



# 软件质量保证与测试

## 软件测试过程

# 内容提要

- 软件产品组成
- 软件测试的模型
- 软件测试的目的和原则
- 软件测试的术语
- 软件测试过程
- 完整的软件测试系统

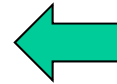
# 内容提要

- **软件产品组成**
- 软件测试的模型
- 软件测试的目的和原则
- 软件测试的术语
- 软件测试过程
- 完整的软件测试系统

## 软件开发的组成部分

- ❑ 软件产品绝不是一张光盘或一个程序，很多隐藏的过程和附件对于软件产品而言非常重要。
  - 比如，MATLAB的说明书（离开这个说明书我们将无法使用该软件）
- ❑ 对于测试员而言，必须了解软件开发的整个过程和软件产品的所有组成部分，才能够保证对整个软件产品进行测试。

## 软件开发的隐藏部分



客户需求  
开发进度  
测试计划  
产品审查  
用户反馈  
易用性数据  
设计文档  
竞争对手情况  
观察和感受说明书

软件产品包含大量看不见的努力

## □ 用户需求

### 任务

- 确定用户需求：功能、性能、运行环境约束

重要性：软件开发依据，软件验收的标准

困难性：难以说清、动态变化、歧义、复杂

技术途径和工具

- 需求分析人员需与用户不断、反复地交流和商讨，使用用户需求逐步准确化、一致化、完全化

- 抽象、问题分解、快速原型、多视点等技术

阶段性产品

- 软件需求规格（功能，性能和运行环境约束）

## □ 产品说明书

在对用户的需求进行了解之后，需要进行整理，得到产品规格说明书（product specification），其定义的产品功能，外观等详细信息是开发的基础。

## □ 软件开发进度表

为了使软件产品按照指定时间和质量完成，需要制定软件开发的进度，这属于项目管理的范畴。

工具：

Microsoft Project

## □ 软件设计文档

在进行软件编码之前，要进行软件设计，这些设计文档包括：

- 软件构架——软件主要部分及其交互
- 数据流示意图——数据在程序中的流动
- 状态变化示意图——软件不同状态及其变化方式
- 流程图——描述软件执行的过程
- 注释——软件代码的说明

## □ 测试文档

在发布软件之前，需要对软件进行测试，以下是主要的测试文档：

- **测试计划**——软件测试的整体方案
- **测试用例**——软件测试项目的详细步骤
- **软件缺陷报告**——描述测试到的软件缺陷
- **归纳、统计和总结**——统计测试的结果，用图形、图表的方式表达结果

## □ 最终的软件产品

安装光盘  
用户手册  
帮助文件  
样本和示例

产品支持信息  
错误信息  
广告和宣传材料  
产品标签  
其他信息



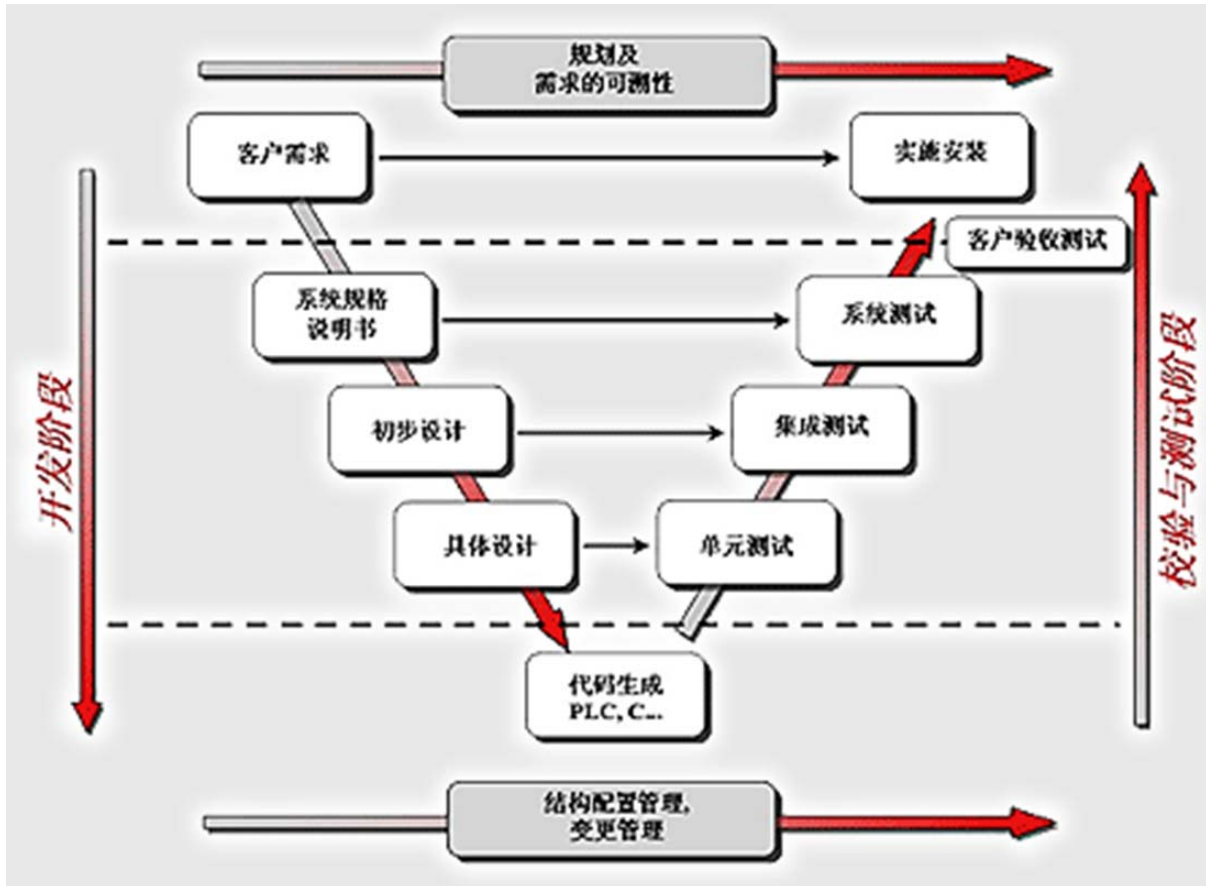
# 内容提要

- 软件产品组成
- 软件测试的模型
- 软件测试的目的和原则
- 软件测试的术语
- 软件测试过程
- 完整的软件测试系统

## 软件测试的V模型

□ 就像软件开发有过程模型一样，测试也有测试模型。

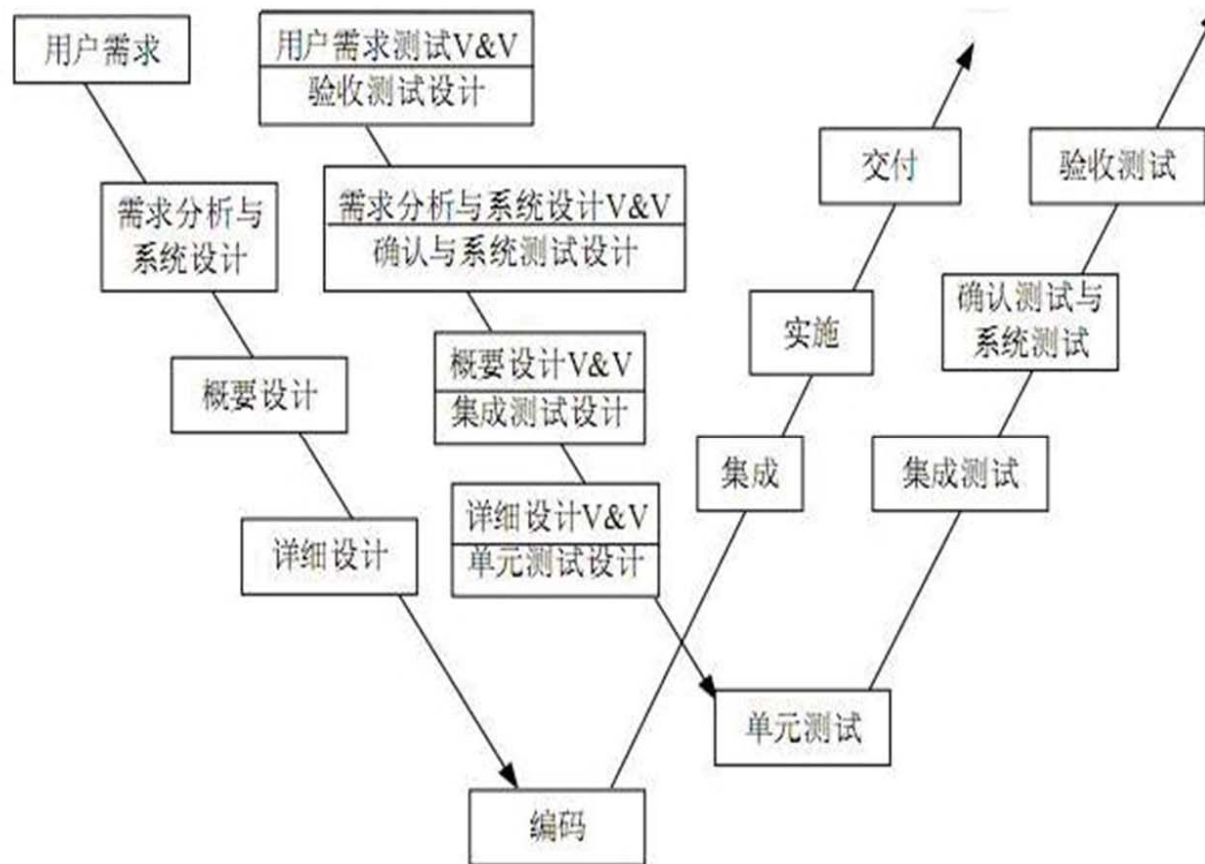
**V模型**是将软件开发过程和软件测试过程对应起来，最早由Paul.Rook提出。



**局限性:** 把测试作为编码之后的一个验收测试活动，需求分析等前期产生的错误直到后期的验收测试才能发现。无法体现“尽早地和不断地进行软件测试”的原则

## 软件测试的W模型

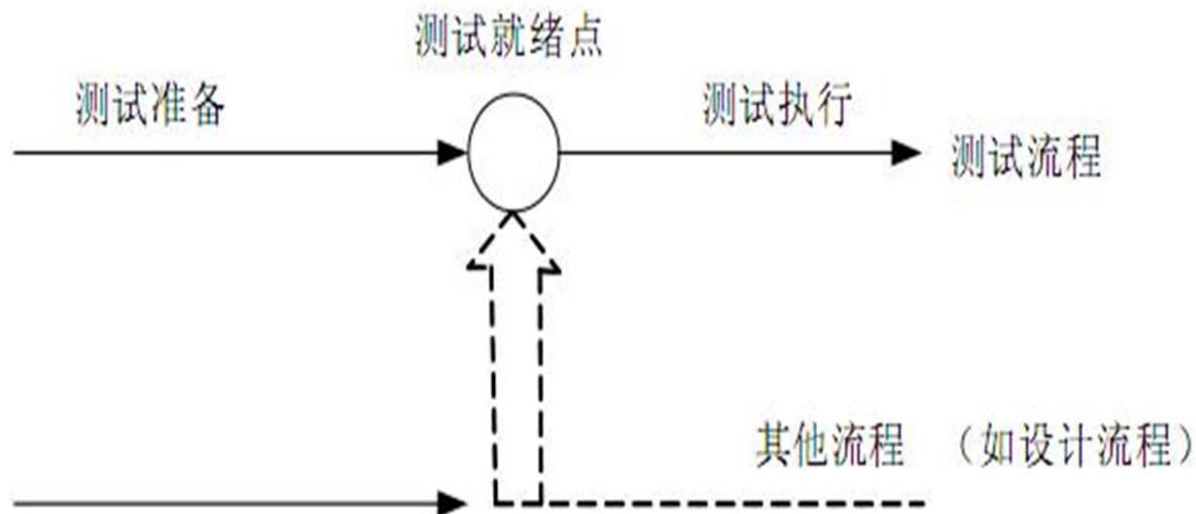
**W模型**由Evolutif公司提出，W模型增加了软件开发各个阶段的验证和确认活动，避免将最初的错误设计带入到代码中之后再再进行验证，造成程序难于回溯——测试与开发同步进行。



**局限性：**在W模型中，需求、设计、编码验收等测试活动被视为串行的，同时，测试和开发活动也保持着一种线性的前后关系，上一阶段完全结束，才可正式开始下一个阶段工作。这样就无法支持迭代的开发模型。

## 软件测试的H模型

在**H模型**中，软件测试是一个独立的流程，贯穿于整个产品周期，与其他流程并发进行，当某个测试时间点就绪时，即从测试准备阶段进入测试执行阶段

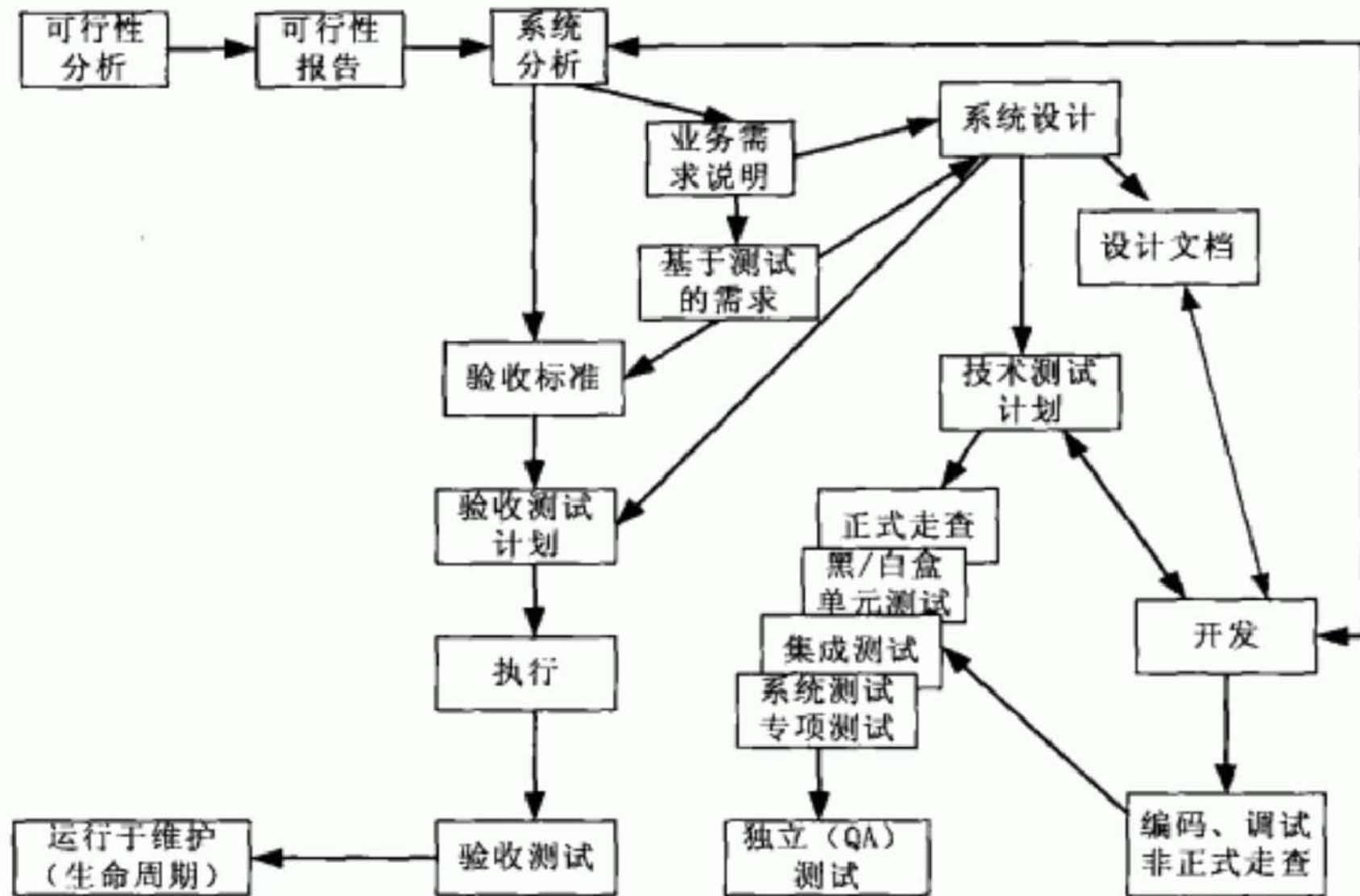


**局限性：**过于模型化，重点在于理解其中的意义以指导实际工作，而模型本身并无太多的可执行的指导意义。

## 前置测试模型

- 由RobinFGoldsmith等人提出，是一个将测试和开发紧密结合的模型。
- 将开发和测试的生命周期整合在一起，表示了项目生命周期从开始到结束之间的关键行为。对每一个交付内容进行测试，在设计阶段进行测试计划和测试设计，让验收测试和技术测试保持相互独立。
- 价值体现：用较低的成本来及早发现错误，并且充分强调了测试对确保系统的高质量的重要意义。

## 前置测试模型



# 内容提要

- 软件产品组成
- 软件测试的模型
- *软件测试的目的和原则*
- 软件测试的术语
- 软件测试过程
- 完整的软件测试系统

## 软件测试的目的

- 软件测试就是在软件投入运行前，对软件的需求分析、设计、实现编码进行最终审查。
  - 表面上看，软件工程的其他阶段都是建设性的，而软件测试是摧毁性的。
- 但是，软件测试的最终目的是建立一个高可靠性的软件系统的一部分。

定义：软件测试就是为了发现缺陷而运行程序的过程。



# 软件测试的目的

- 著名软件工程专家G. J. Myers将软件测试的目的归纳：
  - 测试是程序的执行过程，目的在于发现错误；
  - 一个好的测试用例在于能发现至今未发现的错误；
  - 一个成功的测试是发现了至今未发现的错误的测试。

# 软件测试的原则

- 测试的独立性原则（心理学）
- 测试应尽早介入（经济学）
- 完全测试程序是不可能的
- 软件测试是有风险的行为
- 找到的缺陷越多，就说明软件缺陷越多
- 软件缺陷有免疫力
- 并非所有软件缺陷都能修复
- 产品说明书不断变化
- 软件测试员在产品小组中不受欢迎
- 软件测试需要技术。

## 测试的独立性原则（心理学）

- ❑ 程序员通常不能高效测试自己的程序，因为很难适应这种心理，即尽量暴露自己的错误，这种掩饰行为降低了测试的效率；
- ❑ 测试员的工作则是暴露程序的缺陷，他们担心的是找不到错误，在这种心理下，测试员会想尽办法找出程序的错误。
- ❑ 测试的独立性原则包含两个方面的意思：
  - 程序员应该避开测试自己的程序
  - 程序开发组织应避开测试自己开发的程序

## 测试应尽早介入（经济学）

- ❑ 早期的缺陷在其后的开发和发布过程中会不断被放大，需要很高的修复费用，还可能因为会触及到整个软件的构架而无法修复；
- ❑ 美国软件质量安全中心2000年对美国100家知名的软件厂商统计，软件缺陷在开发前期发现比开发后期发现在资金、人力上节约90%；软件缺陷在推向市场前发现比在推出后发现，在资金、人力上节约90%。

# 不可能完全测试程序

不能够完全测试软件的原因：

1. 输入量太大
2. 输出结果太多
3. 软件实现的路径太多
4. 软件说明书没有客观标准

所有的输入：

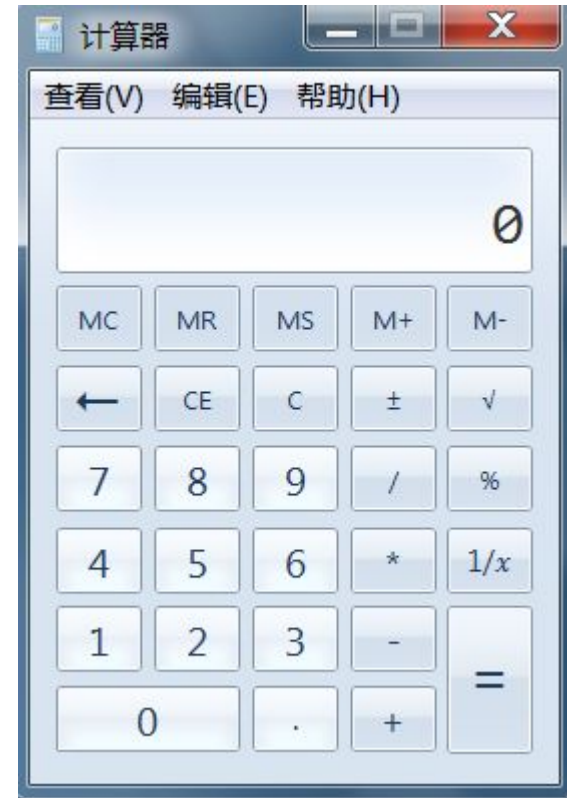
1+1, 1+2...

1+999999999999

9

2+1, 2+2

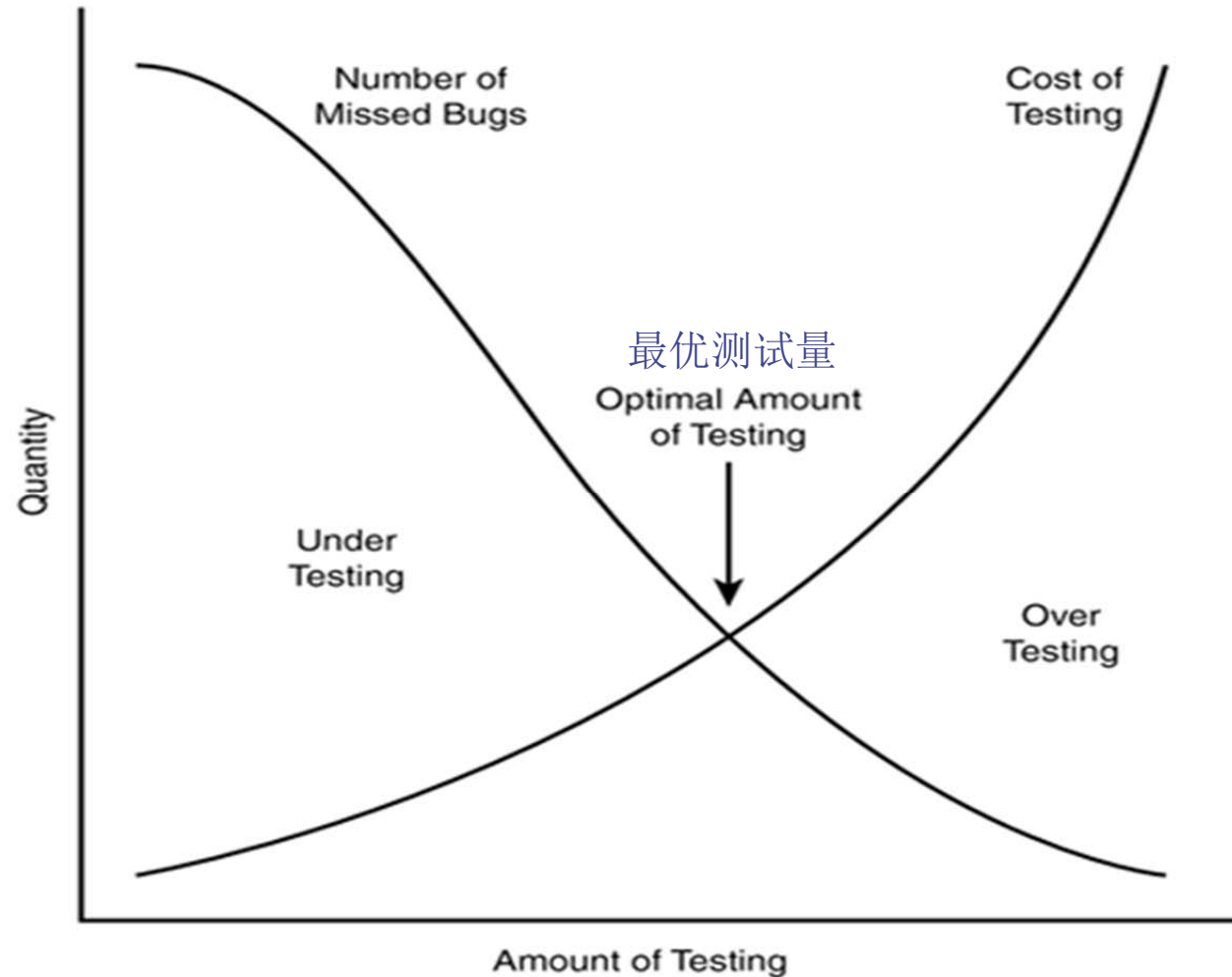
...



正确性 VS 完备性

# 软件测试是有风险的行为

- 由于不能进行完全测试，某些隐藏的缺陷就是风险；
- 软件测试员需要学习把无穷的测试减少到可以控制的范围，以及针对风险采用什么样的明智决策，去粗存精。



## 找到的软件缺陷越多，则实际存在的缺陷越多

其中的原因是：

- 程序员倦怠
- 程序员容易犯同样的错误，比如，数据越界、内存泄露等
- 某些软件缺陷是其它软件缺陷的征兆

**祸不单行！！**

## 杀虫剂现象——抗药性

- ❑ 采用某种农药消灭害虫，后来发现害虫对这种农药具有了免疫力，这就是**杀虫剂现象**
- ❑ 对于软件，如果一直使用同样的测试方法，只能查找到同样问题，而其它软件缺陷可能被忽略
- ❑ 为了克服这种现象，需要尝试不同的测试方法，以便找出更多的软件缺陷



# 并非所有的软件缺陷都能修复

其中的原因是：

- 没有足够的时间
- 不能算是真正的软件缺陷
- 修复风险太大，修复一个错误可能引入其他软件缺陷
- 有些错误不值得去修复，这归结为商业风险决策

## 如何应对产品需求的变化

- 由于IT业的发展迅速，导致一些软件在完成之前产品的需求已经发生变化，比如竞争对手推出了新的功能，此时，我们应该如何应对？
- 软件测试员必须要想到产品需求的可能变化，因此需要灵活制定测试计划和执行测试技术（自动化测试）

软件说明书（需求）一直在变化

## 软件测试人员在开发团队里“讨人嫌”

- 软件测试员的任务是找出软件的缺陷，公布发现的问题，相当于监督员
- 为了克服监督员不受欢迎的局面，测试员可以采取以下措施：
  - 尽可能早地找出软件缺陷，这样容易改进
  - 控制情绪，不要把自己找到软件缺陷的喜悦在相应的程序员面前过度表达
  - 不要总报告坏消息，需要及时沟通

# 软件测试是一项讲究条理的技术

□ 软件测试已经成为软件行业的强制要求，软件测试人员被作为开发小组不可缺少的核心小组成员，软件测试的专业技术得到了有效的发展。

## 其它软件测试的原则

- 所有测试都应追溯到用户需求
- 应在测试工作开始前设计测试计划
- 测试应从小规模到大规模
- 避免不能重现或匆忙的测试
- 深入细致地审查测试结果
- 不要为了测试方便而修改程序
- 好的测试用例具有较高的发现过去未被发现过的
- 错误的概率，而不应只表明程序运行正常

## 对软件测试工作的误解

- 软件测试技术要求不高，至少比编程容易多了。
- 软件测试随便找一个能力差的人就能做。
- 有时间就多测试一些，来不及就少测试一些。
- 软件测试是测试人员的事，与开发人员无关。
- 设计—实现—测试，软件测试是开发后期的一个阶段。软件开发完成后才能进行测试。
- 软件测试是没有前途的工作，只有程序员才是程序高手。

## 练习

1. 对程序的测试最好由谁来做，对程序的调试最好由谁来做（ ）。

- A) 程序员      第三方测试机构
- B) 第三方测试机构      程序员
- C) 程序开发组      程序员
- D) 程序开发组      程序开发组

2. 经验表明，某模块与其他模块相比，若该模块已发现并改正的错误较多，则该模块中残存的错误数目与其他模块相比，通常应该（ ）。

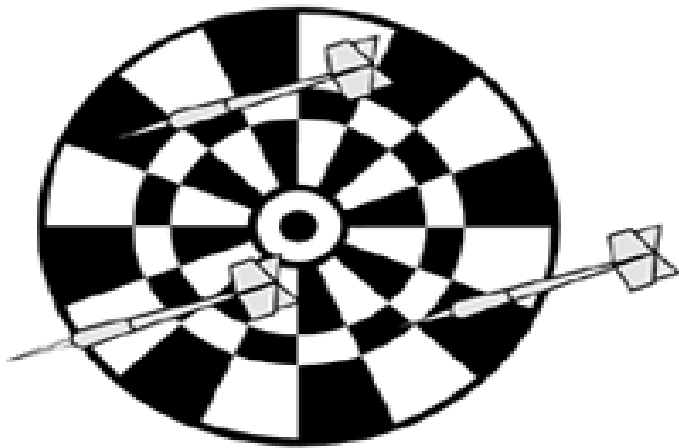
- A) 较少 B) 较多 C) 相似 D) 不确定

# 内容提要

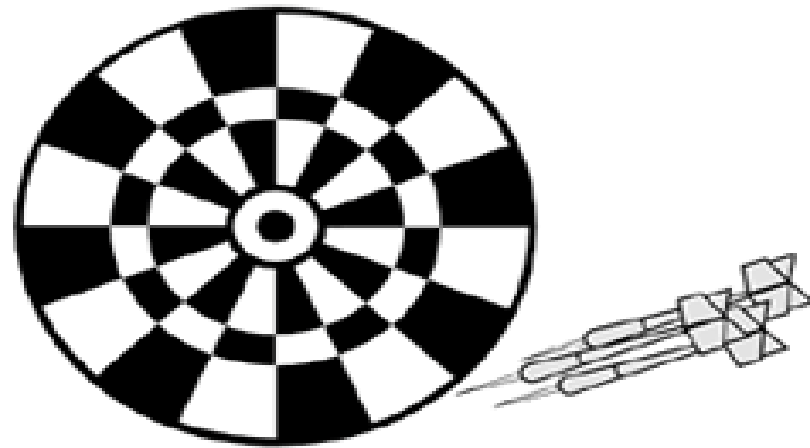
- 软件产品组成
- 软件测试的模型
- 软件测试的目的和原则
- *软件测试的术语*
- 软件测试过程
- 完整的软件测试系统



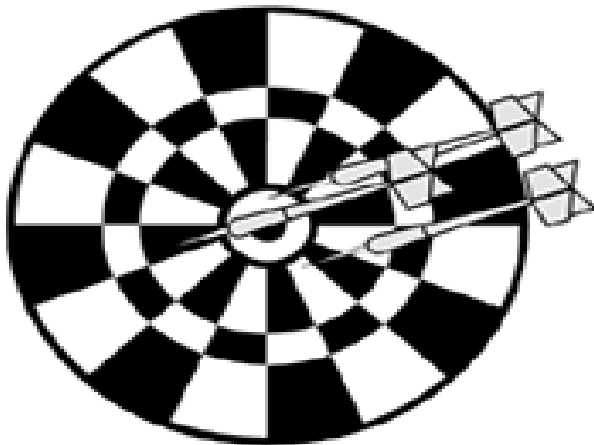
# “精确” 和 “准确” (Precision and Accuracy)



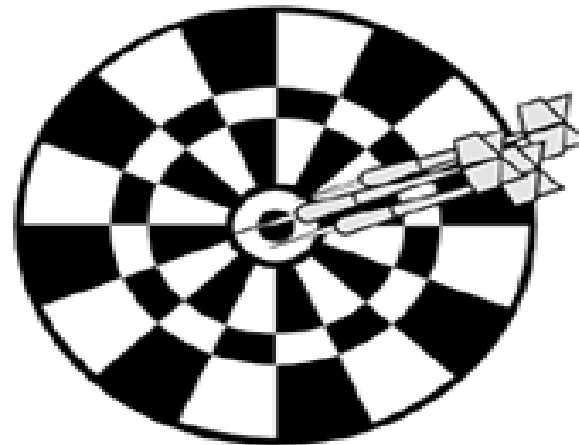
Neither Accurate nor Precise



Precise, but not Accurate



Accurate, but not Precise



Accurate and Precise

## “验证”和“合法性检查” (Verification and Validation)

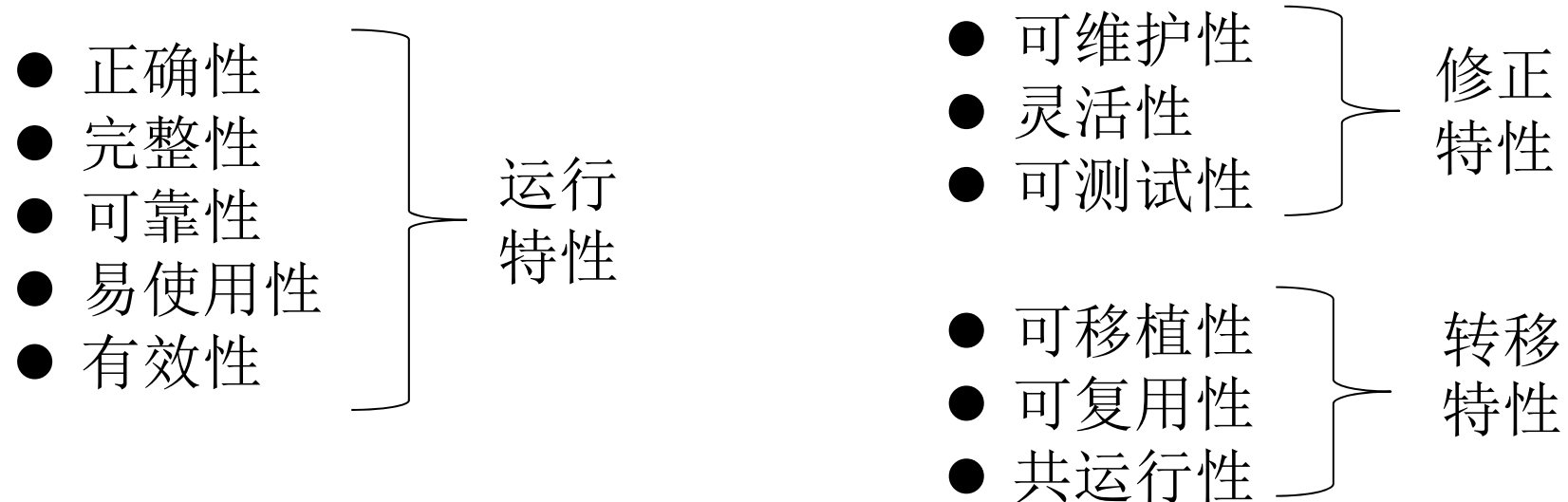
- Verification(验证) is the process confirming that software meets its **specification**.
  - Are we building the product right?
  - 软件是否满足需求说明
  
- Validation(确认) is the process confirming that it meets the **user's requirements**.
  - Are we building the right product?
  - 软件是否满足用户的需求

# “质量”和“可靠性” (Quality and Reliability)

## 软件质量

- 满足软件需求;
- 满足规定的标准, 并满足隐含的需求;
- 软件质量是客户满意度的体现;

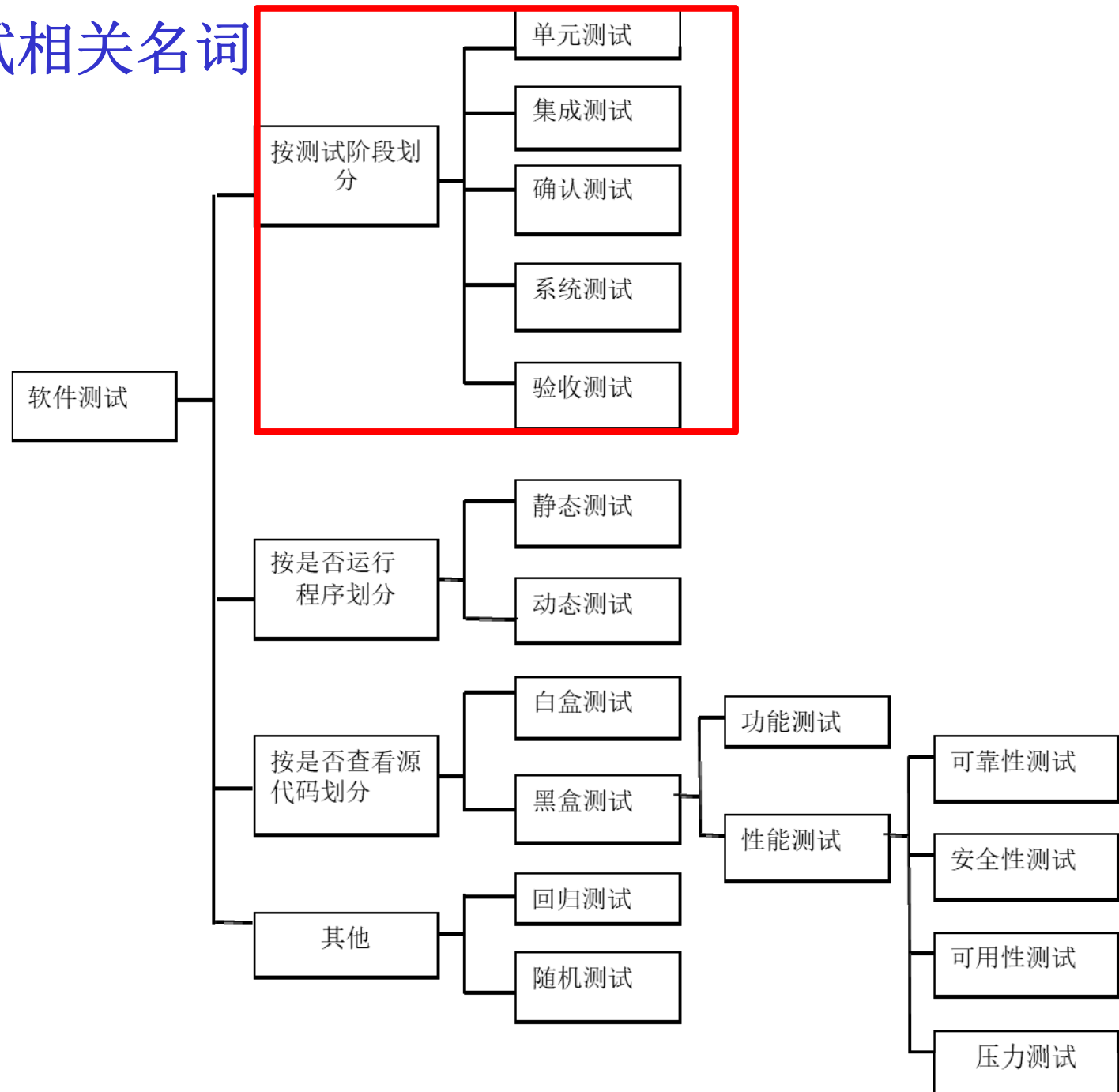
## 软件质量特性:



## “质量”和“可靠性” (Quality and Reliability)

- 可靠性：是质量的一个方面
  - 定义为：在规定的条件下和规定的时间内，计算机程序完成规定功能（无故障运行）的概率。
- 实践证明，对软件进行充分的测试才能够有效的保证软件质量。

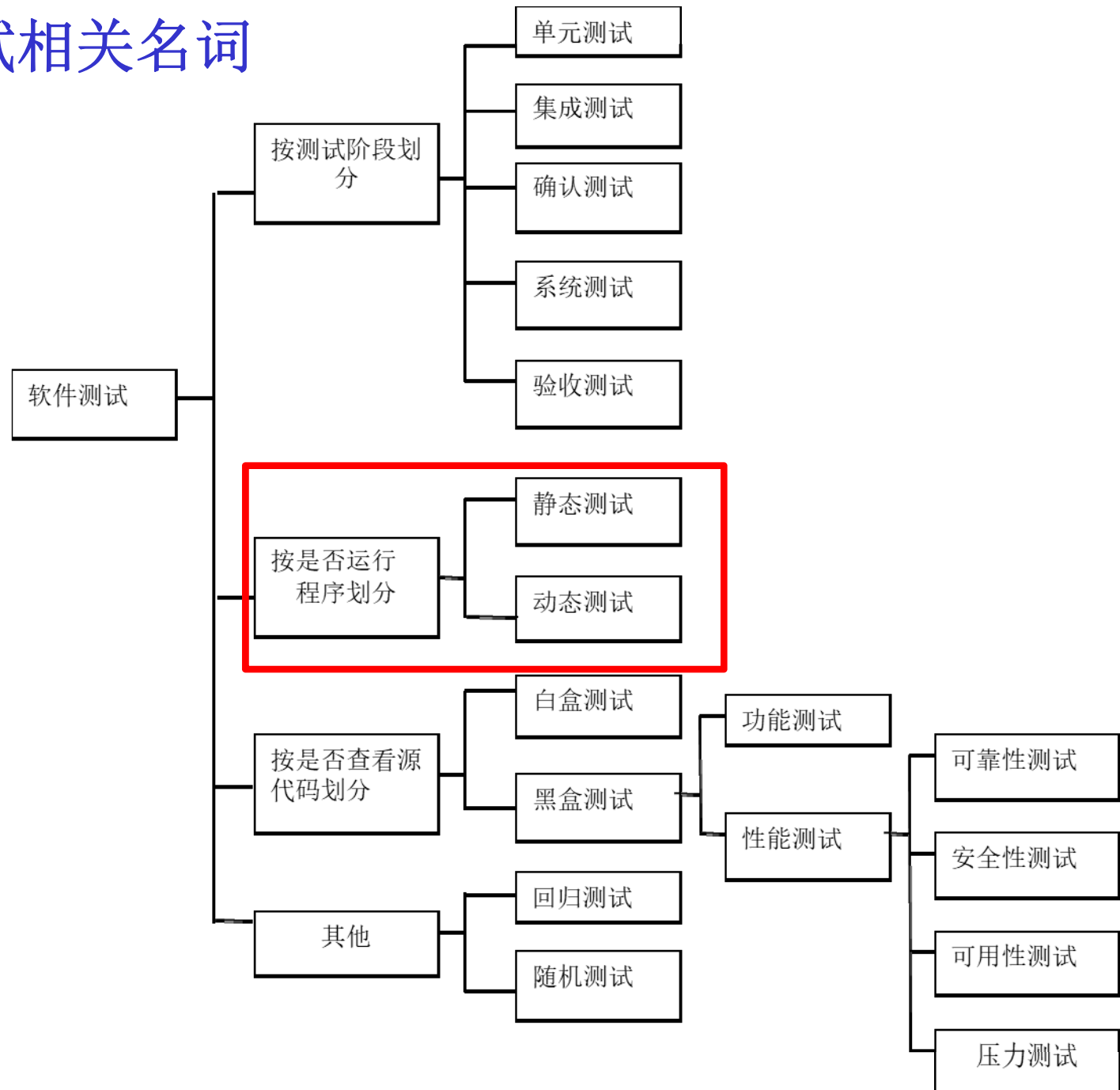
# 软件测试相关名词



## 测试阶段

- ❑ 单元测试(unit testing): 对每个模块的考察、测试;
- ❑ 集成测试(integration testing): 组装测试, 联调, 考察模块间的接口和联系;
- ❑ 系统测试(system testing): 把计算机和其他部件联结起来, 考察软件是否满足系统总的功能要求和性能要求;
- ❑ 验收测试(acceptance testing): 用户为主的测试。

# 软件测试相关名词



## 静态测试和动态测试

- 静态测试：不执行程序，仅分析、检查和审阅系统的相关文档，如需求文档、系统设计图、程序代码等。
- 动态测试：运行并使用软件，通过测试用例，比较实际输出与预期输出，以发现存在的还未被揭示的错误。

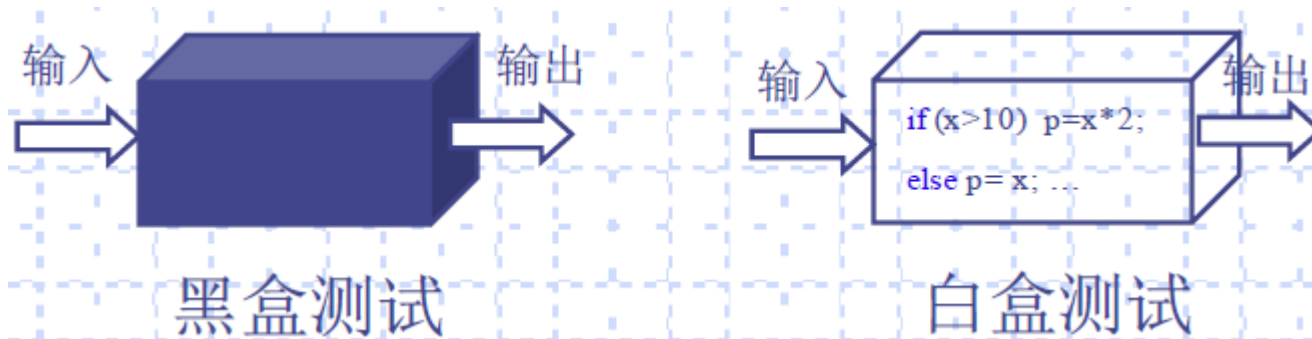


# 软件测试相关名词

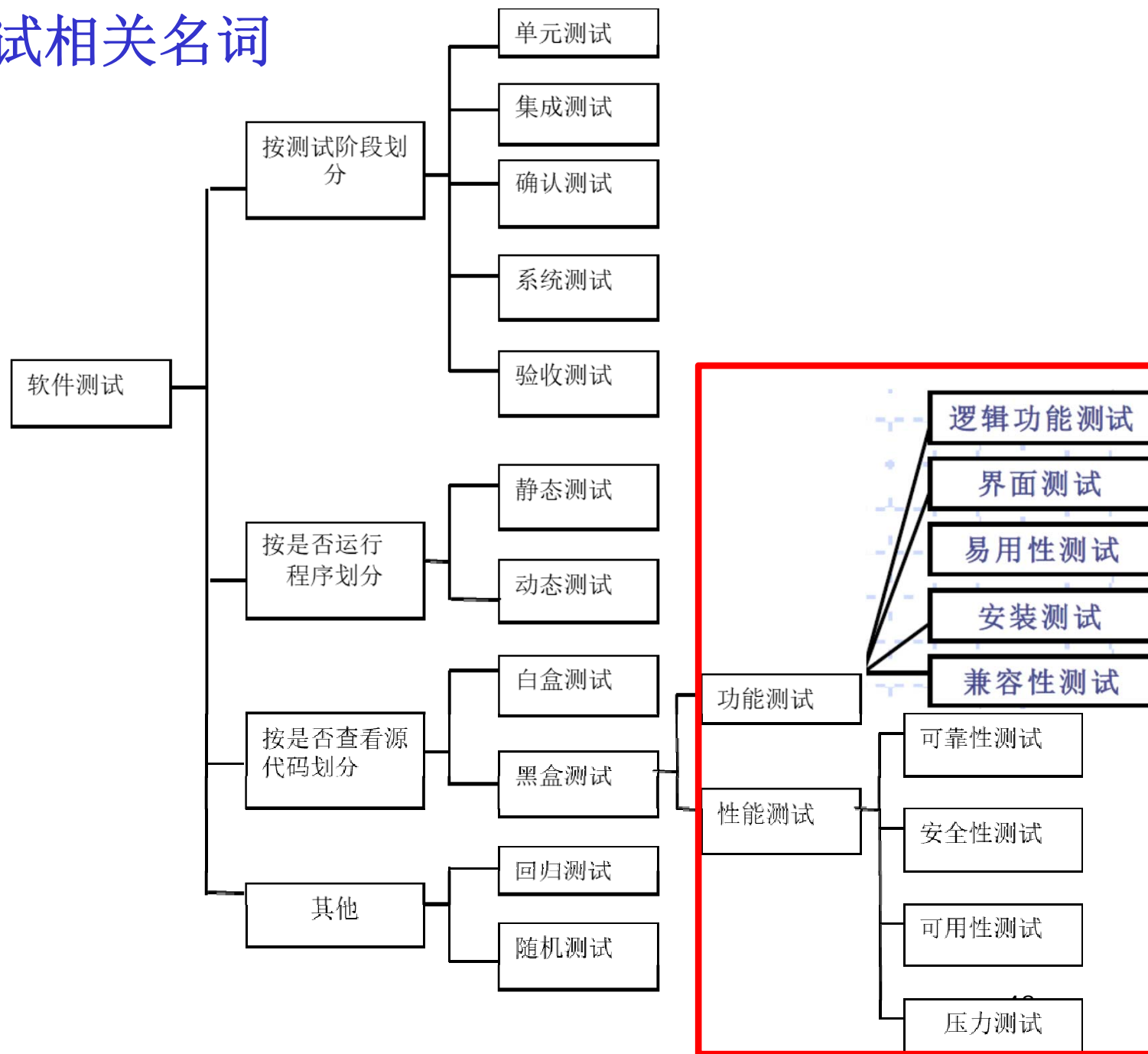


## 黑盒测试和白盒测试

- **黑盒测试**——是**基于需求说明书**的软件测试，在这种测试下，不需要了解软件的内部结构，以及软件代码的具体实现
- **白盒测试**——了解软件的**内部结构**以及实现的具体代码，从而可以根据程序的内部情况来设计更具有针对性的测试案例



# 软件测试相关名词



# 功能测试

- ❑ 是黑盒测试的一方面，检查软件功能是否符合用户的需求。
- ❑ 功能测试是黑盒测试中最重要的测试，比性能测试更加重要。因为软件的性能比较差用户还可以忍耐，但如果软件实现的根本不是用户所需要的功能……
- ❑ 功能测试分为：逻辑功能测试、界面测试、易用性测试（残疾人）、安装测试、兼容性测试。

# 性能测试

- 是软件测试的高端领域，性能测试工程师的待遇和白盒测试工程师不相上下。通常招聘时所说的高级软件测试工程师一般就指性能测试工程师或白盒测试工程师。性能测试一般要用到自动化测试工具。
- 软件性能包括很多方面，主要有时间性能和空间性能两种。主要有一般性测试、稳定性测试、负载测试、压力测试。

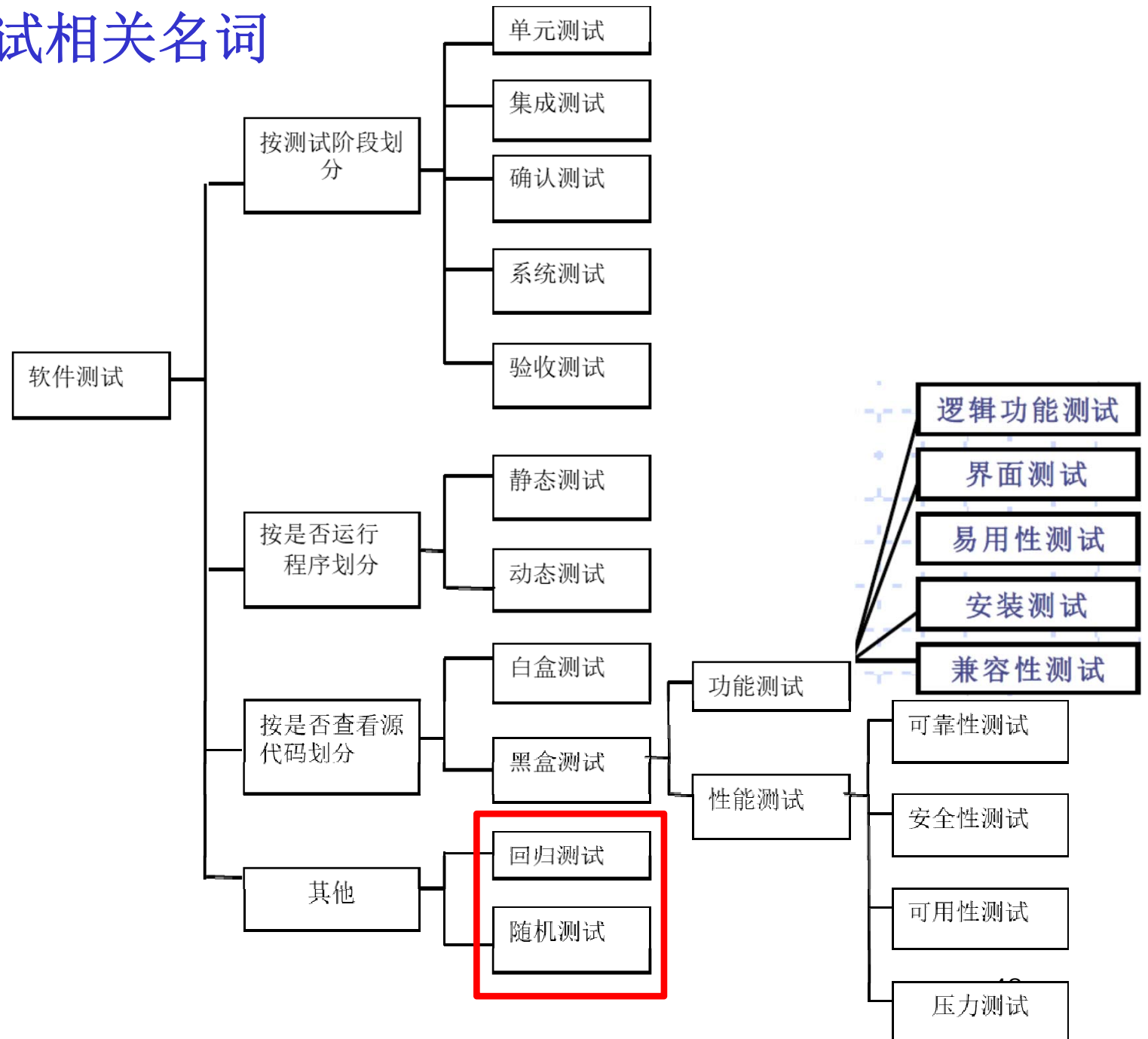
## 性能测试——负载测试

- 通过测试系统在资源超负荷情况下的表现，以发现设计上的错误或验证系统的负载能力。
- 负载测试中，将使测试对象承担不同的工作量，以评测和评估测试对象在不同工作量条件下的性能行为，以及持续正常运行的能力。
- 负载测试的目标是确定并确保系统在超出最大预期工作量的情况下仍能正常运行。

## 性能测试——压力测试

- 压力测试的基本思路：在计算机数量较少或系统资源匮乏的条件下运行测试。
- 通常要进行压力测试的资源包括内存、CPU可用性、磁盘空间和网络带宽等。一般用并发来做压力测试。

# 软件测试相关名词





## 回归测试

- 对软件的新版本进行测试，可能重复执行上一个版本测试时的用例，也可能需要设计新的测试用例。
- 回归测试可以在任何测试阶段进行，既有黑盒回归也有白盒回归。
- 回归测试时如何选择测试用例是目前的一个研究热点。

# 冒烟测试

- ❑ 来源：电路板质量检查，通电看是否冒烟。
- ❑ 软件冒烟测试的目的：在对一个新版本进行系统大规模的测试之前，先验证一下软件的基本功能是否实现，是否具备可测试性；
- ❑ 在软件测试中，冒烟测试主要指测试小组对主要功能进行先期测试，如果连主要功能都没有实现，直接打回去进行开发。

## 随机测试

- 也叫猴子测试，测试中所有的输入数据都是随机生成的，其目的是模拟用户的正式操作，并发现一些边缘性的错误。
- 优点是能够发现一些意想不到的错误
- 随机测试的缺点与不足
  - 测试不系统，无法统计代码覆盖率和需求覆盖率，很难再现、回归测试。

## 练习

### 1. 判断对错

1) 黑盒测试的测试用例是根据应用程序的功能需求设计的。

2) 白盒测试可以更细致的检查软件的源代码，因此其测试效果优于黑盒测试。

### 2. 验收测试以（ ）文档作为测试的基础。

- |            |          |
|------------|----------|
| A. 需求规格说明书 | B. 设计说明书 |
| C. 源程序     | D. 开发计划  |

## 练习

3. 从技术角度分，不是一类的测试是( )

- A. 黑盒测试      B. 白盒测试
- C. 单元测试      D. 灰盒测试

4. 组装测试又称为( )

- A. 集成测试      B. 系统测试
- C. 回归测试      D. 验收测试

5. ( )需要考察模块间的接口和各模块之间的联系。

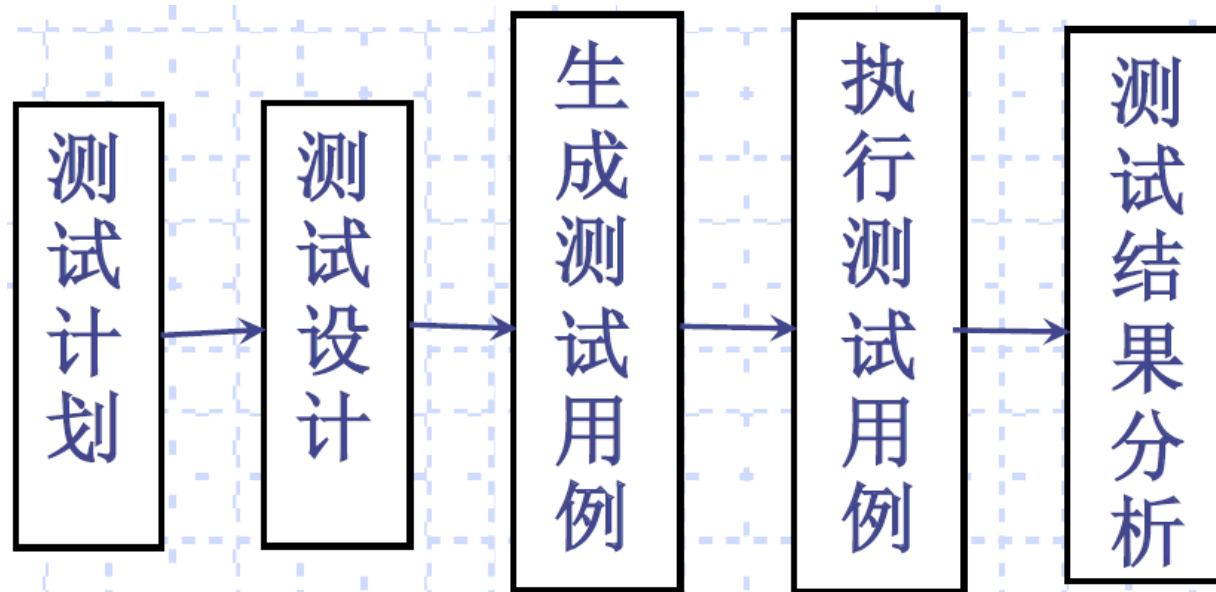
- A. 单元测试      B. 集成测试
- C. 验收测试      D. 系统测试

# 内容提要

- 软件产品组成
- 软件测试的模型
- 软件测试的目的和原则
- 软件测试的术语
- 软件测试过程
- 完整的软件测试系统

## 最简单的测试过程

- 很多人认为，测试软件就是运行软件，看软件是否出错。实际上，运行软件只是软件测试的一部分。
- 测试过程一般包括测试计划、测试设计、生成测试用例、执行测试用例、测试结果分析几个阶段。



## 一个软件测试的例子

对 “Hello world” 程序进行测试，程序代码如下：

```
main (int argc, char* argv[] )  
{  
    printf( “Hello world” );  
    return 0;  
}
```



## 测试过程

- (1) 测试计划：黑盒测试
- (2) 测试设计：运行程序，查看运行结果
- (3) 建立测试用例

测试Hello world程序的测试用例

用例编号	说明	操作过程	输入值	期望的结果
1	测试程序功能	运行软件	无	在控制台上打印出 "Hello world"

- (4) 执行测试用例：将Hello world 程序编译、连接，得到可执行程序Hello.exe，然后运行，因为测试不要求输入值，因此运行软件即是执行测试

## 测试过程

(5) 记录运行结果并分析：控制台上打印出Hello world字样



结果分析：测试的实际结果与期望的结果一致，程序的打印功能是正确的。

## 另一个软件测试的例子

对 “Hello world” 程序进行改进，程序代码如下：

```
void static main (int argc, char* argv[] )  
{  
    switch ( *argv[1] ) {  
        case „1“: // 输入1打印Hello world  
            printf( "Hello world!"); break;  
        case „2“: // 输入2打印Hello guy  
            printf( "Hello guy!"); break;  
        default: // 输入其它参数打印Hello  
            printf( "Hello!"); break;  
    }  
}
```

## 测试过程

- (1) 测试计划：黑盒测试
- (2) 测试设计：运行程序，检查不同输入参数（合理输入和不合理输入）情况下的运行结果
- (3) 建立测试用例

### 新的测试用例

编号	说明	操作过程	输入值	期望的结果
1	测试程序功能	运行软件	1	控制台显示 "Hello world!"
2	测试程序功能	运行软件	2	控制台显示 "Hello guy!"
3	测试程序功能	运行软件	3	控制台显示 " Hello! "
4	测试程序健壮性	运行软件	无	控制台显示 " Hello! "

# 测试过程

## (4) 执行测试用例

在控制台运行该程序，分别给程序带不同的输入参数，可执行程序的名字是Hello.exe

>Hello 1

>Hello 2

>Hello 3

>Hello

## 测试过程

(5) 记录运行结果并分析:

测试结果记录

编号	输入	期望结果	实际结果	通过/ 失败
1	1	打印"Hello world!"	打印"Hello world!"	通过
2	2	打印"Hello guy!"	打印"Hello guy!"	通过
3	3	打印"Hello!"	打印"Hello!"	通过
4	无	打印"Hello!"	弹出错误对话框	失败

## 测试过程

(5) 记录运行结果并分析（续）：

执行不带参数的Hello程序后，弹出错误对话框，程序崩溃  
分析一下被测程序

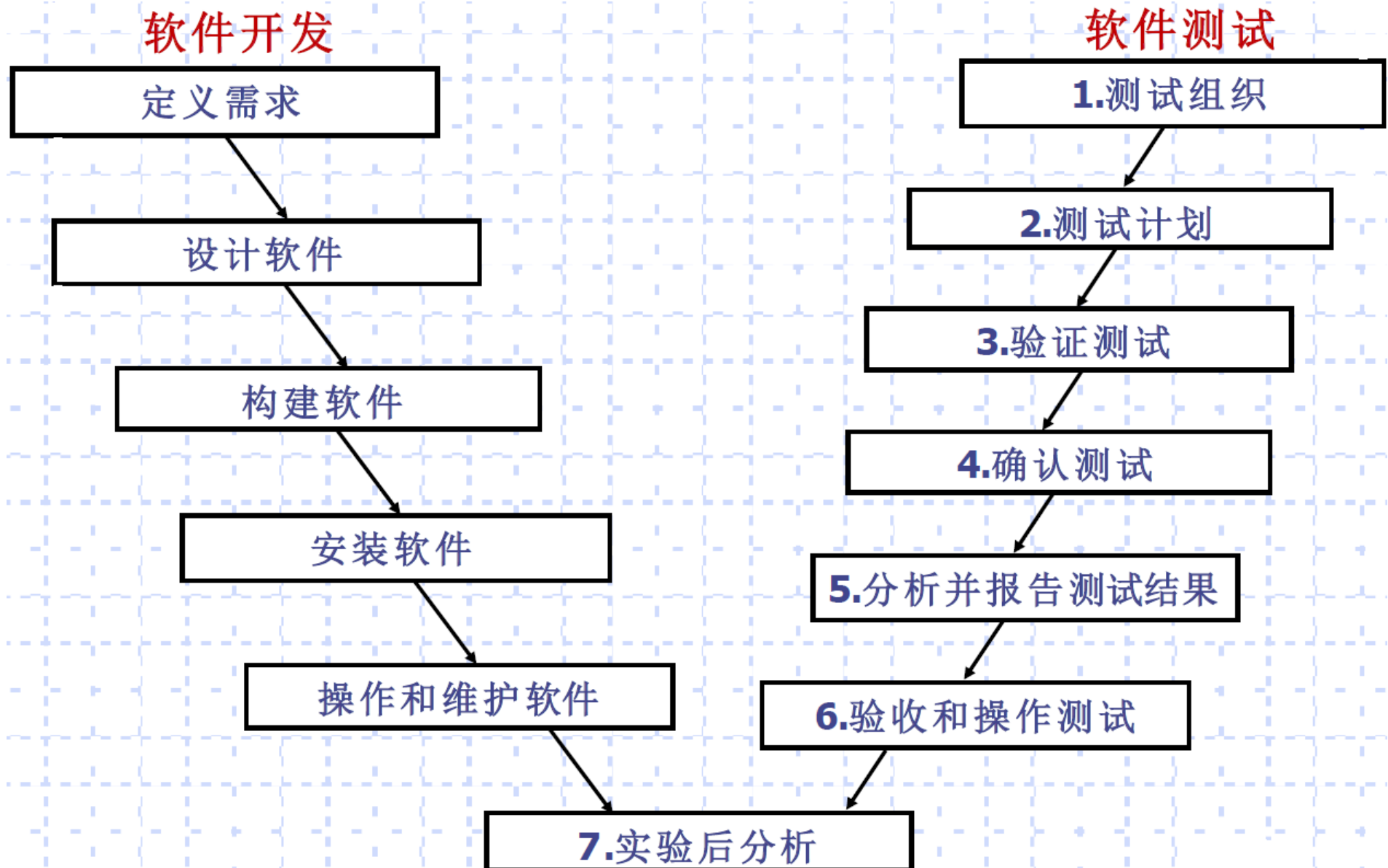
- ❑ 程序中使用了\*argv[1]，不输入任何参数时，argv[1]为空（NULL），程序访问空指针当然会出现问题；
- ❑ 如果在程序中对参数个数进行检查，就不会出现上面的问题
- ❑ 真是想不到，这么简单的一个程序也会出错
- ❑ 是的，即使我们认为再简单的程序也可能包含错误，软件测试可以找到这些错误

## Perry的7步软件测试流程

- ❑ 美国质量保证协会QAI的创始人之一William. E. Perry创建，基于1000多家公司的实际测试经验，这些公司都是QAI会员。
- ❑ 该测试过程是迄今为止最完善、最实用、最具有实际指导意义的测试模型。



# Perry软件测试流程图示



# Perry软件测试流程说明

- ❑ Perry的7步软件测试流程是与软件开发过程相对应的，相融合的，是一个顺序向下的一个过程
- ❑ 开发和测试不应当是完全独立的过程，将开发和测试整合为一个过程，对于开发高质量的软件而言至关重要

# Perry软件测试流程说明——组织测试

□对测试进行组织属于测试前的准备活动之一，它包含两个目的：

- 确定测试的内容

没有测试计划那么详细，只是确定测试的范围。比如只进行功能测试，需采用静态和/或动态测试等；

- 建立测试团队

确定测试团队中每个成员的任务。

## Perry软件测试流程说明——测试计划

- 测试计划也属于测试准备活动之一，它是测试工作的详细指南，包含测试风险分析、策略、方法、资源、进度等重要内容；
- 测试计划将确定测试的具体目标，测试工作都应该是按计划进行的，没有测试计划的测试很难到达测试目的。

# Perry软件测试流程说明——验证测试（软件评审）

□验证测试是高层测试，对软件需求说明、设计等进行验证，目的是确保构建正确的系统。

□验证测试采用的技术通常是手动的，比如评审、走查、审查、需求跟踪、静态分析等等，这些静态技术的运用可以找到系统的高层缺陷。

# Perry软件测试流程说明——确认测试

□确认测试是利用各种测试技术和方法确定所构建系统的运行正确性

□确认测试是在系统上运行测试用例。这部分测试就是我们传统意义上理解的测试

确认测试包含3方面的任务：

- 建立测试用例（数据/脚本）
- 执行测试
- 记录测试结果

## Perry软件测试流程说明——分析并报告测试结果

- 分析和报告测试结果是测试的输出，目的是将测试中获得的信息告知相关人员。包括软件在哪些方面存在问题，也包括测试员的一些建设性意见。
- 通常测试人员要编写两种测试报告：中间报告和最终的测试报告。
- 测试报告应该阐述缺陷造成的利害关系，以利于尽快地纠正缺陷。

# Perry软件测试流程说明——验收和操作测试

□验收测试的目标是确定软件各个方面都能满足用户的需求.

□通过内部测试的软件需要在用户的实际环境中最后确认，只有在用户的实际环境中可用的系统才是真正有效地系统。



## Perry软件测试流程说明——实现后分析（测试评估）

□实现后分析是面向未来的。主要回答如下问题：测试是否有效？测试人员的表现如何？如何改进测试过程？等等。

□在每次软件测试结束后，通过评估测试效率可以最大限度地改进测试。

□这种评估除了由测试员参加之外，还应该邀请开发人员、软件用户以及质量专业人员共同参与。

# 内容提要

- 软件产品组成
- 软件测试的模型
- 软件测试的目的和原则
- 软件测试的术语
- 软件测试过程
- 完整的软件测试系统

# 完整的软件测试系统

测试本身包含丰富的内容，为了能够真正高效和有效地进行软件测试，首先要构建一个完整的测试系统，这个测试系统将作为测试的基础

# 什么是软件测试系统

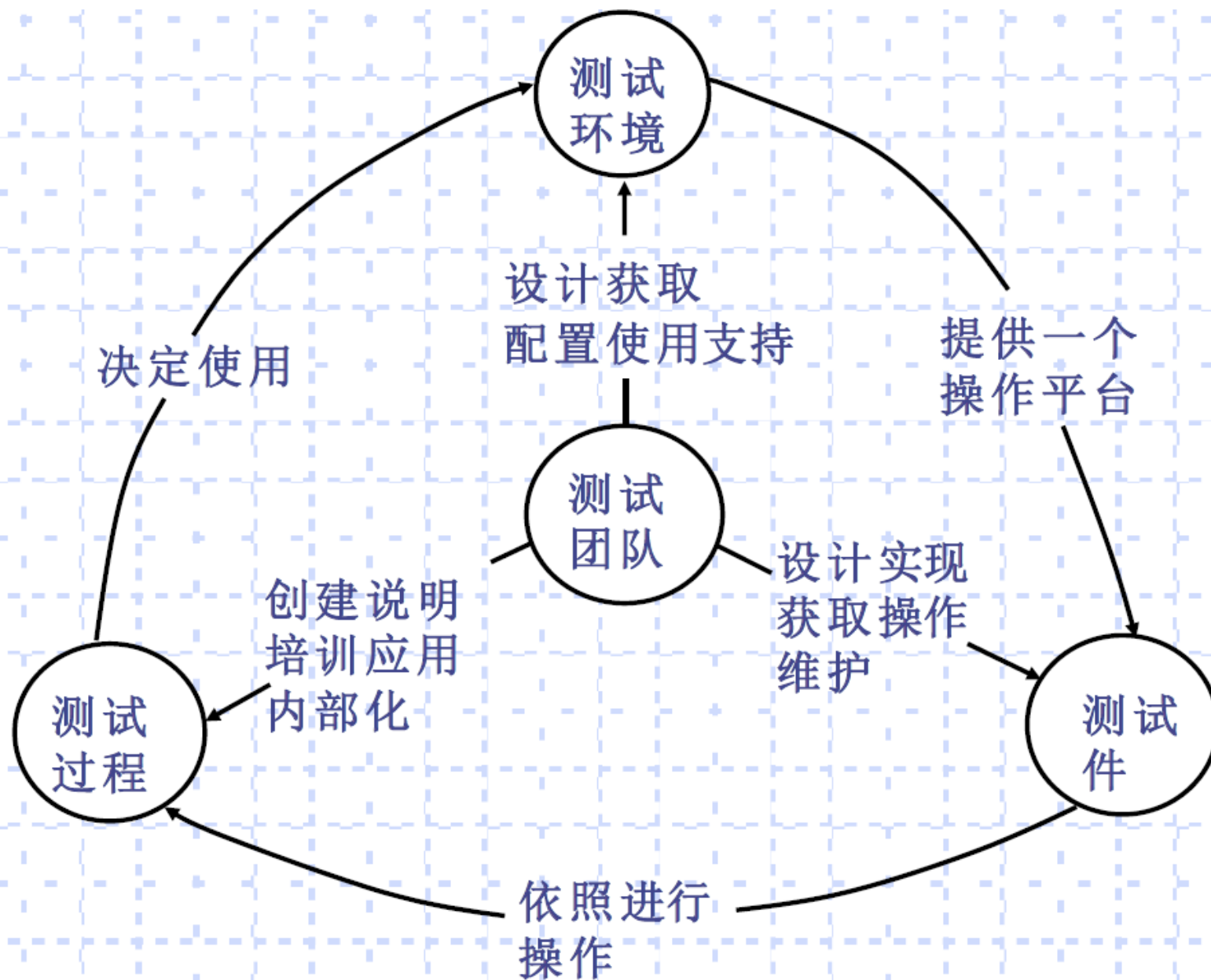
- 软件测试系统（Software testingsystem）是进行软件测试所包含的全部内容
- 包括测试人员、测试环境、测试过程、测试件、测试原则、测试管理、测试策略以及测试技术等

# 软件测试系统的作用

软件测试系统彼此联系完成整个测试任务，其输入是一项测试任务，输出是测试的结果，如测试总结报告等。



# Rex.Black 的软件测试系统



# Rex.Black 测试系统说明

■在Rex. Black的软件测试系统中：测试团队居中，处于核心位置，测试团队根据测试任务设计、创建测试环境、测试件以及测试过程；然后依赖测试环境并按照测试过程操作测试件，完成测试工作，最后得出的测试报告又加入到测试件中

# 测试系统相关术语

## 1. 测试团队(test team)——

进行软件测试工作的相关人员，包括测试经理、测试工程师、测试技术员等，他们分工协作，团结一致，利用自己的测试知识、技术以及经验共同完成测试任务

测试团队处于测试系统的核心位置



# 测试系统相关术语

## 2. 测试件（testware）——

测试件是软件测试中使用的所有人工制品，包括在测试过程中产生的测试计划、测试设计和执行测试以及测试报告所需要的人工制品，例如文档、脚本、输入、预期结果、安装和清理步骤、文件、数据库、环境和任何测试中使用的软件和工具

# 测试系统相关术语

3. 测试环境（test environment）——  
执行测试需要的环境，包括硬件、仪器、模拟器等，软件工具和其他支持要素，比如测试空间
4. 测试过程（test process）——  
完成测试工作的流程，它由一组文档化的指导原则指导进行，比如制定测试计划、设计测试、执行测试以及测试总结等

# 举例

为了测试前面的Hello软件，建立一个测试系统，包括：

1. 测试团队：包含一名测试员
2. 测试环境：放在实验室的一台电脑，当然还包括放电脑的桌子和测试员要坐的椅子
3. 测试件：包括测试用例，测试记录的Excel软件及制作的测试表格
4. 测试过程：执行的简单流程，生成测试用例，执行测试用例，记录结果及分析结果

# 软件测试系统的目的

1. 一个测试系统必须是**功能可用的**，即可以按照要求完成测试任务
2. 一个测试系统必须是**可靠的**，应该可以重复前面的测试，比如自动化测试系统
3. 一个测试系统应该是**规范的**，其测试的结果最大限度地减少了程序的风险
4. 一个测试系统应该是**高效的**，要求能够较快的完成测试任务
5. 一个测试系统应该是**可维护的**，可以根据测试任务的变化而修正自己，以适应新的要求