



# ISTQB初级认证

## 第1章 软件测试基础

作者：郑文强

Email: [zwqwwuy@163.com](mailto:zwqwwuy@163.com)

博客: [http://blog.csdn.net/Wenqiang\\_Zheng](http://blog.csdn.net/Wenqiang_Zheng)

# 声明

---

→ 本课件的开发基于**ISTQB Foundation Level Syllabus (Version 2007)**。

→ 感谢**ISTQB**和大纲作者的努力，对应的大纲可以从 [www.istqb.org](http://www.istqb.org) 下载获得。

→ 本课件为个人开发，只能用于个人学习目的，不能用于任何商业活动。

→ 更多**ISTQB**初级认证资料，参考：  
[http://blog.csdn.net/Wenqiang\\_Zheng/archive/2011/04/09/6311523.aspx](http://blog.csdn.net/Wenqiang_Zheng/archive/2011/04/09/6311523.aspx)

# 课程内容

---

- 1. 为什么测试**
- 2. 什么是测试**
- 3. 测试基本原则**
- 4. 测试基本过程**
- 5. 测试的误区**
- 6. 测试的心理学**

# 为什么测试

---

## ISTQB考试知识点

- ★ 通过具体例子描述软件中的缺陷会以什么样的方式损害个人、环境或者公司利益（**K2**）；
- ★ 区分引起缺陷的根本原因及其影响（**K2**）；
- ★ 通过举例的方式说明为什么需要测试（**K2**）；
- ★ 描述为什么测试是质量保证的一部分，通过举例说明测试是如何来提高软件质量的（**K2**）；
- ★ 理解术语错误、缺陷、故障、失效的概念以及相应的定义（**K1**）；

# 软件系统无处不在

- ★ 网上购物系统
- ★ 银行系统
- ★ 手机
- ★ 卫星通信系统
- ★ 汽车系统
- ★ .....

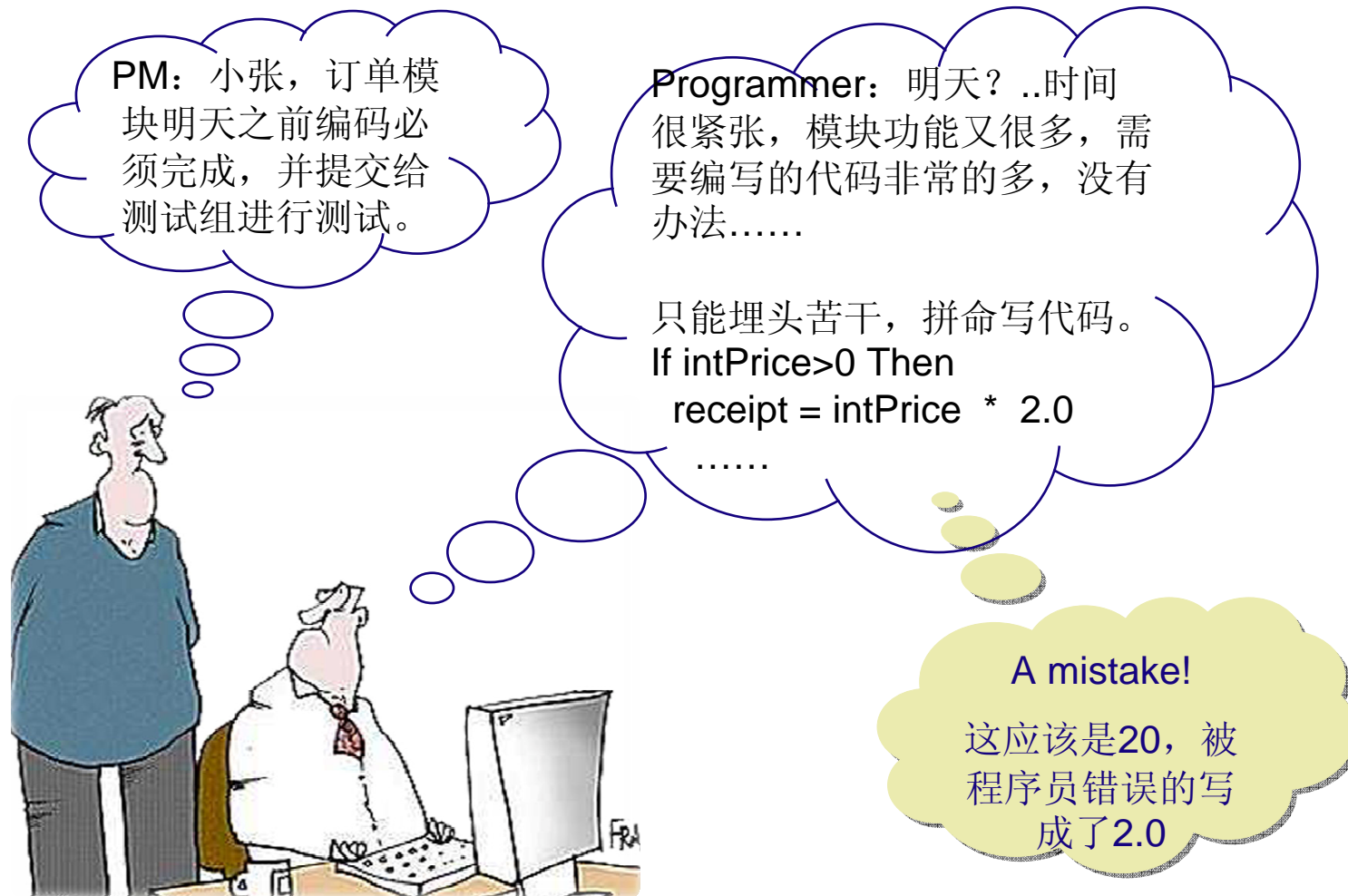
软件系统成为我们生活中不可或缺的一部分.....

我们是否有这样的经验：软件或者系统没有按照预期进行工作？

软件或者系统的不正常运行，导致怎样不同的结果或者后果？

缺陷造成的各种后果无处不在！

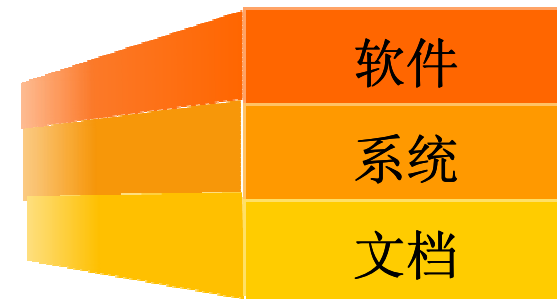
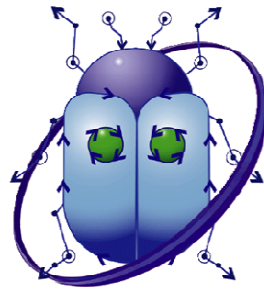
# 为什么会存在缺陷



# 为什么会存在缺陷

## 人的因素

- ★ 人总是容易犯错误的
- ★ 开发人员的态度
- ★ 开发人员的经验和技能不足
- ★ 沟通的问题
- ★ 需求变更频繁
- ★ 管理上的缺失
- ★ 进度压力大
- ★ ...



# 为什么会存在缺陷（续）

---

## 其他因素

- ★ 周围环境的影响：电磁辐射等
- ★ 使用环境的影响：硬件系统的老化等
- ★ 软件系统越来越复杂
- ★ .....



# 缺陷相关术语

---

## 错误

- ★ 与错误相关的英语表示为**ERROR**、**ERRORNEOUS ACTION**、**MISTAKE**等等；
- ★ 定义：人为的产生不正确结果的行为[与**IEEE 610**一致]；
- ★ 人为的原因导致一个不正确的结果。它可以是程序内的内部错误，也可能是文档内的错误；

# 缺陷相关术语

---

## 故障/缺陷

- ★ 与故障/缺陷相关的英语表示为**FAULT**、**DEFECT**、**BUG**等等；
- ★ 定义：可能会导致软件组件或系统无法执行其定义的功能的瑕疵，例如：错误的语句或者变量定义；
- ★ 内在的缺陷是人为错误的具体表现，可以是不正确的文档、程序段、指令或数据定义；

# 缺陷相关术语

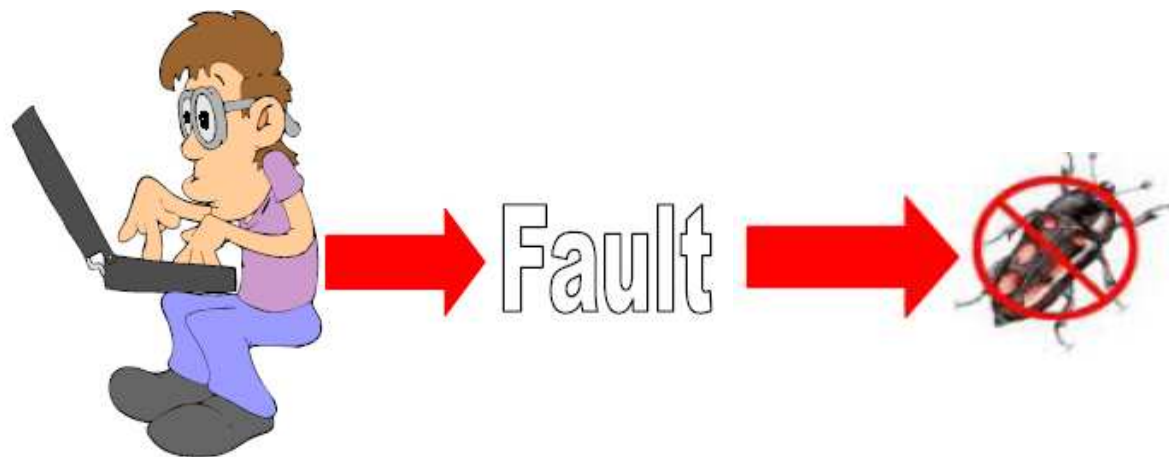
---

## 失效/失败

- ★ 与失效/失败相关的英语表示为**FAILURE**等等;
- ★ 定义：组件/系统与预期的交付、服务或结果存在的偏差[与**Fenton**一致];
- ★ 外部失效是内部缺陷在执行测试软件时的外部反映。它是（规范说明）期望的值与实际（观察到）的值存在偏差，如系统的不正确的反应、崩溃、死机等;

# 术语之间的关系

---



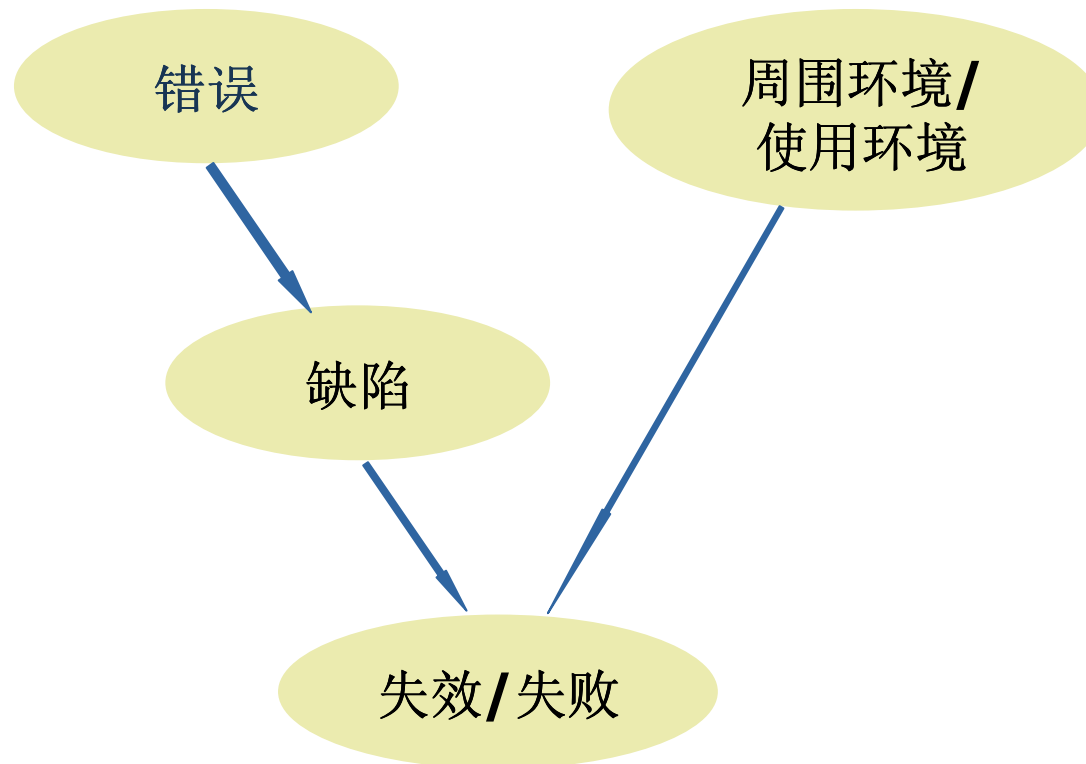
程序人员犯了一个错误  
**(mistake)**

错误**(mistake)**在程序或软件中就表现为缺陷  
**(defect/fault)**

运行带有缺陷**(defect/fault)**的软件或者程序，就可能观察到失效/失败  
**(fail/failure)**

# 术语之间的关系

---



# 术语之间的关系

---

- ★ 缺陷是由于人为错误或者环境因素导致软件系统中存在的瑕疵；
- ★ 运行存在缺陷的软件系统，并不一定会产生运行错误或对用户产生消极的影响，即失效；
- ★ 假如运行存在缺陷的软件系统，发生了错误或者对用户产生了影响，这就是失效；
- ★ 一个缺陷的存在不一定会产生失效，但是有的缺陷可以产生非常多的失效；
- ★ 缺陷屏蔽：一个缺陷会被测试对象中的另外缺陷所掩盖；

# 软件测试的角色

---

- ★ 软件测试是软件质量保证的一个重要环节；
- ★ 软件测试可以降低软件系统在运行环境中的使用风险；
- ★ 系统或者产品通过软件测试发现其中的缺陷，对缺陷修改可以提高软件质量；
- ★ 软件测试也可以为了满足合同或者法律法规的要求，或者满足行业的标准；

# 软件测试的角色（续）

---

- ★ 软件测试可以为项目相关干系人提供信息，帮助他们作出合理的决定；
- ★ 软件测试得到的数据和度量，可以帮助进行测试过程和开发过程的改进；
- ★ 假如测试过程中没有发现或者发现很少的缺陷，可以提高对软件系统的信心；
- ★ .....



# 软件测试和质量保证

---

软件测试：寻找缺陷的策略，关注工作产品

**VS**

质量保证：预防缺陷的策略，关注过程的管理和控制

# 不同的目标

## 质量保证

- ★ 通过监控软件开发过程来保证产品质量；
- ★ 保证软件 and 开发过程符合相应标准与规范；
- ★ 保证软件产品、软件过程中存在的问题得到处理，必要时将问题反映给高级管理者；
- ★ 确保项目组指定的计划、标准和规范适合项目组需要，同时满足评审和审计需要；

## 软件测试

- ★ 尽早、尽可能多地发现软件系统中存在的缺陷及问题；

# 不同的工作内容

## 质量保证

- ★ 建立质量保证活动实体
- ★ 制定质量保证计划
- ★ 坚持各阶段的评审和审计，跟踪结果，并作相应处理
- ★ 监控软件产品质量
- ★ 采集质量保证活动数据
- ★ 度量质量保证活动

## 软件测试

- ★ 编写测试计划
- ★ 评审开发工作产品
- ★ 编写和执行测试用例
- ★ 测试结果分析和总结
- ★ 测试数据收集和度量

# 测试是否足够

---

测试是否足够，依赖于测试目标、产品和项目的风险、项目的限制条件，比如时间、成本和资源等！

# 测试是否足够

---

## ★ 假如测试可以帮助：

- ✦ 减少软件系统运行过程中发生风险的概率；
- ✦ 更好的满足合同或者标准的要求；
- ✦ 评估和增加软件系统的质量，提高用户对软件的信心；

## ★ 或者可以提供信息：

- ✦ 帮助干系人更好的作出决定：发布软件或继续测试；
- ✦ 帮助进行过程改进和缺陷预防；

**我们可以继续进行测试！**

# 测试是否足够

---

## 错误的成本

### ★ 用户的成本

- ◆ 软件产品的缺陷**&**错误（数据丢失，软件的错误反应，硬件设备的损坏）；
- ◆ .....

### ★ 软件开发方的成本

- ◆ 用户的不满意（形象）；
- ◆ 错误更改费用；
- ◆ 对损害的赔偿费用；
- ◆ .....

# 测试是否足够

---

## 测试的成本

- ★ 测试人员
- ★ 测试设备
- ★ 测试环境
- ★ 办公场所
- ★ 测试过程
- ★ .....

# 测试是否足够

---

## 测试成本的影响因素

- ★ 软件开发过程的成熟度
- ★ 需求是否经常变更
- ★ 软件测试过程成熟度
- ★ 测试环境的质量 & 可用性
- ★ 软件质量和软件可测性
- ★ 员工的素质
- ★ 所遵循质量目标
- ★ .....



# 测试是否足够

---

错误的成本



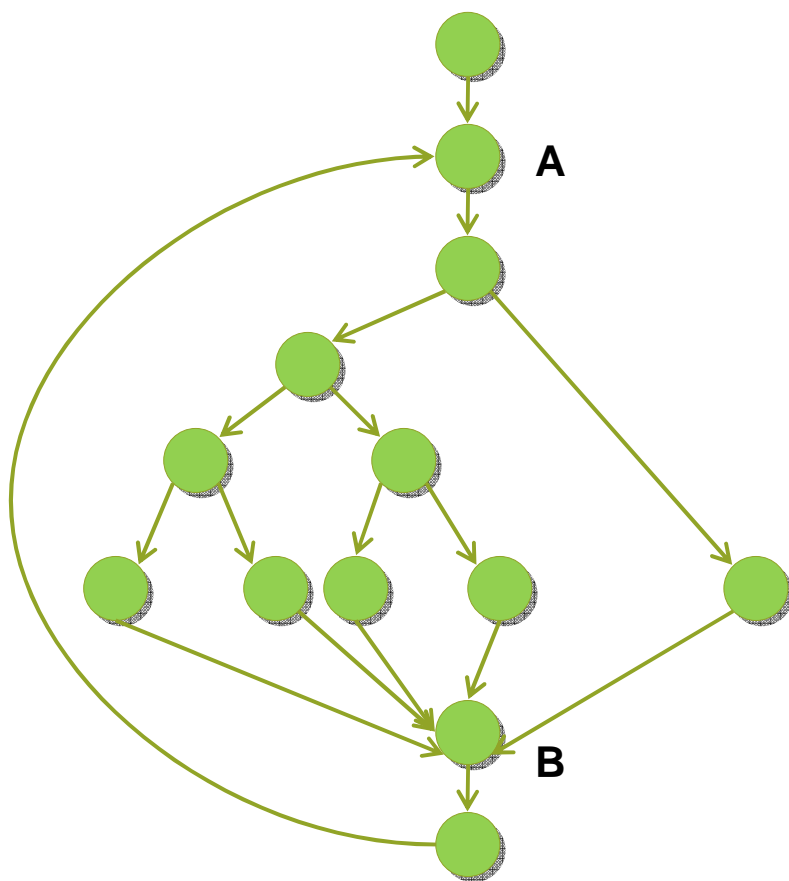
测试的成本



测试的成本必须明显小于错误的成本!

# 多少测试才充分

假设我们需要测试如下的一段程序，我们的目标是测试各条执行路径的情况，执行时从A点开始，直到B点，然后循环回到A点，同时我们假设这段程序被限制循环20次。



$$5^{20} + 5^{19} + 5^{18} + \dots + 5^1 =$$

**119209289550780**

手工测试（3分钟/次）

大约需要**680418319**年  
执行完

自动化测试（3秒钟/次）

大约需要**11340305**年执  
行完

# 多少测试才充分

---

因此，要进行完全的测试，在有限的时间和资源的条件下，找出所有的软件缺陷和错误，使软件产品须趋于完美，是不可能的！

# 多少测试才充分

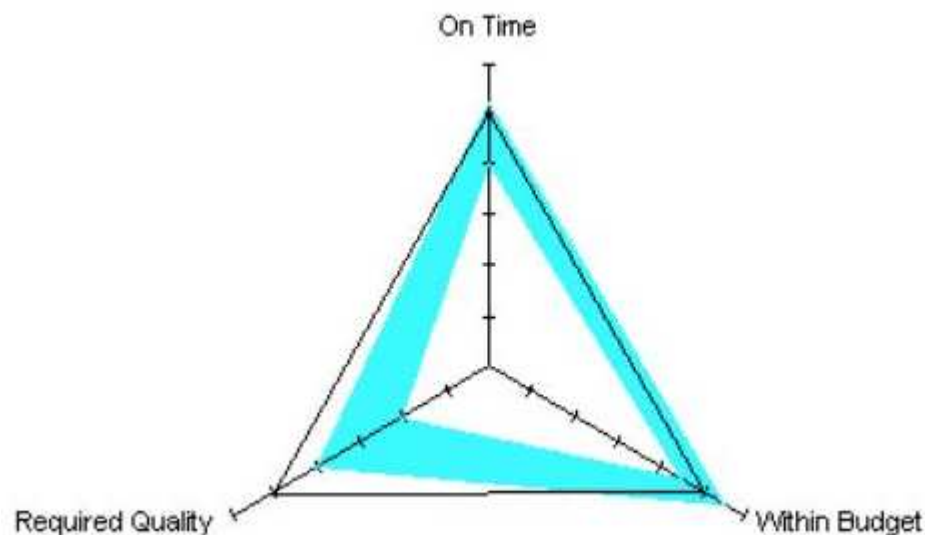
---

## 什么时候结束测试

- ★当测试时间用尽时；
- ★当继续测试没有发现新的失效时；
- ★当继续测试没有发现新的缺陷时；
- ★当无法考虑新的测试用例时；
- ★当测试的回报很小时；
- ★当达到所要求的测试覆盖时；
- ★当所有已发现的缺陷都已经清除时；

# 多少测试才充分

## 什么时候结束测试



什么时候结束测试，需要考虑不同利益干系人在时间、成本和质量三者之间的平衡！

# 课程内容

---

- 1. 为什么测试**
- 2. 什么是测试**
- 3. 测试基本原则**
- 4. 测试基本过程**
- 5. 测试的误区**
- 6. 测试的心理学**

# 什么是测试

---

## ISTQB考试知识点

- ★ 认识测试的总体目标（**K1**）。
- ★ 描述在软件开发、软件维护和软件运行过程中，测试作为发现缺陷、提供信息和信心以及预防缺陷的一种手段（**K2**）。

# 测试的定义

---

## 测试的正向思维



测试就是建立一种信心，确信程序能够按期望的设想进行；

评价一个程序和系统的特性或能力，并确定它是否达到期望的结果。软件测试就是以此为目的的任何行为；

代表人物：Bill Hetzel博士



# 测试的定义

---

## 测试的逆向思维



测试是为发现错误而执行一个程序或者系统的过程；

代表人物：Glenford J. Myers博士

# 测试的定义

---

## 测试的逆向思维（续）

测试是为了证明程序有错，而不是证明程序无错误；

一个好的测试用例是在于它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；

# 测试的定义

---

## IEEE定义的测试



在规定条件下运行系统或构件的过程。观察和记录结果，并对系统或构件的某些方面给出评价；

分析软件项目的过程。检测现有状况和所需状况之间的不同，并评估软件项目的特性；

# 测试的定义

---

## 狭义的测试

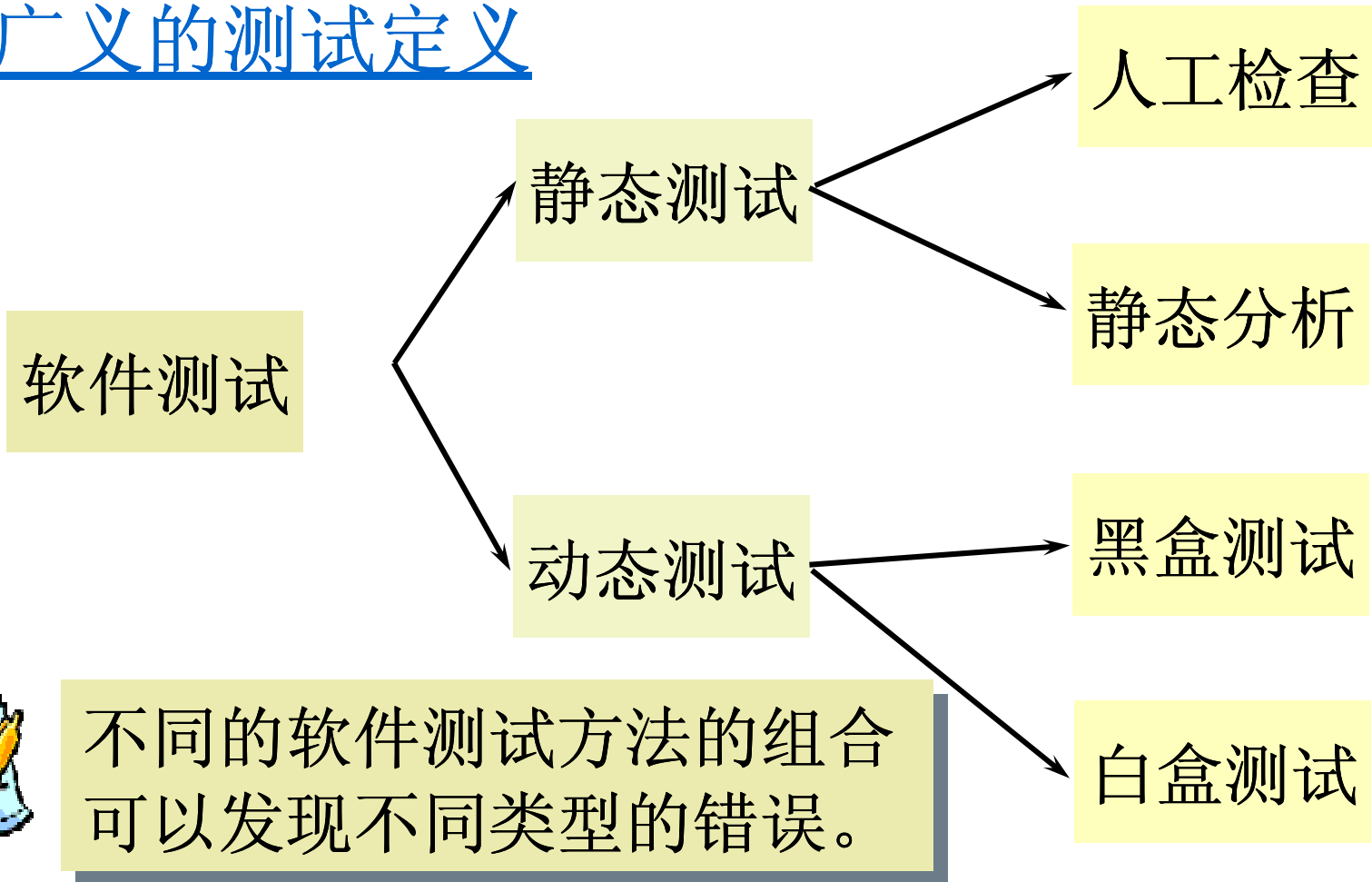
- ★ 一般狭义的测试是运行软件系统进行测试的过程，即执行测试的过程；

## 广义的测试

- ★ 测试活动贯穿于整个软件开发生命周期；
- ★ 在测试执行之前之后都有相应的测试活动，比如测试计划、设计、实现、评估和报告等等；
- ★ 测试包含了动态测试和静态测试；

# 测试的定义

## 广义的测试定义



# 测试的目的

---

- ★ 软件测试可以降低软件系统在运行环境中的风险；
- ★ 通过软件测试发现其中的缺陷，并进行修改，提高软件系统的质量；
- ★ 软件测试也可以为了满足合同或者法律法规的要求，或者满足行业的标准；

# 测试的目的

---

- ★ 软件测试可以为项目相关干系人提供信息，帮助他们作出合理的决定；
- ★ 软件测试得到的数据和度量，可以帮助进行测试过程和开发过程的改进（预防缺陷）；
- ★ 假如测试过程中没有发现或者发现很少的缺陷，可以提高对软件系统的信心；
- ★ .....

# 测试的目的

---

测试的目的依赖于测试级别、测试视角和测试参与人员！

- ★ 单元测试和集成测试的主要目的是什么？
- ★ 系统测试的主要目的是什么？
- ★ 验收测试的主要目的是什么？
- ★ 维护测试、回归测试的主要目的是什么？
- ★ 运行测试的主要目的是什么？
- ★ 入网测试的主要目的是什么？



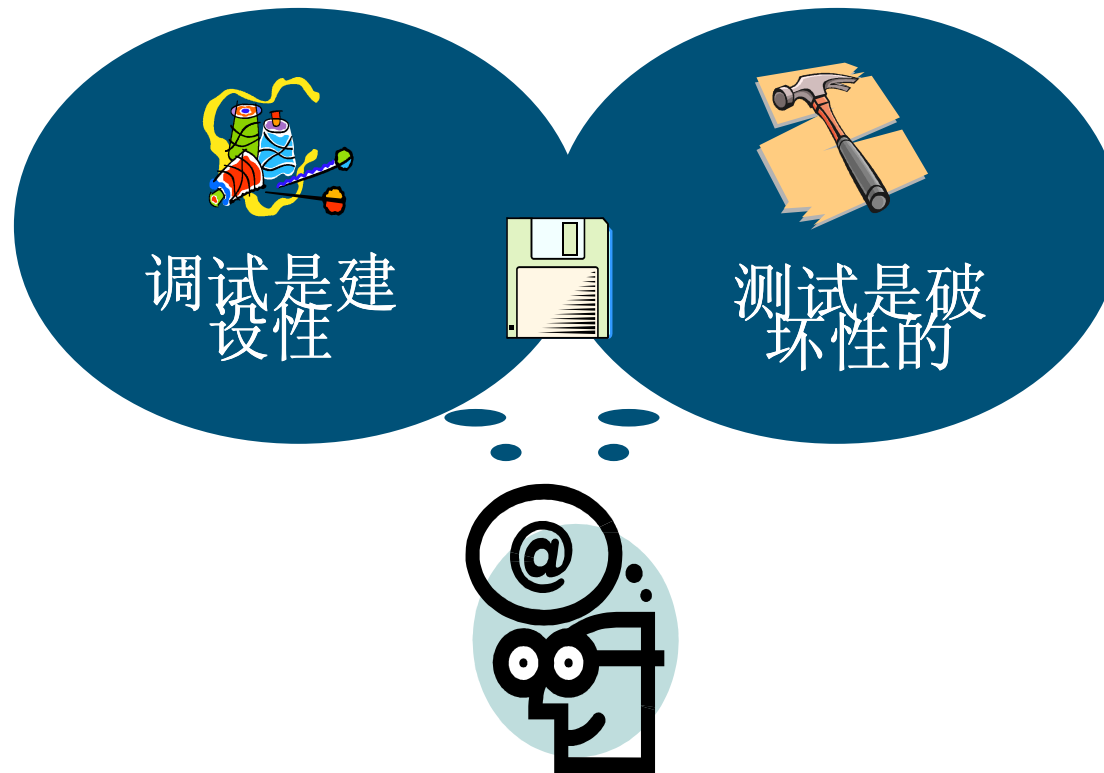
# 测试的目的

---

针对任何级别的软件测试，其测试目的都不是唯一的，通常是前面这些测试目的的组合，或者说这些测试目的是相辅相成的！

# 调试和测试

---



# 测试和调试

---

- ★ **人员的不同**：通常来说，测试人员进行测试活动，开发人员进行调试活动（当然，开发人员也会进行一些测试活动，比如，单元测试通常由开发人员来进行）；
- ★ **目标的不同**：测试和调试在目标、方法和思路上有所不同。比如，测试的目的之一是发现软件中的缺陷，而调试的主要目的通常是为了定位和修改软件中的缺陷；
- ★ **过程的不同**：测试是从已知的条件开始，使用预先定义的过程，并且有预知的结果；调试是从未知的条件开始，结束的过程可能不可预计；

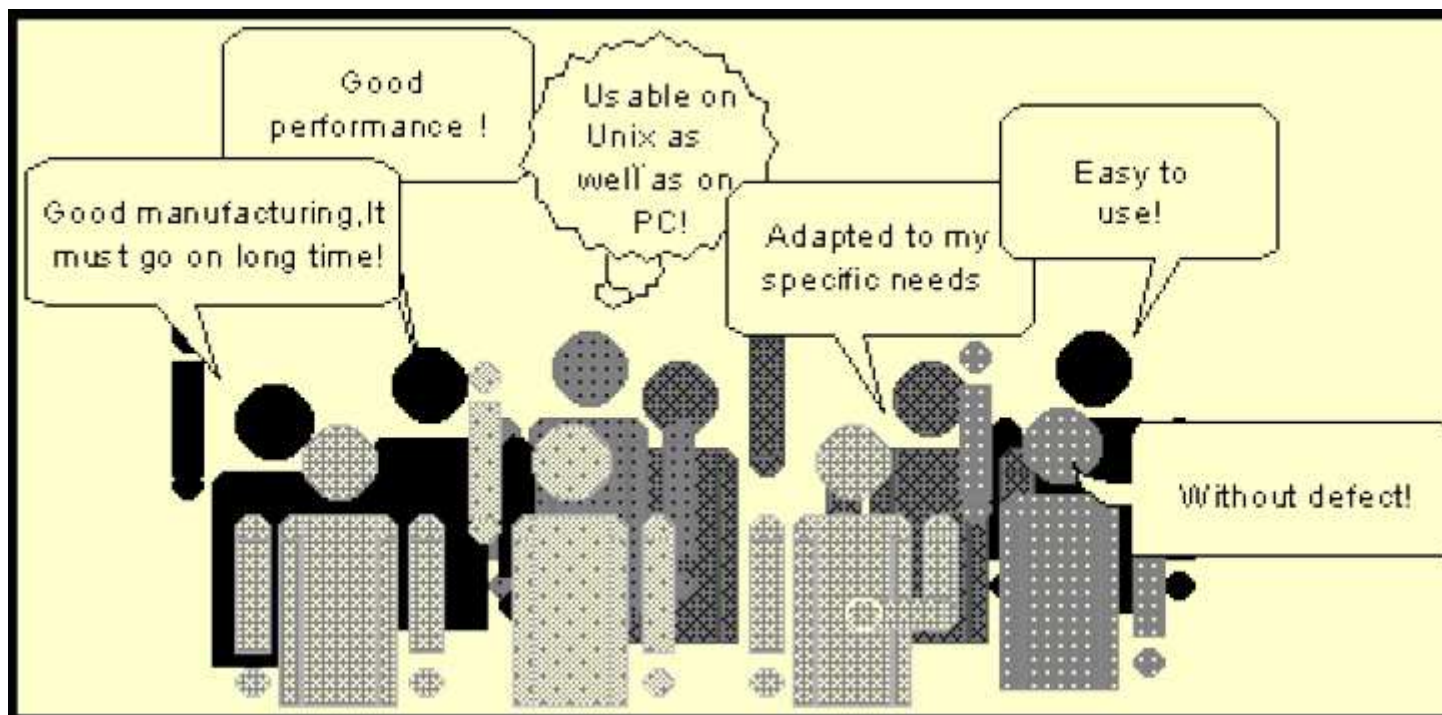
# 测试和调试

---

- ★ **计划的不同**：测试可以计划，可以预先制定测试用例和过程，工作进度可以度量；描述调试的过程或持续时间相对比较困难；
- ★ **对象的不同**：测试的对象包括软件开发过程中的文档、数据以及代码，而调试的对象一般来说只是代码；
- ★ .....

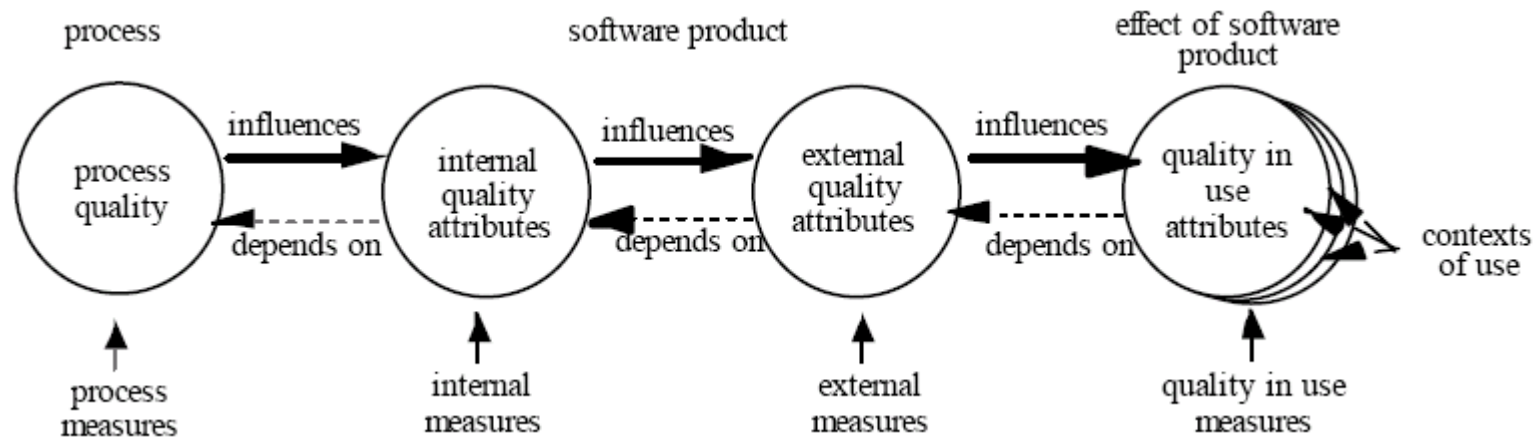
# 软件测试和质量

## 软件质量的不同含义



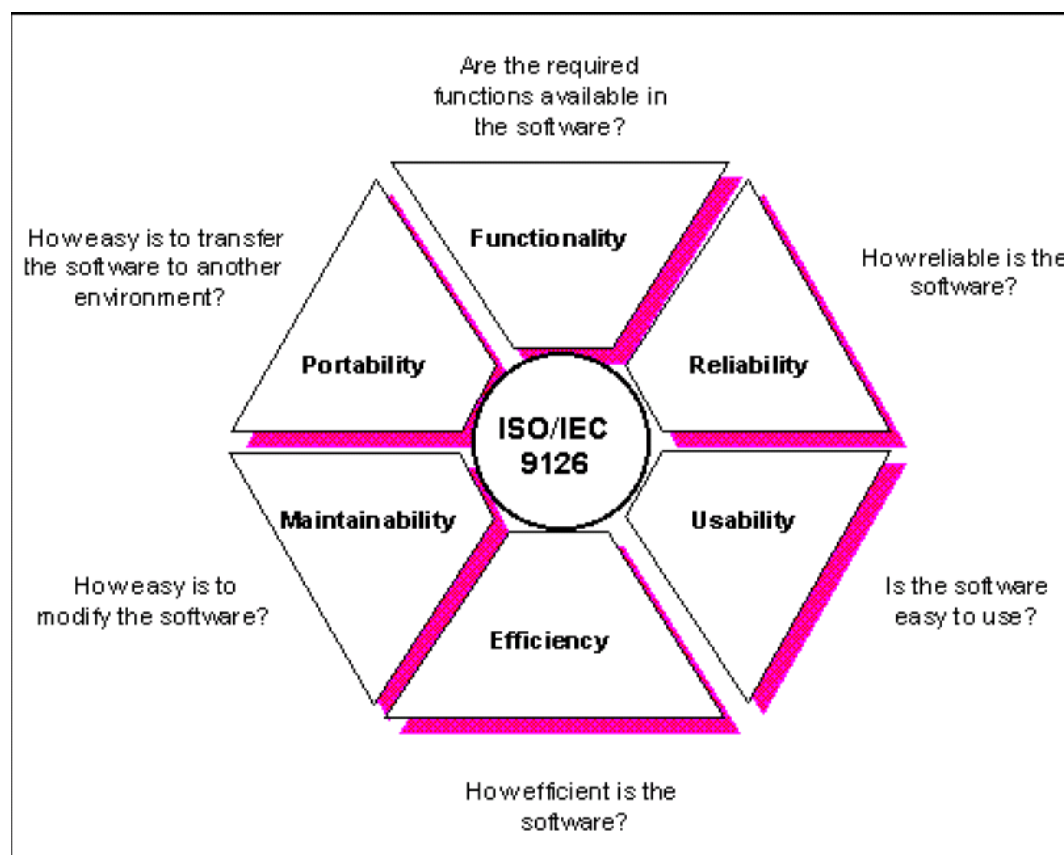
# 软件测试和质量

## 软件质量的不同含义



# 软件测试和质量

## 软件质量模型



# 测试工作量

---

## 穷尽测试是不可能的

```
proc sum()  
  ineger i  
  integer j  
  integer k  
  integer sum  
  
  for (i=1 upto 10000)  
    for (j=1 upto 10000)  
      for (k=1 upto 10000)  
  
        sum=i+j+k  
  
      endfor  
    endfor  
  endfor  
endproc
```



# 测试工作量

---

## 测试工作量的分配

- ★ 测试工作量占整个研发的**25%**到**50%**之间；
- ★ 测试人员和开发人员之间的比例：从**1:10**到**3:1**；
- ★ 系统失效导致的高风险；
- ★ 如何平衡时间、成本、资源和质量之间的关系？

# 测试工作量

---

穷尽测试不可能，因此，如何运用有限的测试资源就非常重要！

为了得到让人满意的结果，测试必须以结构化、系统化的方式设计和执行。这样才能在有限的测试资源和工作量下，找到尽可能多的缺陷，并且避免执行那些找不到较多缺陷或者不能获得较多系统质量信息的不必要的测试！

# 课程内容

---

1. 为什么测试
2. 什么是测试
3. 测试基本原则
4. 测试基本过程
5. 测试的误区
6. 测试的心理学

# 测试基本原则

---

## ISTQB考试知识点

★ 说明测试的基本原则（**K2**）。

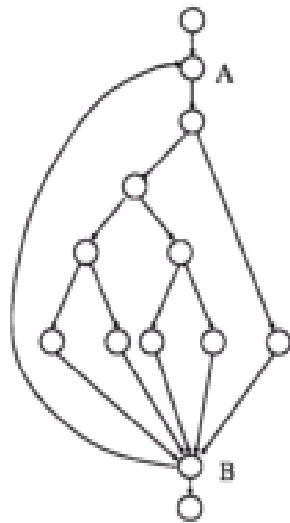
# 测试基本原则

---

- ★ **原则1**：测试可以显示缺陷的存在；
  - ✦ 通过测试，可以减少测试对象中存在未被发现缺陷的可能性；
  - ✦ 但即使在测试过程中没有发现缺陷，也不能证明测试对象是完全正确的；

# 测试基本原则

★ 原则2：穷尽测试是不可能的；



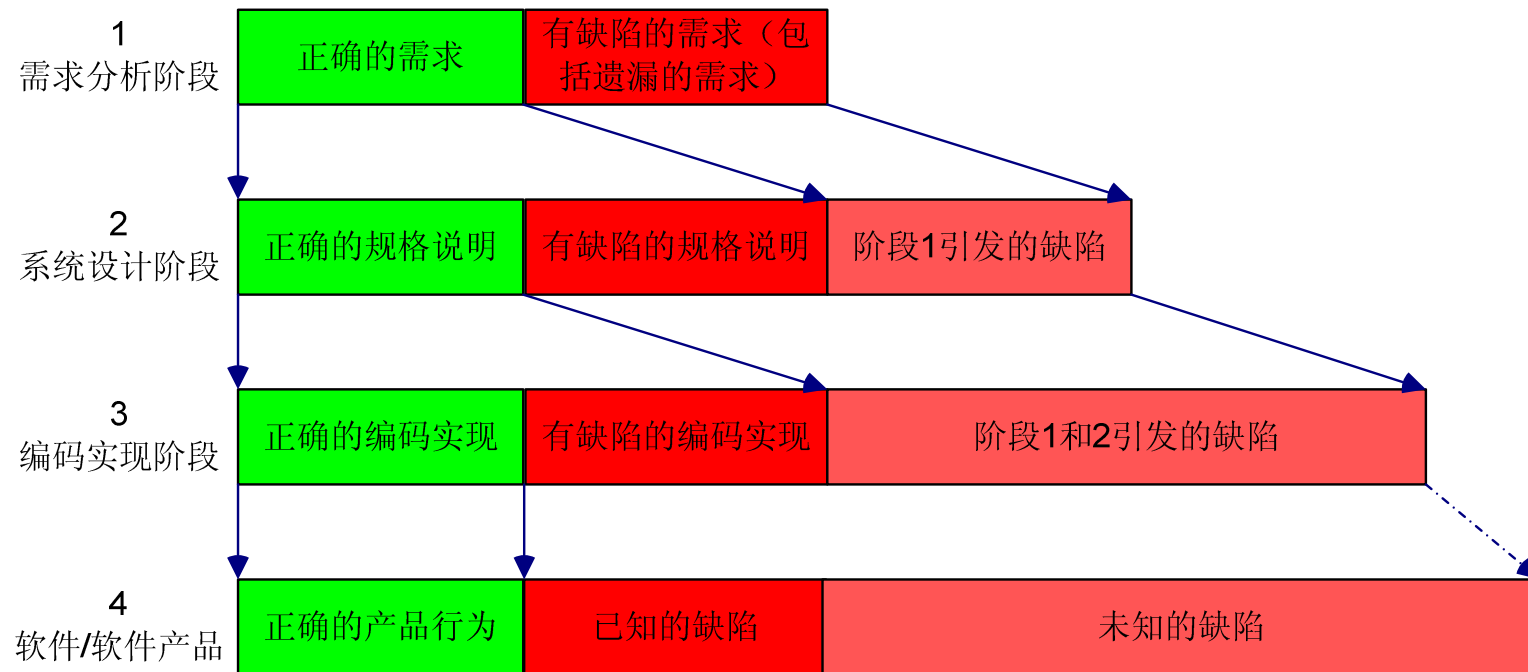
点 A 和点 B 之间是一个循环，有一个从点 B 到点 A 的返回。假设循环的上限是**20**次并且所有的连接都是独立的。假如我们要进行完全测试，需要进行的测试次数是：

$$5^{20} + 5^{19} + 5^{18} + \dots + 5^1$$

这是一个天文数字！

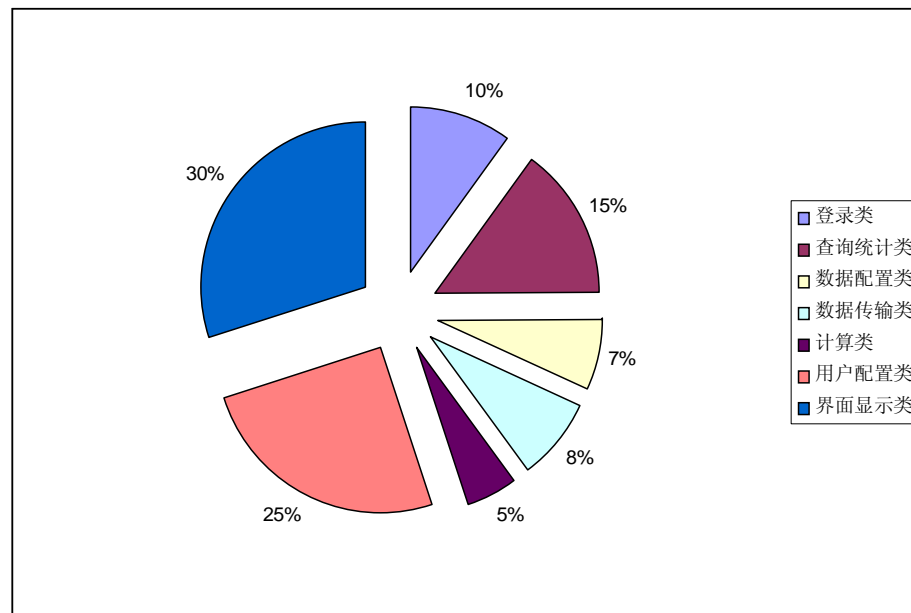
# 测试基本原则

## ★ 原则3：测试的尽早介入；



# 测试基本原则

★ **原则4**：缺陷的集群效应，即测试对象中发现的**80%**的缺陷来自**20%**的模块；





# 测试基本原则

---

- ★ **原则5**：杀虫剂的悖论，即同样的测试用例被重复使用多次，将不能发现新的缺陷；
- ★ **原则6**：测试活动依赖于测试对象的背景，比如安全性相关的测试对象和一般的商业对象，测试活动是不完全一样的；
- ★ **原则7**：不存在缺陷并不代表是有用的系统；

# 课程内容

---

- 1. 为什么测试**
- 2. 什么是测试**
- 3. 测试基本原则**
- 4. 测试基本过程**
- 5. 测试的误区**
- 6. 测试的心理学**

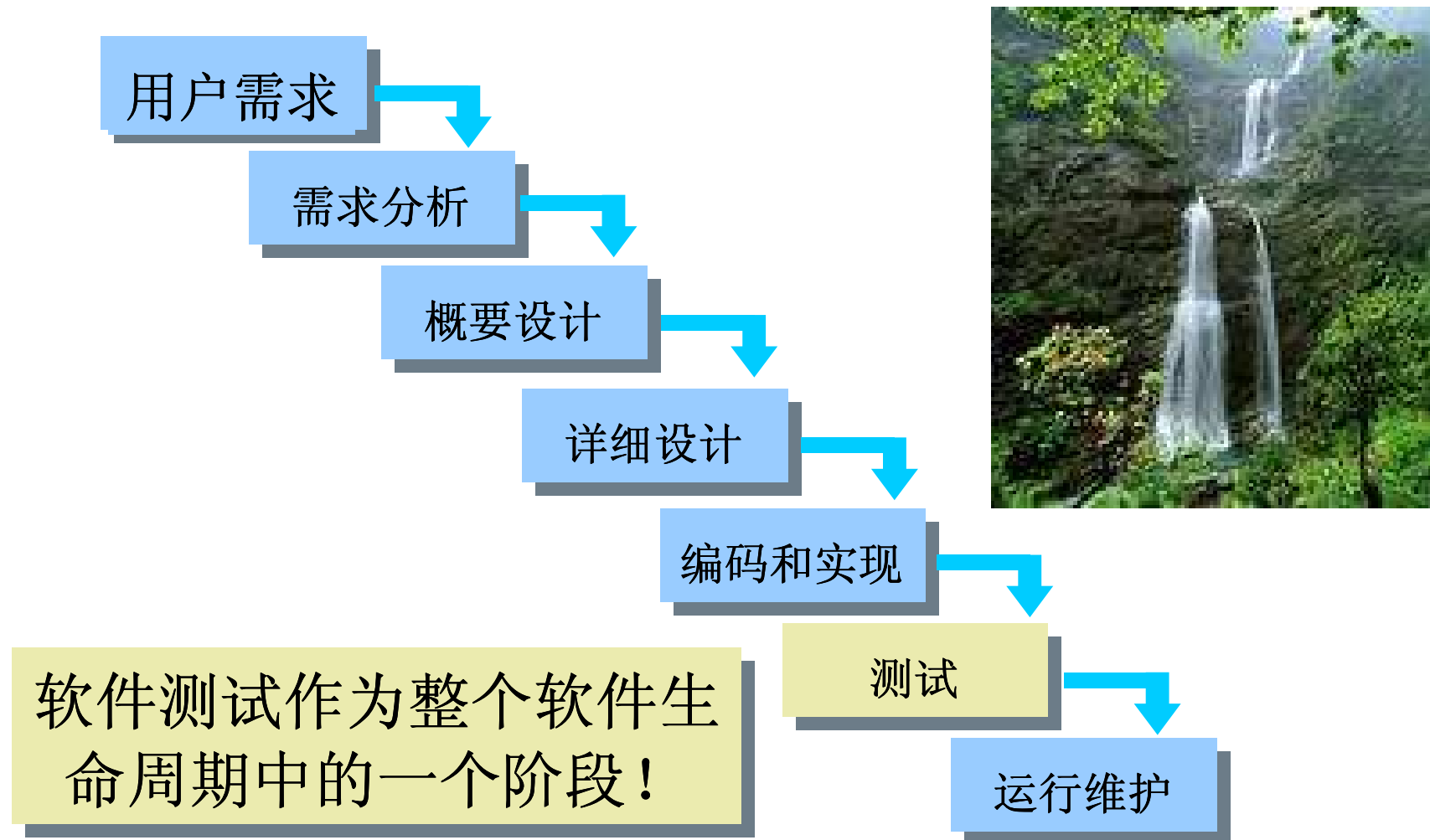
# 测试基本过程

---

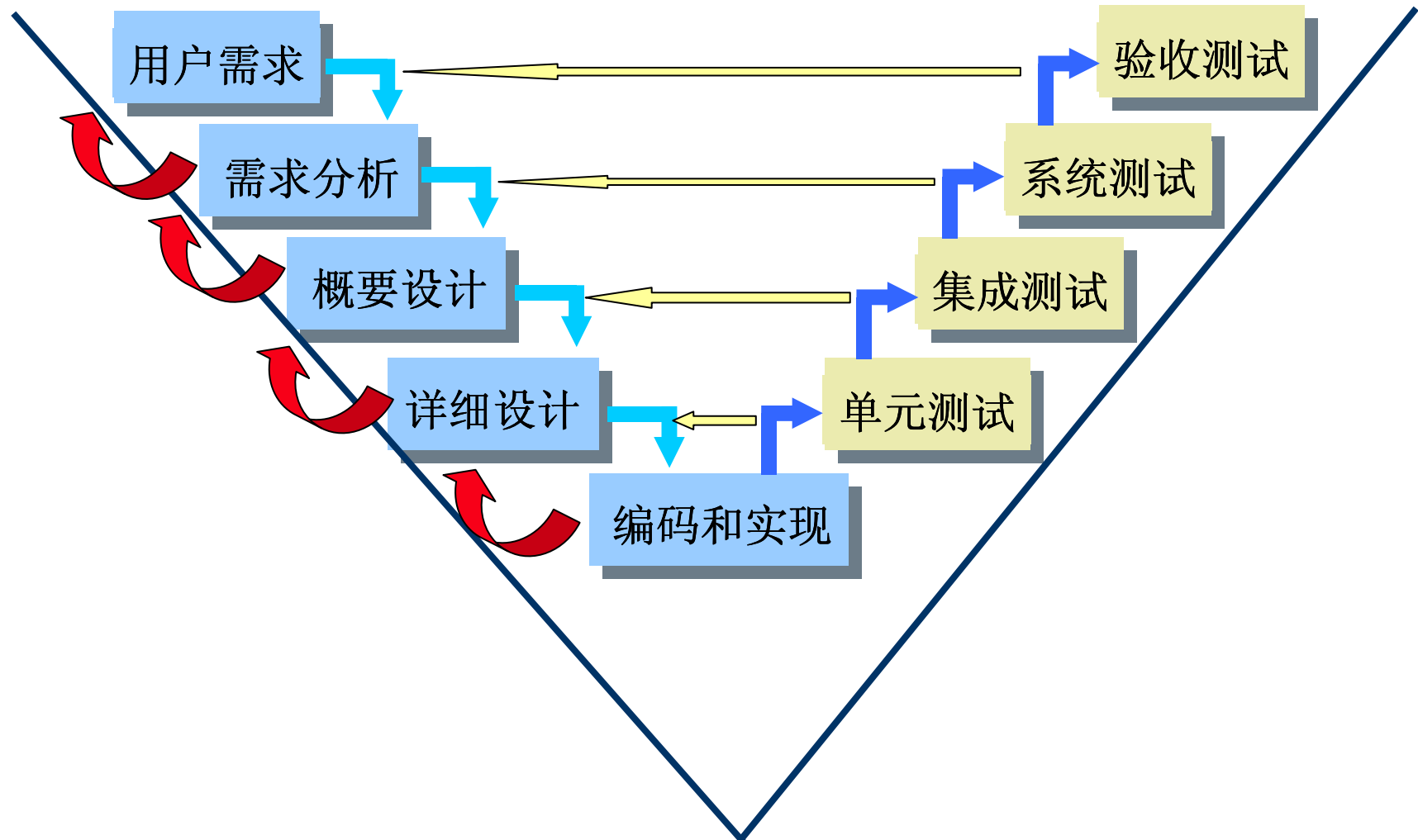
## ISTQB考试知识点

- ★ 认识从计划到测试结束过程中测试的基本活动，以及在每个测试活动中的主要任务（**K1**）。

# 测试基本过程



# 测试基本过程



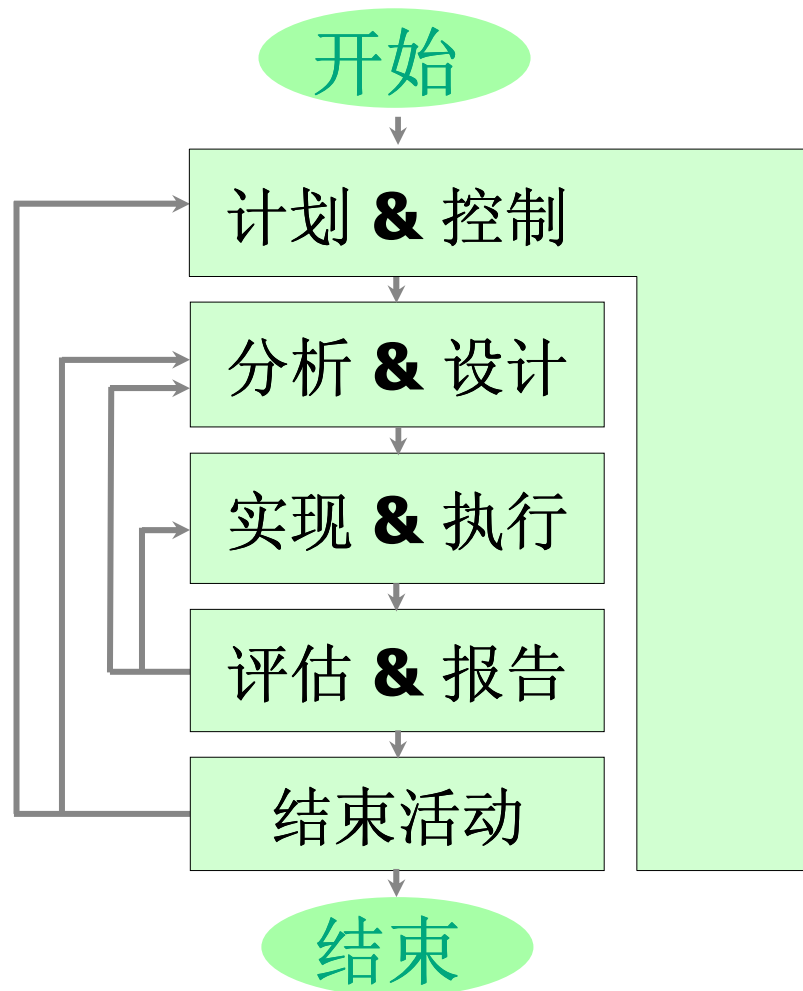
# 测试基本过程

---

## 结构化的软件测试过程

- ★ 结构化的软件测试，使软件测试在整个软件开发过程中可以对软件进行系统的、结构化的测试；
- ★ 结构化的软件测试过程包括：
  - ◆ 计划和控制
  - ◆ 分析和设计
  - ◆ 实现和执行
  - ◆ 出口准则评估和报告
  - ◆ 测试结束活动

# 测试基本过程



软件测试的这些阶段，  
尽管看起来是按照顺序  
进行的，但它们经常是  
重叠的！

# 测试基本过程

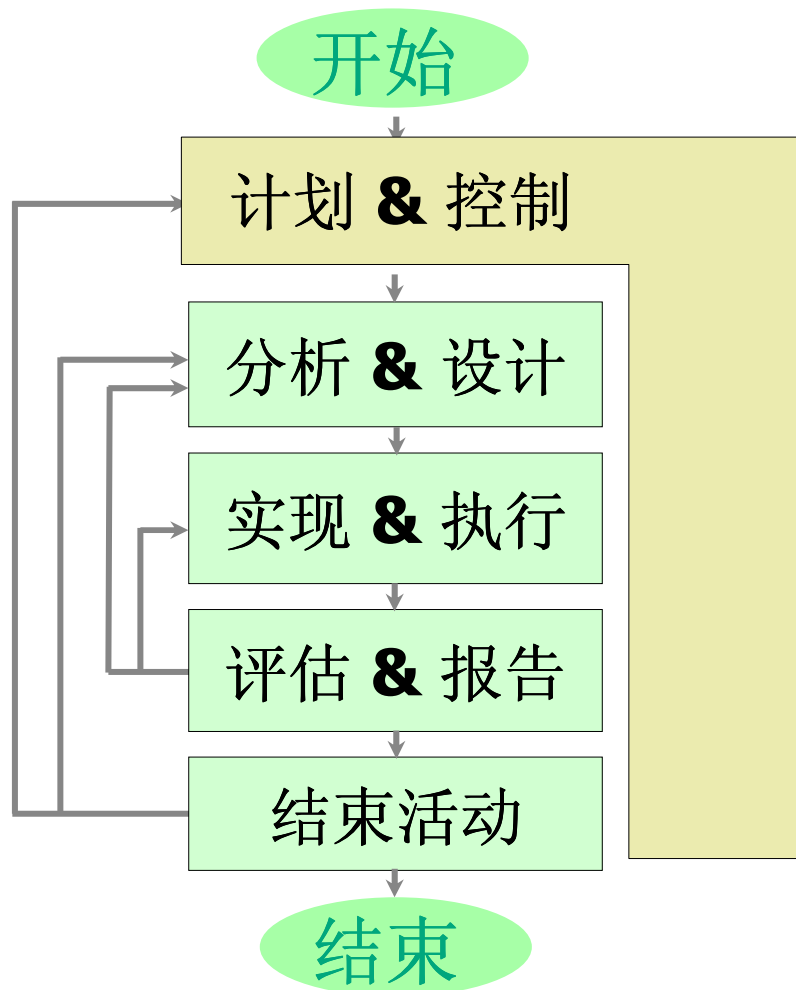
---

主要文档输出：

- ✦ 测试计划/测试进度
- ✦ 概要测试规格说明
- ✦ 详细测试规格说明/测试脚本/测试环境
- ✦ 测试日志/缺陷报告/测试报告
- ✦ 测试总结报告



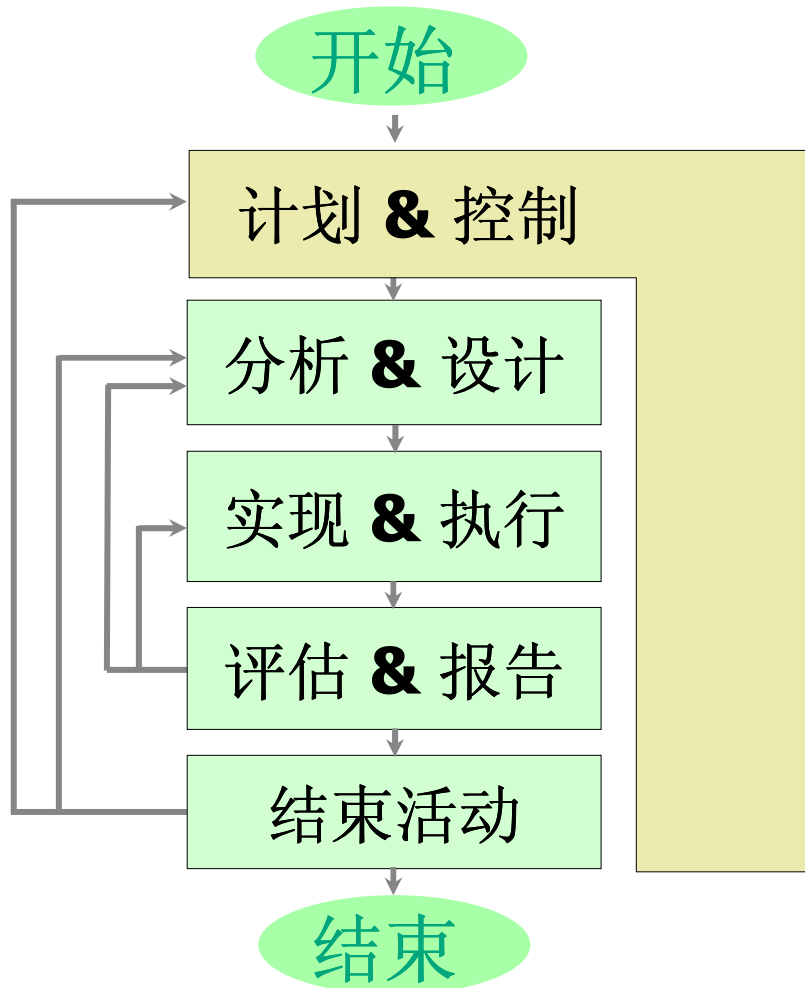
# 测试计划和控制



## 测试计划活动：

- ◆ 识别测试任务和测试范围；
- ◆ 确定测试方法和测试技术；
- ◆ 确定测试级别和测试准则；
- ◆ 确定测试工作量和测试风险；
- ◆ 确定测试资源；
- ◆ 确定测试工作产品和输出；
- ◆ .....

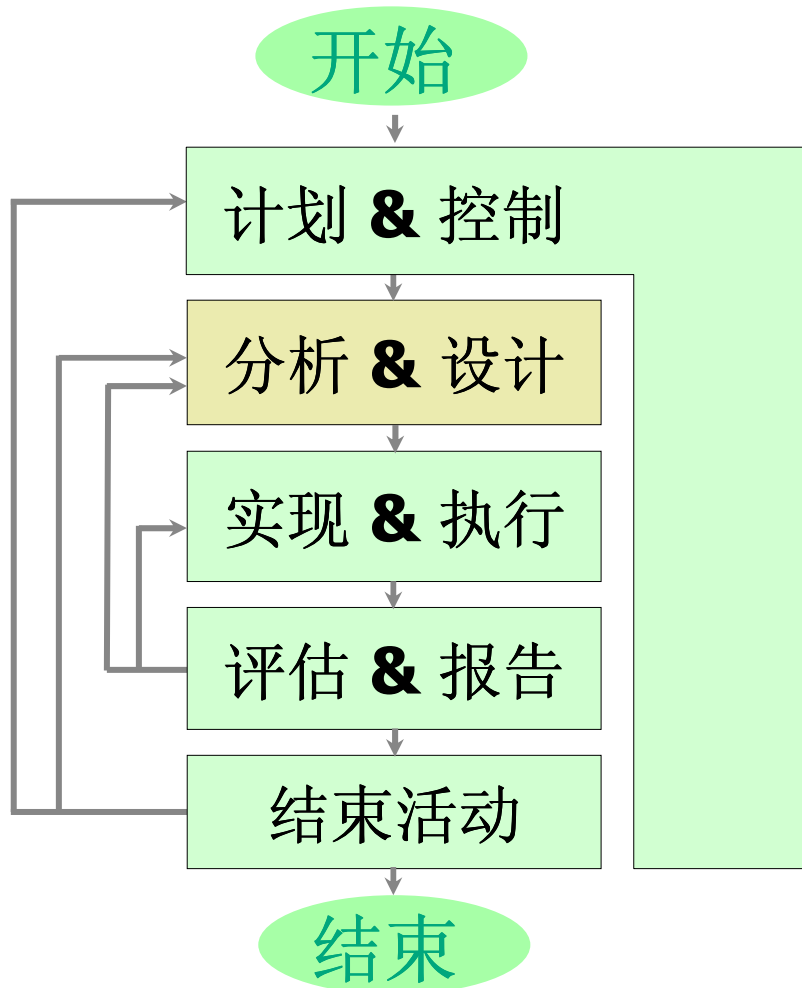
# 测试计划和控制



## 测试控制活动:

- ✦ 监控和记录测试进度、测试覆盖率以及出口准则评估;
- ✦ 测量和分析测试结果;
- ✦ 缺陷跟踪和回归测试计划;
- ✦ 跟踪和监控工作量分布和风险状态;
- ✦ 更新测试计划;
- ✦ .....

# 测试分析和设计



## 测试分析和设计活动：

- ✦ 评审测试依据，比如需求、设计文档；
- ✦ 分析和确定测试优先级；
- ✦ 识别测试条件、测试需求、测试数据；
- ✦ 设计概要测试用例并确定优先级；
- ✦ 规划测试环境搭建，计划测试基础设施和工具；
- ✦ .....

# 测试分析和设计：案例

某公司计划在圣诞节时给员工发放奖金，公司规定凡在公司工作3年以上（不含3年）5年（含5年）以下的员工，可得到本月工资的50%做为奖金；凡工作5年以上8年（含8年）以下的员工，可得到本月工资的75%做为奖金；凡工作8年以上者，可得到本月工资的100%做为奖金。

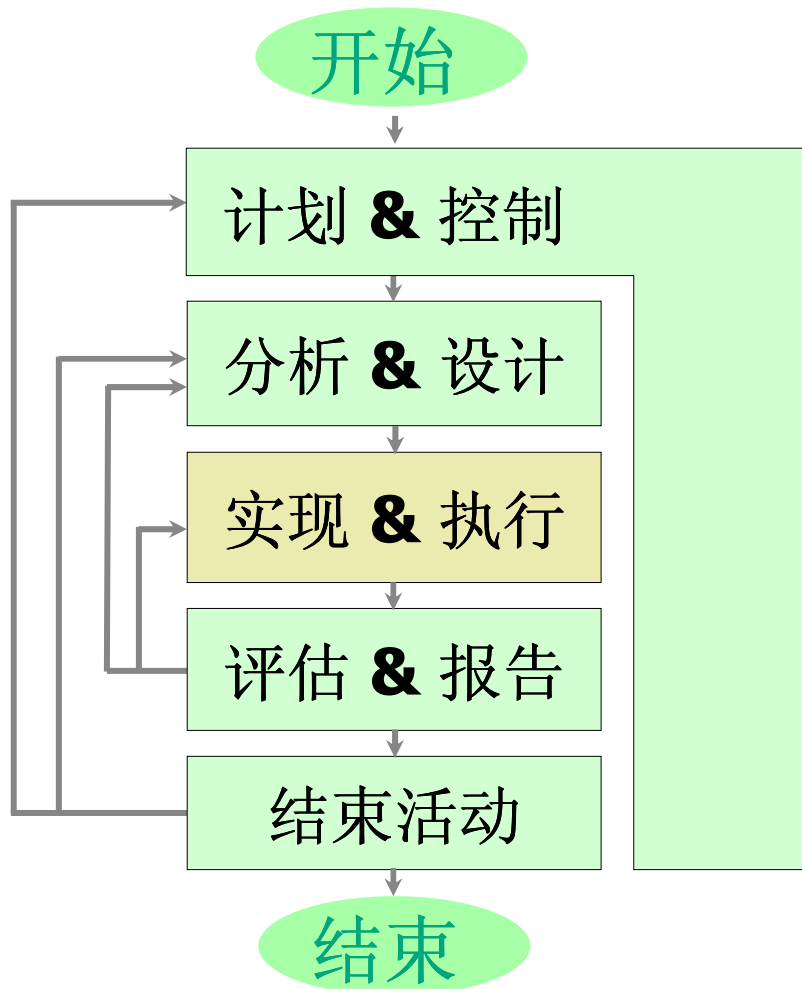
概要测试用例

测试用例编号	输入	期望结果
<b>1</b>	<b><math>X \leq 3</math></b>	<b>0</b>
<b>2</b>	<b><math>3 &lt; X \leq 5</math></b>	<b>50</b>
<b>3</b>	<b><math>5 &lt; X \leq 8</math></b>	<b>75</b>
<b>4</b>	<b><math>X &gt; 8</math></b>	<b>100</b>

详细测试用例

测试用例编号	输入	期望结果
<b>1</b>	<b>2</b>	<b>0</b>
<b>2</b>	<b>4</b>	<b>50</b>
<b>3</b>	<b>7</b>	<b>75</b>
<b>4</b>	<b>12</b>	<b>100</b>

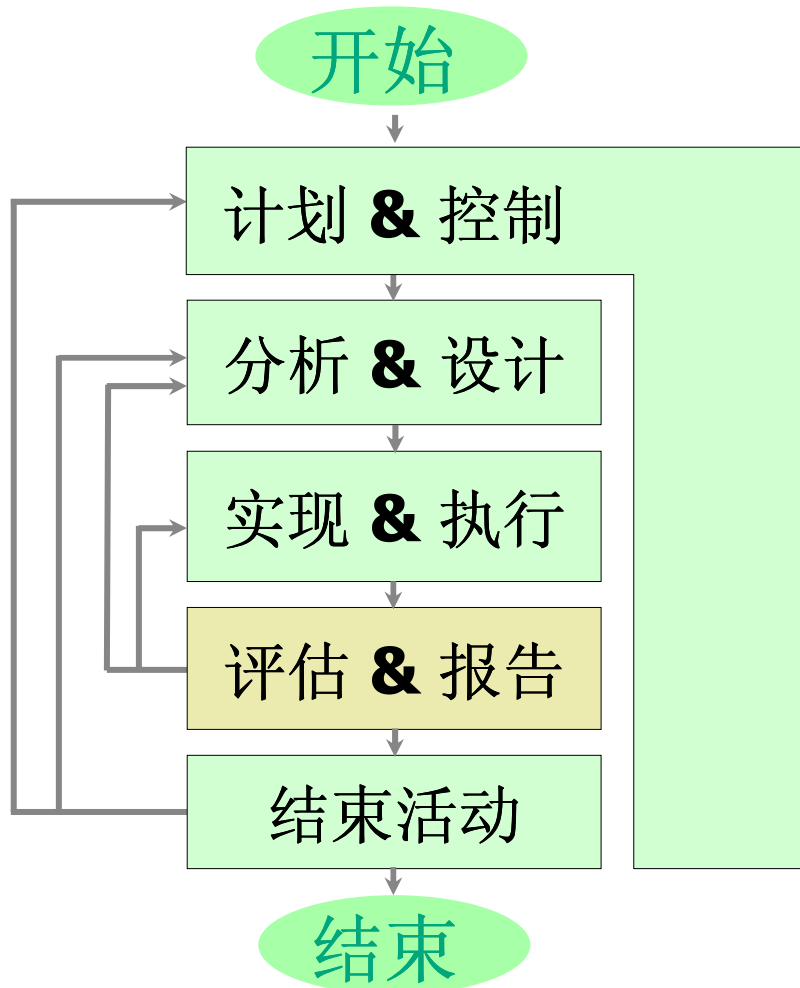
# 测试实现和执行



## 测试实现和执行活动：

- ✦ 详细测试用例的开发、实现和维护；
- ✦ 创建测试数据，准备测试用具和开发测试脚本；
- ✦ 搭建测试环境；
- ✦ 执行测试用例，记录测试结果，提交缺陷报告；
- ✦ 缺陷验证测试和回归测试；
- ✦ .....

# 出口准则评估和报告



## 评估和报告活动:

- ◆ 根据测试出口准则检查测试日志;
- ◆ 根据检查结果, 判断是否需要更多的测试, 或者需要修改测试出口准则;
- ◆ 为项目利益干系人提供测试报告;
- ◆ .....

# 出口准则评估和报告

## 出口准则例子

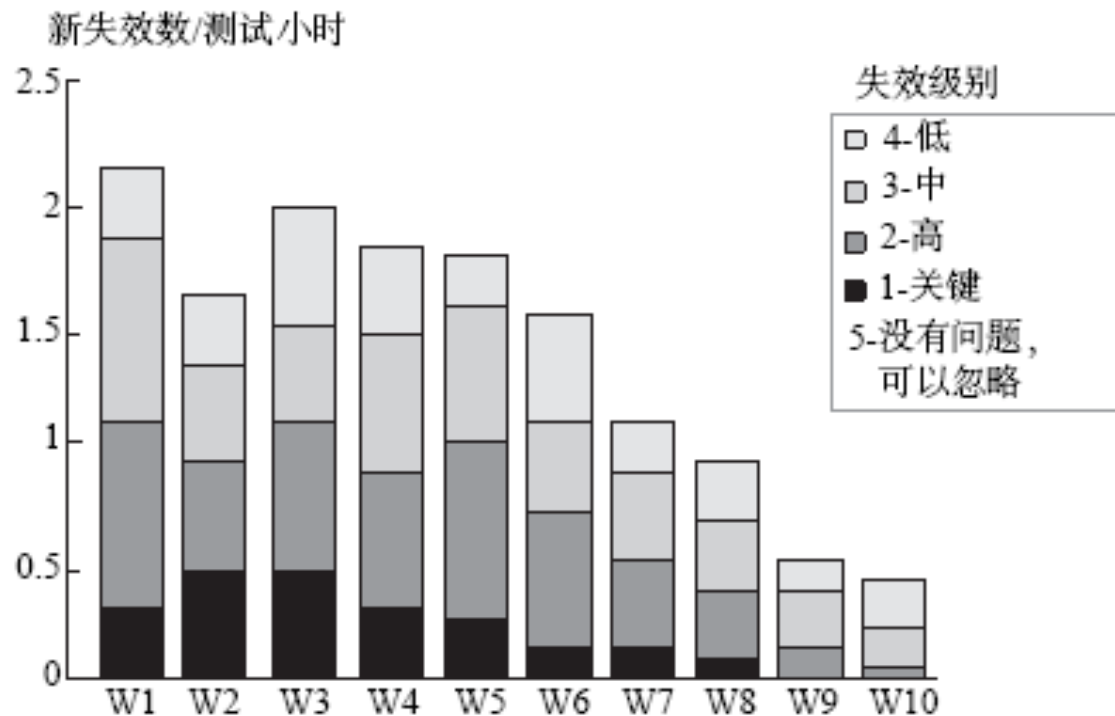
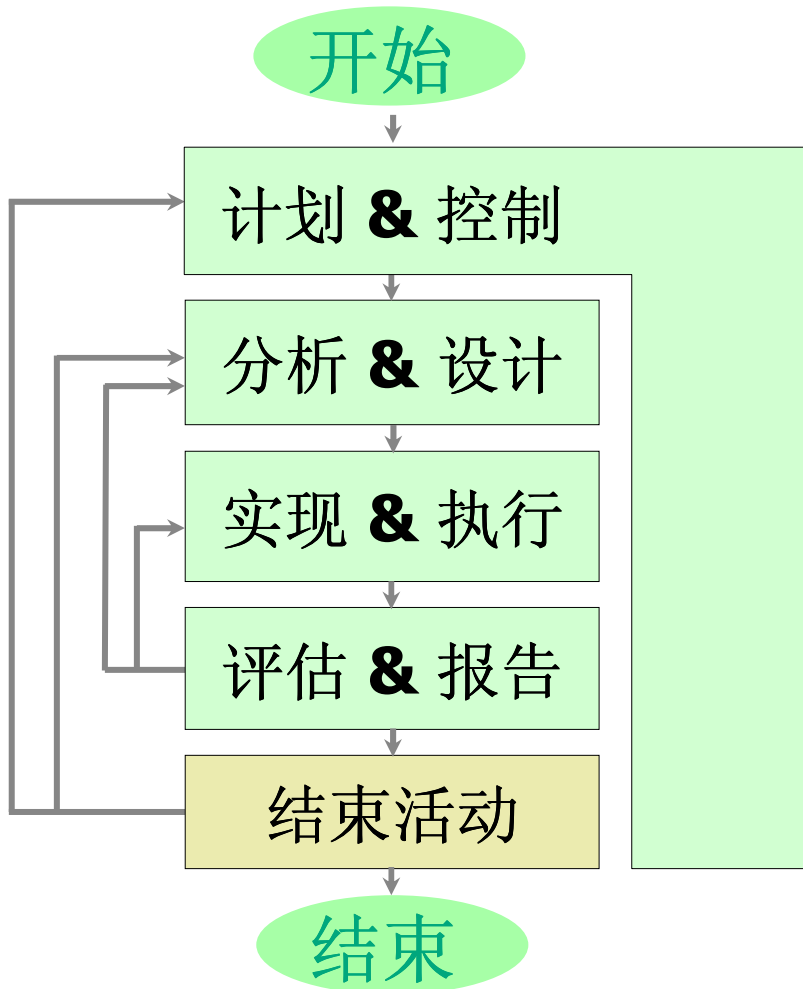


图2-5 失效率

# 测试结束活动



## 测试结束活动:

- ✦ 检查测试工作产品是否正常提交;
- ✦ 提交的缺陷是否关闭;
- ✦ 提交的变更记录是否关闭;
- ✦ 测试件、测试环境和测试基础设施的归档和移交;
- ✦ 分析和总结经验教训, 输出测试总结报告;
- ✦ .....



# 课程内容

---

- 1. 为什么测试**
- 2. 什么是测试**
- 3. 测试基本原则**
- 4. 测试基本过程**
- 5. 测试的误区**
- 6. 测试的心理学**

# 测试的误区

---

## **ISTQB**考试知识点

★无。

# 测试的误区

---

- ★ 测试活动是整个开发生命周期中的一个阶段，测试活动在开发活动之后；
- ★ 软件质量是有测试人员决定的，假如在客户中发现缺陷，那是测试人员的责任；
- ★ 测试人员不需要具备很高的技能，能力不佳的人才去做测试人员；

# 测试的误区

---

- ★ 测试工作是测试人员的事情，和开发人员无关；
- ★ 测试工作的多少，或者工作量，是由项目的进度来决定的，时间多就多做一点，少就少做一点；
- ★ 软件测试工作是没有前途的；
- ★ 自动化测试将会取代手动测试；
- ★ .....

# 课程内容

---

- 1. 为什么测试**
- 2. 什么是测试**
- 3. 测试基本原则**
- 4. 测试基本过程**
- 5. 测试的误区**
- 6. 测试的心理学**

# 测试的心理学

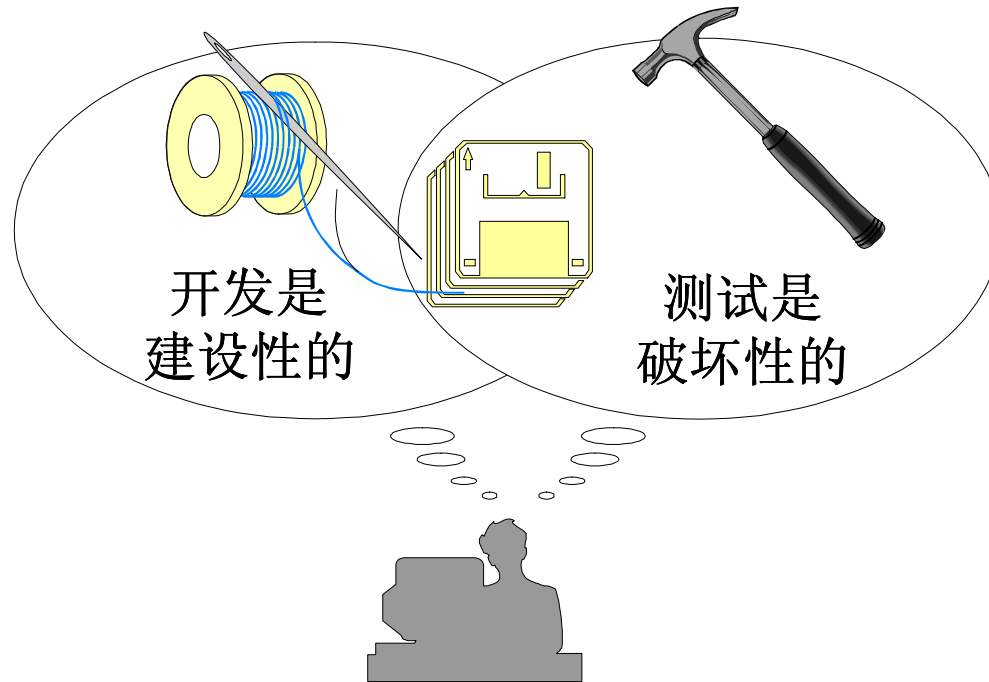
---

## ISTQB考试知识点

- ★ 认识测试的成功与否，会受测试心理因素的影响（**K1**）：
  - ✦ 清晰的测试目标决定了测试人员效率；
  - ✦ 人们往往会忽视自己的错误；
  - ✦ 认识到就事论事的交流方式以及反馈与问题相关信息的重要性；
- ★ 测试人员(**tester**)和开发人员(**developer**)思维方式的差异（**K2**）。

# 开发和测试的区别

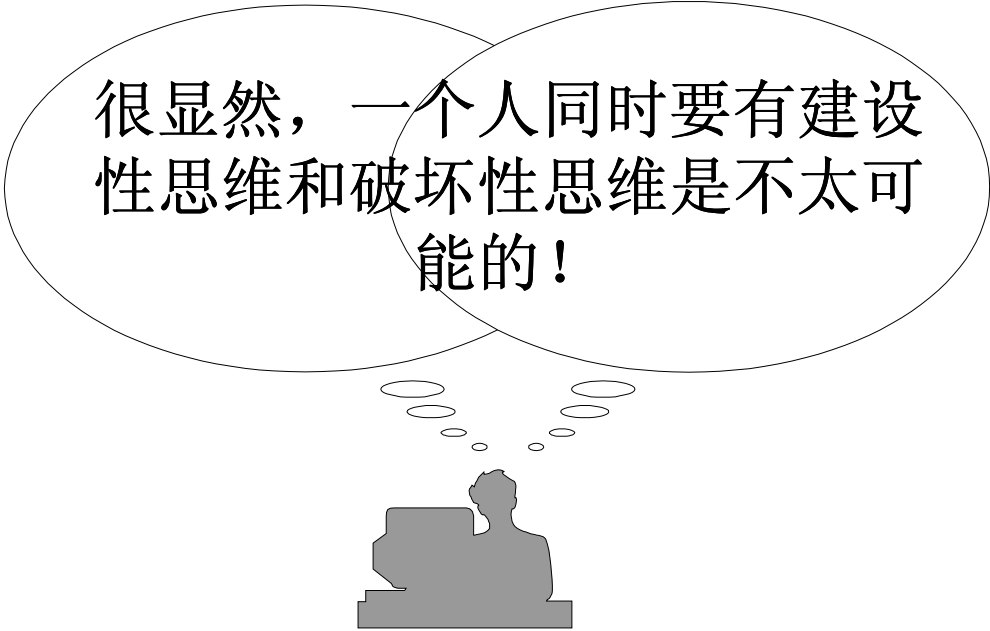
- ★ 开发是建设性的活动，而测试是可以看成破坏性的活动；



软件测试对于软件产品风险管理  
却是建设性的！

# 开发人员的测试

---



很显然，一个人同时要有建设性思维和破坏性思维是不太可能的！

因此，提高测试的效率和质量，需要将开发人员和测试人员应该分开！



# 不同级别的独立测试

---

- ★ 编写软件的人员来执行测试（低级别的独立）；
- ★ 由另外的开发人员（如来自同一开发小组）来执行测试；
- ★ 其他小组成员（如独立的测试小组）或测试专家（如可用性或性能测试专家）组成的测试团队来执行测试；
- ★ 其他组织或其他公司的成员组成测试团队来执行（如测试外包或其他外部组织的鉴定）测试；

独立测试的优点：客观性、专业性、权威性、资源有保障！

# 开发和测试的合作



- ★ 合作而不是争斗的方式开始项目，提醒项目的每位成员：我们的共同目标是追求高质量的产品；
- ★ 对产品中发现的问题以中性的和以事实为依据的方式来沟通，而不要指责引入这个问题的小组成员或个人。比如，客观而实际地编写缺陷报告和评审发现的问题；
- ★ 尽量理解其他成员的感受，以及他们为什么会有这种反应；
- ★ 确信其他成员已经理解你的描述，反之亦然；

---

# 答疑

