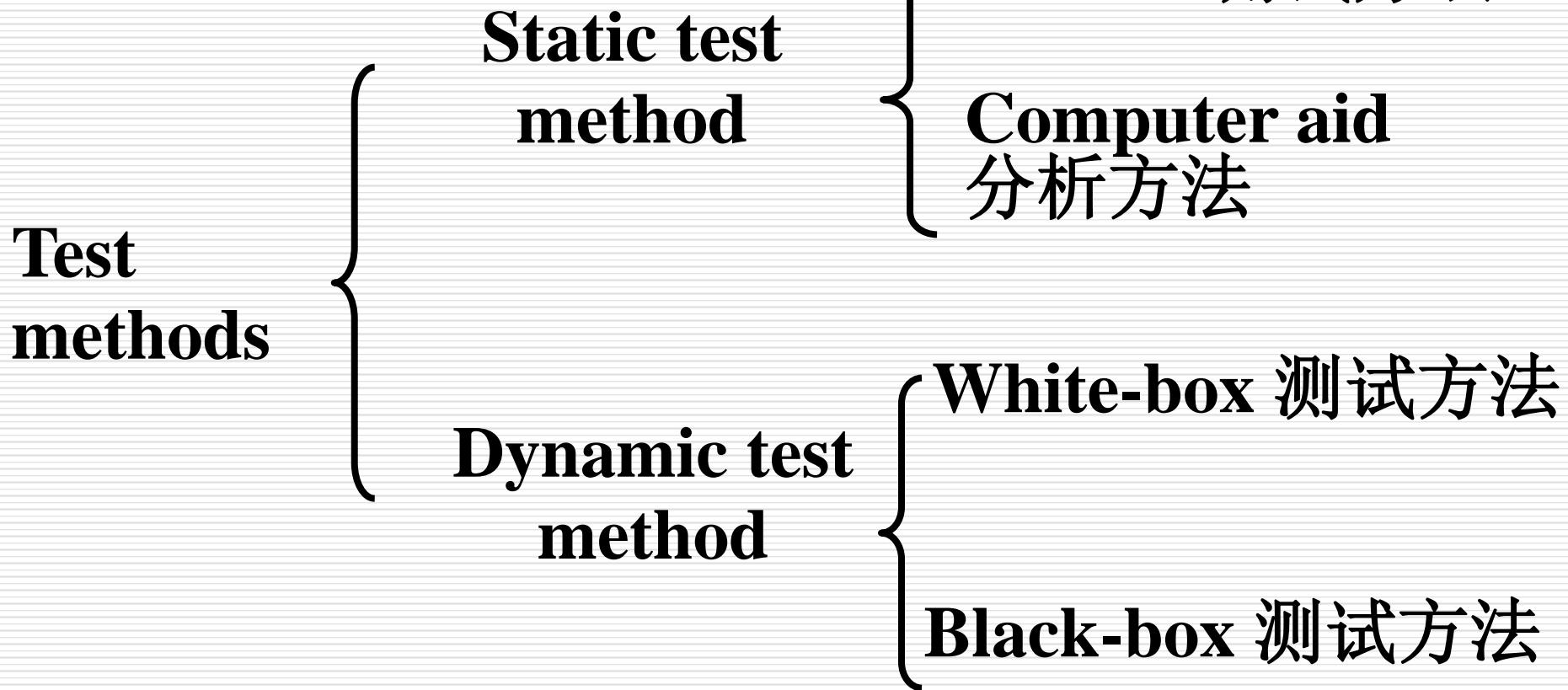
- ✓ Test need normal testing cases as well as exceptional one
- ✓ Regressive test is necessary when code modified
- ✓ 80% of errors occur in 20% of classes
- ✓ Testing should begin “in the small” and progress toward testing “in the large”
- ✓ Exhaustive testing is not possible
- ✓ Quality of testing depend on test design
- ✓ Keep test documents till system given up

软件测试是有风险的行为



每一个软件项目都有一个最优的测量

如何测试：方法分类



Static test

- ✓ inspections
 - ✓ walkthroughs（走查）
 - ✓ reviews
- 基本特征是在对软件进行 **分析、检查和审阅**，不实际运行被测试的软件。
 - 静态测试约可找出**30~70%**的逻辑设计错误.
 - 对需求规格说明书、软件设计说明书、源程序做检查和审阅，包括：
 - ✓ 是否符合标准和规范；
 - ✓ 通过结构分析、流图分析、符号执行指出软件缺陷；

Dynamic test

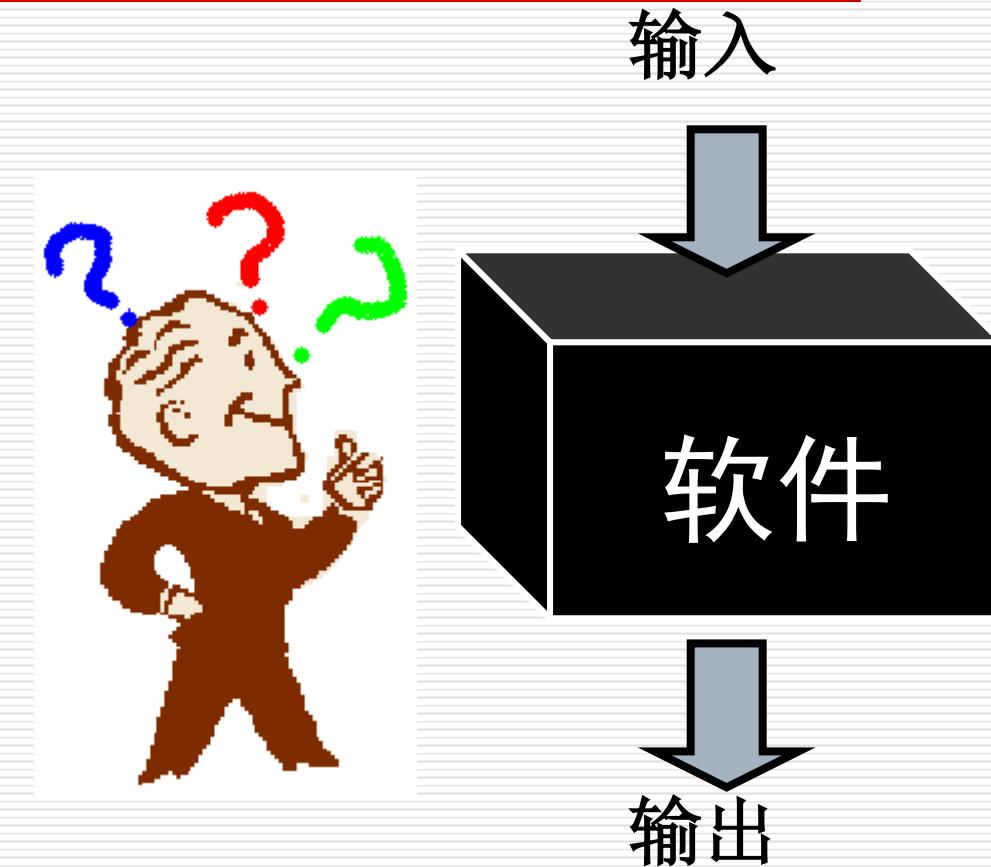
➤ 动态测试

通过运行被测程序来检验软件的动态行为和运行结果的正确性

➤ 动态测试的两个基本要素：

- ✓ 被测试程序
- ✓ 测试数据（测试用例）

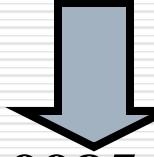
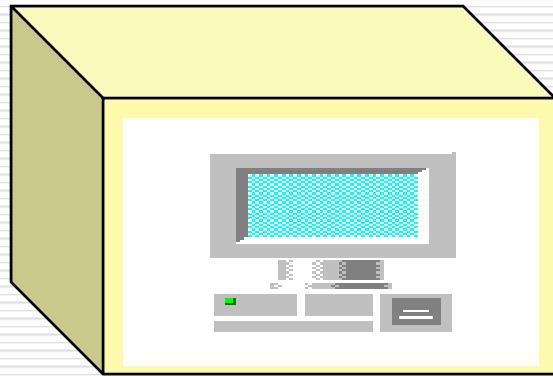
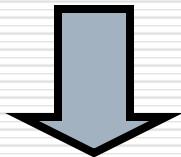
动态黑盒测试 —— 闭着眼睛测试软件



不深入代码细节的测试方法称为动态黑盒测试。
软件测试员充当客户来使用它。

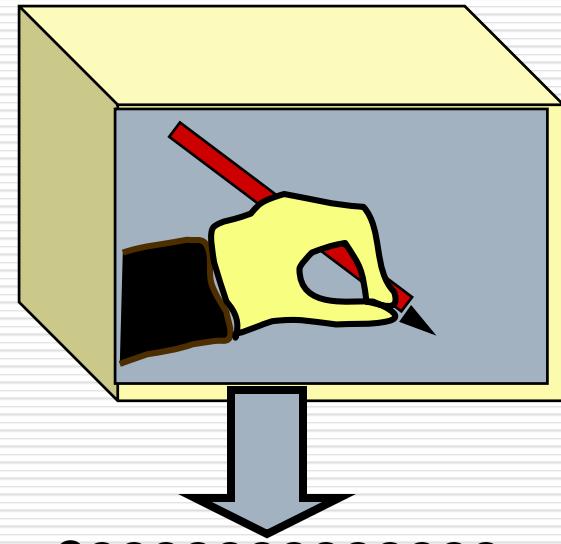
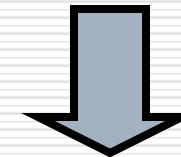
动态白盒测试 —— 带X光眼镜测试软件

$250 * (1 + 0.015) * ((1 + 0.015)^{360} - 1) / 0.015$



3581322.293419985680302829734315

$250 * (1 + 0.015) * ((1 + 0.015)^{360} - 1) / 0.015$



?????????????????

假如知道一个盒子包含一台计算机, 而另一个盒子是人用纸笔计算, 就会选择不同的测试用例

了解软件的运作方式会影响测试手段

test types (通俗用语, 非专业用语)

- 黑盒测试
- 白盒测试
- 单元测试
- 累计综合测试
- 集成测试
- 功能测试
- 系统测试
- 端到端测试
- 健全测试
- 回归测试
- 接受测试
- 负载测试
- 强迫测试
- 性能测试
- 可用性测试
- 安装/卸载测试
- 恢复测试
- 兼容测试
- 安全测试
- 比较测试
- Alpha测试
- Beta测试

