



# 第8章 系统测试

## System Testing

Xu Sihan (徐思涵)  
College of Computer Science  
Nankai University

*Slides adapted from materials by Prof. Qiang Liu (Tsing Hua University) and  
Ivan Marsic (Rugers University)*



# 软件测试的定义



# 软件测试的定义

## 正向思维：验证软件正常工作

- 评价一个程序或系统的特性或能力并确定是否达到预期的结果。
- 在设计规定的环境下运行软件的所有功能，直至全部通过。

## 逆向思维：假定软件有缺陷

- 测试是为了发现错误而针对某个程序或系统的执行过程。
- 寻找容易犯错地方和系统薄弱环节，试图破坏系统直至找不出问题。



# 软件测试的目的

- **直接目标：发现软件错误**
  - 以最少的人力、物力和时间找出软件中潜在的各种错误和缺陷，通过修正这些错误和缺陷 提高软件质量，回避软件发布后由于潜在的软件错误和缺陷造成的隐患所带来的商业风险。
- **期望目标：检查系统是否满足需求**
  - 测试是对软件质量的度量和评估，以验证软件的质量满足客户需求的程度，为用户选择和 接受软件提供有力的依据。
- **附带目标：改进软件过程**
  - 通过分析错误产生的原因，可以帮助发现当前开发所采用的软件过程缺陷，从而进行软件 过程改进；通过分析整理测试结果，可以修正软件开发规则，为软件可靠性分析提供依据。



# 测试的局限性

## 测试的不彻底性

- 测试只能说明错误的存在，但不能说明错误不存在
- 经过测试后的软件不能保证没有缺陷和错误

## 测试的不完备性

- 测试无法覆盖到每个应该测试的内容
- 不可能测试到软件的全部输入与响应
- 不可能测试到全部的程序分支的执行路径

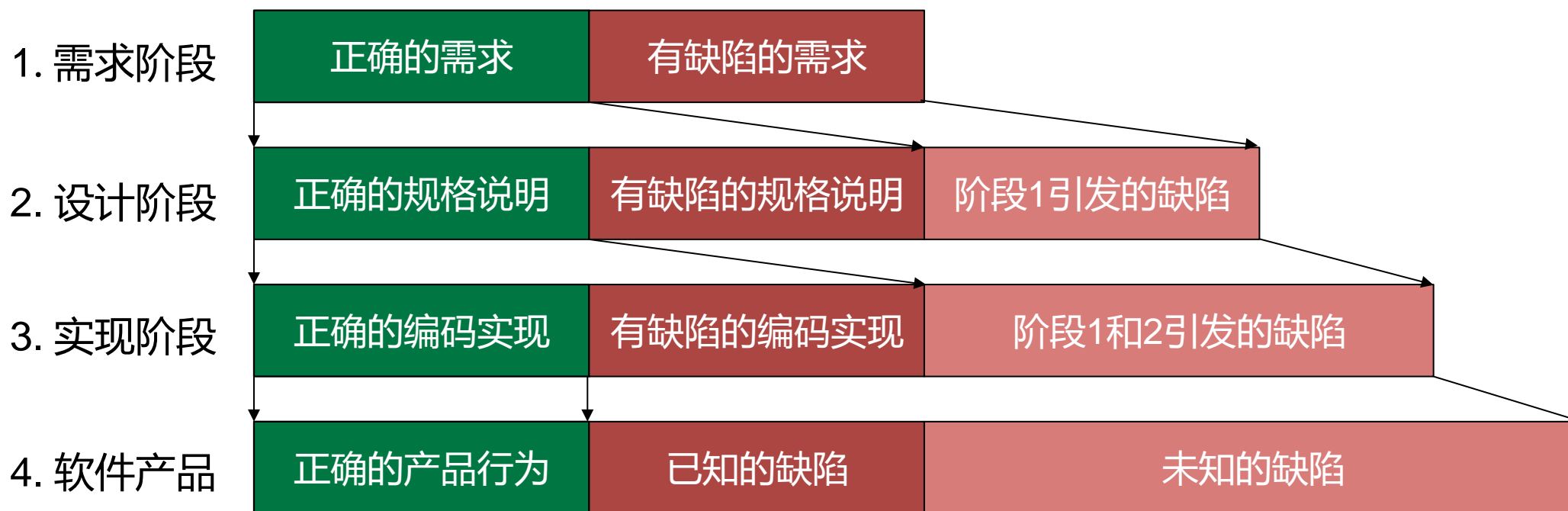
## 测试作用的间接性

- 测试不能直接提高软件质量，软件质量的提高要依靠开发
- 测试通过早期发现缺陷并督促修正缺陷来间接地提高软件质量



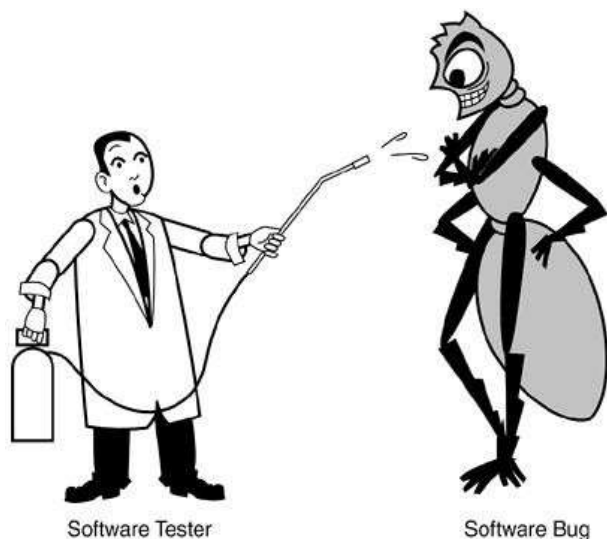
# 测试应尽早介入

尽早地在缺陷刚引入时就发现和修复，可以有效地避免缺陷的雪崩效应。



# 杀虫剂悖论

用同样的测试用例多次重复进行测试，最后将不再能够发现新的缺陷。



如同给果树喷撒农药，为了杀灭害虫只打一种杀虫药，虫子就会有抗体而变得适应，于是杀虫剂将不再发挥作用。

测试用例需要定期评审和修改，同时要不断增加新的不同测试用例来测试软件的不同部分，从而发现更多潜在的缺陷。

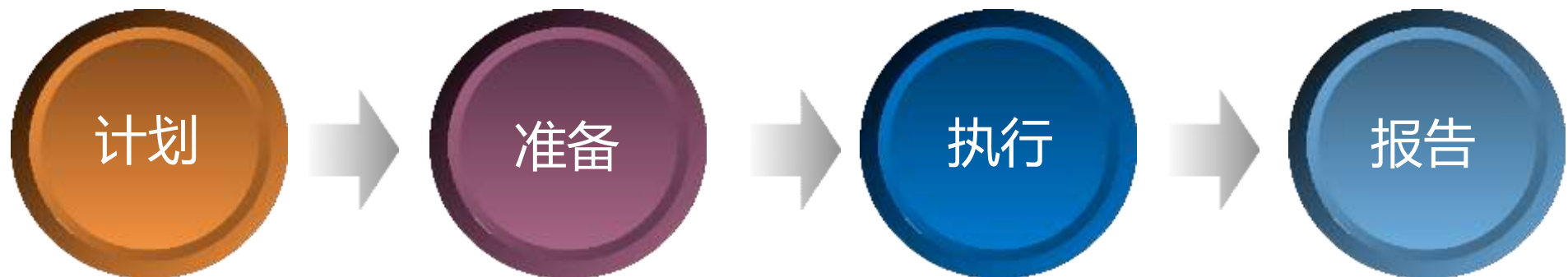


# 软件测试思维





# 软件测试过程



- 识别测试需求
- 分析质量风险
- 拟定测试方案
- 制定测试计划

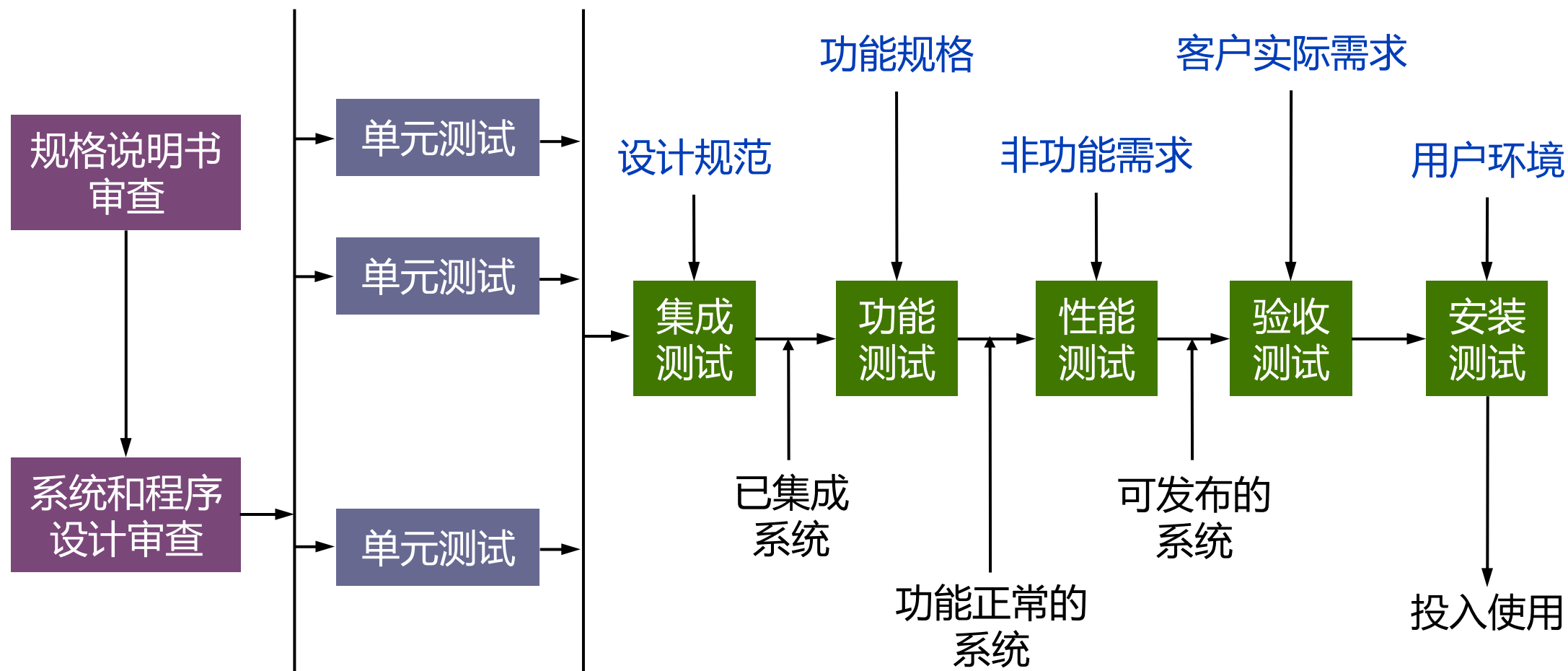
- 组织测试团队
- 设计测试用例
- 开发测试工具和脚本
- 准备测试数据

- 获得测试版本
- 执行和实施测试
- 记录测试结果
- 跟踪和管理缺陷

- 分析测试结果
- 评价测试工作
- 提交测试报告



# 软件测试活动



# 软件测试类型

测试对象角度

单元测试、集成测试、系统测试、验收测试

测试技术角度

黑盒测试（功能测试）、白盒测试（结构测试）

程序执行角度

静态测试、动态测试

人工干预角度

手工测试、自动化测试



# 单元测试

单元测试 (Unit Testing) 是对软件基本组成单元进行的测试，其测试对象是软件设计的最小单位（模块或者类）。



单元测试



单元测试



单元测试



单元测试



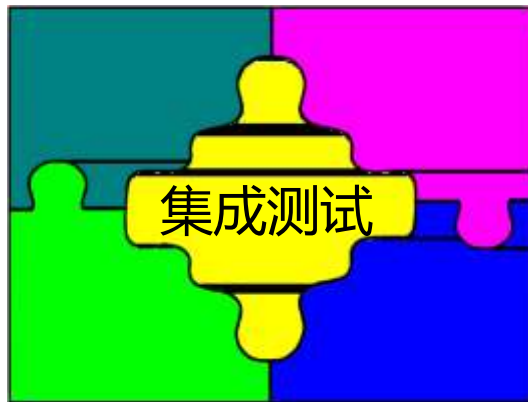
单元测试

单元测试一般由编写代码的开发人员执行，用于检测被测代码的功能是否正确。



# 集成测试

集成测试 (Integration Testing) 是在单元测试的基础上，将所有模块按照总体设计的要求组装成为子系统或系统进行的测试。



- 一次性集成方式：分别测试每个单元，再一次性将所有单元组装在一起进行测试。
- 渐增式集成方式：先对某几个单元进行测试，然后将这些单元逐步组装成较大的系统，在组装过程中边连接边测试。

集成测试对象是模块间的接口，其主要目的是找出在模块接口（包括系统体系结构）设计上的问题。



# 功能测试

功能测试 (Functional Testing) 是在已知产品所应具有的功能基础上，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。



# 功能测试

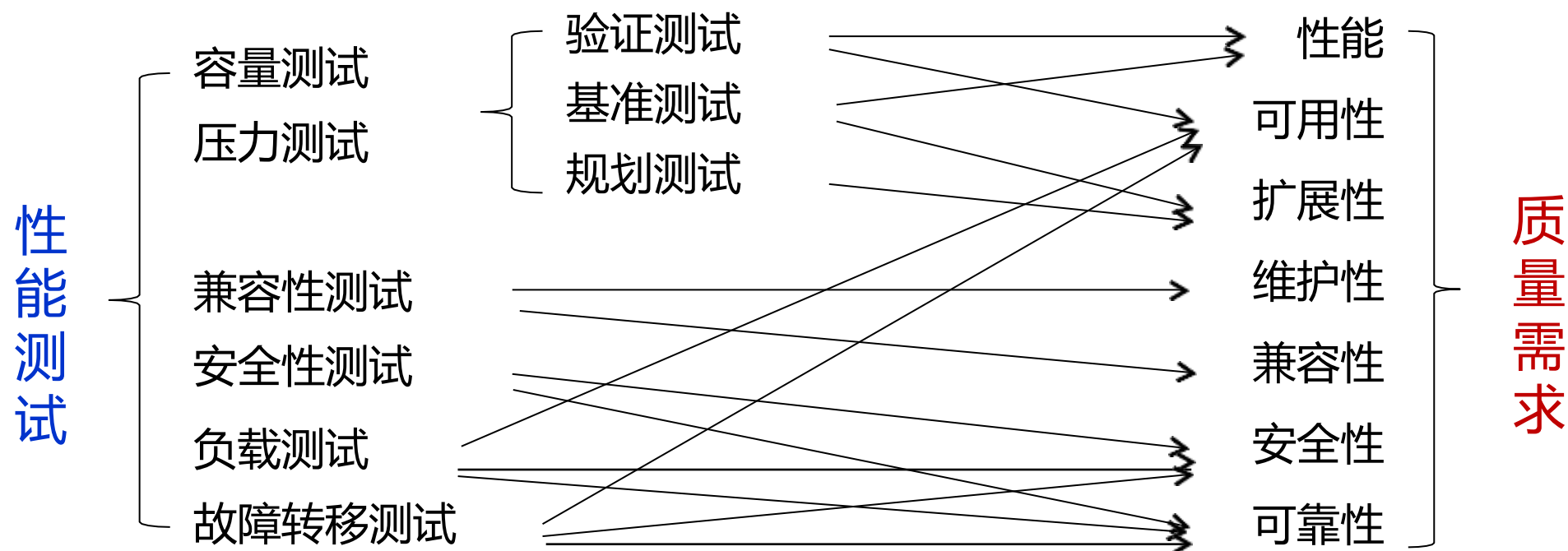


## 网上搜索并选购图书



# 性能测试

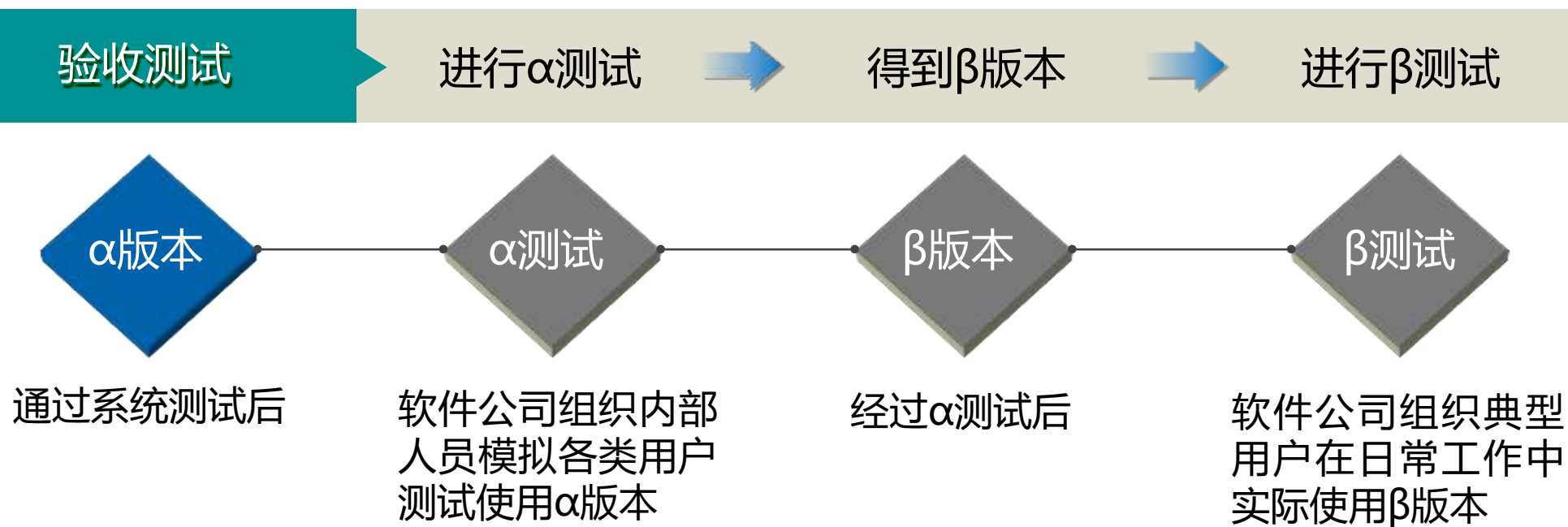
性能测试 (Performance Testing) 是在实际或模拟实际的运行环境下，针对非功能特性所进行的测试，包括压力测试、容量测试、安全测试和可靠性测试等。





# 验收测试

**验收测试**是在软件产品完成了系统测试之后、产品发布之前进行的软件测试活动，其目的是验证软件的功能和性能是否能够满足用户所期望的要求。



# 安装测试

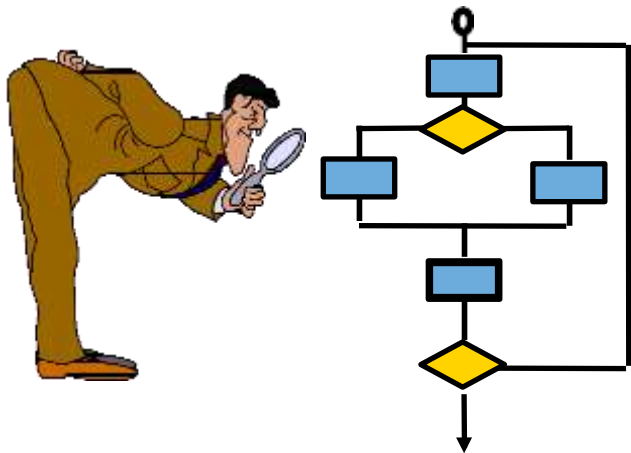
安装测试是系统验收之后，需要在目标环境中进行安装，其目的是保证应用程序能够被成功地安装。

- 应用程序是否可以成功地安装在以前从未安装过的环境中？
- 应用程序是否可以成功地安装在以前已有的环境中？
- 配置信息定义正确吗？
- 考虑到以前的配置信息吗？
- 在线文档安装正确吗？
- 安装应用程序是否会影响其他的应用程序吗？
- 安装程序是否可以检测到资源的情况并做出适当的反应？



# 黑盒测试与白盒测试

- **黑盒测试：**将测试对象看做一个黑盒子，完全 不考虑程序内部的逻辑结构和内部特性，只是 依据程序的需求规格说明书，检查程序的功能 是否符合它的功能说明。



**白盒测试：**把测试对象看做一个透明的盒子，允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。



# 静态测试与动态测试

**静态测试：**通过人工分析或程序正确性证明的方式来确认程序正确性。

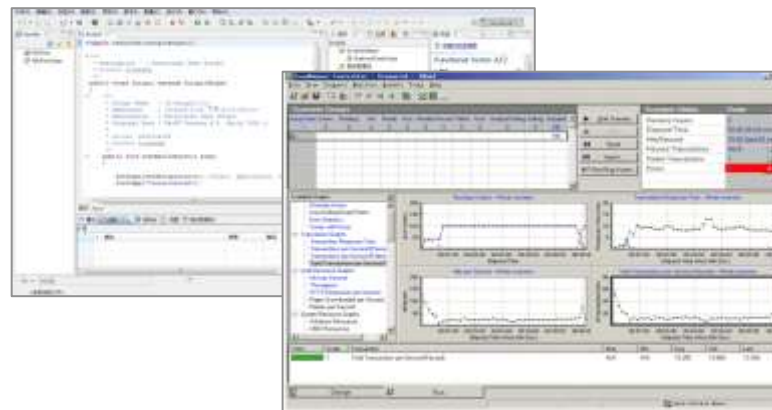


**动态测试：**通过动态分析和程序测试等方式检查程序执行状态，以确认是否有问题。

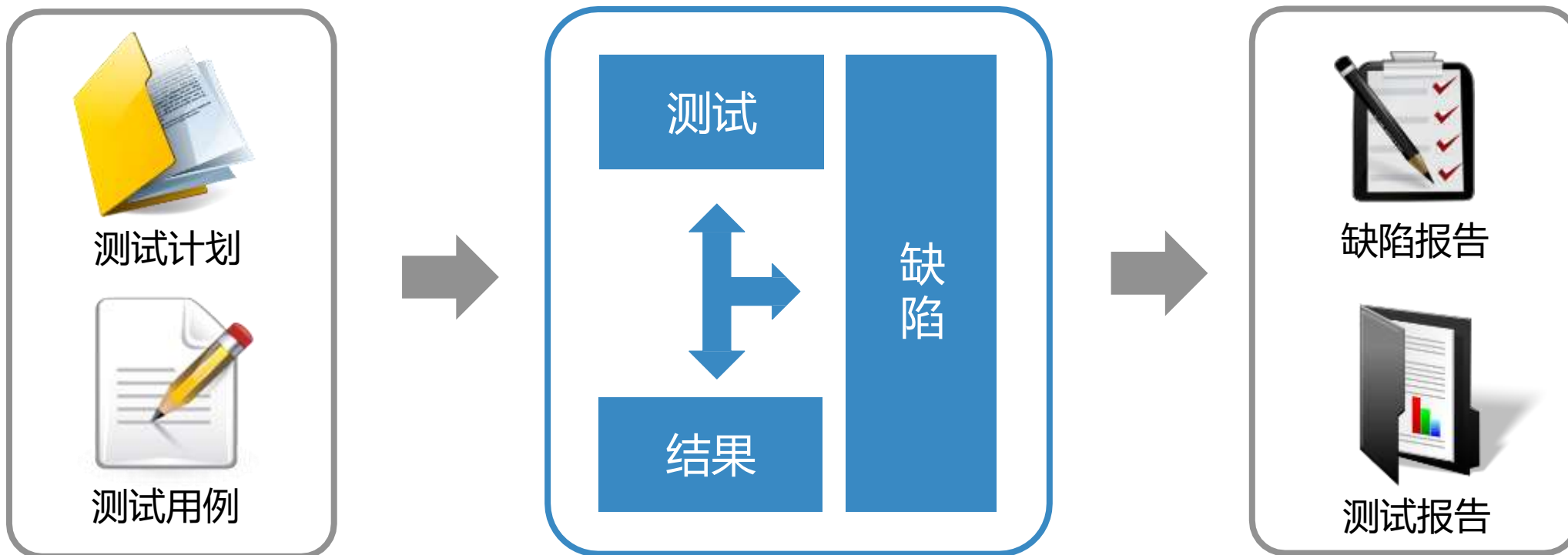


# 手工测试与自动化测试

- **手工测试**：测试人员根据测试大纲中所描述的测试步骤和方法，手工地输入测试数据并记录测试结果。
- **自动化测试**：相对于手工测试而言，主要是通过所开发的软件测试工具或脚本等手段，按照测试工程师的预定计划对软件产品进行的自动测试。



# 软件测试文档



# 软件测试计划



# 测试用例文档

标识符: 1007

测试项: 记事本程序的文件菜单栏——文件 / 退出菜单的功能测试

测试环境: Windows 7 Professional 中文版

前置条件: 无

操作步骤:

1. 打开记事本程序
2. 输入一些字符
3. 鼠标单击菜单“文件→退出”。

输入数据	期望输出	实际结果
空串	系统正常退出, 无提示信息	
A	系统提示“是否将更改保存到无标题(或指定文件名)?”单击“保存”, 系统将打开保存/另存窗口; 单击“不保存”, 系统不保存文件并退出; 单击“取消”系统将返回记事本窗口。	

结论: ☐ 通过 ☐ 不通过

测试人:

测试日期:





# 缺陷报告内容

## 基本描述：

- 用一句话简单地描述清楚问题。

## 详细描述：

1. 描述问题的基本环境，包括操作系统、硬件环境、网络环境、被测软件的运行环境等
2. 用简明扼要的语言描述清楚软件异常、操作步骤和使用数据
3. 截图
4. 被测软件运行时相关日志文件或出错信息
5. 测试人员根据信息可以给出对问题的简单分析
6. 被测软件的版本
7. 缺陷状态、严重性和优先级
8. 提交日期和提交人

## 相关附件：

- 截图文件、出错信息



# 缺陷报告内容

缺陷的**严重性**是指缺陷对软件产品使用的影响程度。

## 缺陷严重性

## 描述

致命的（1级）	造成系统或应用程序崩溃、死机、挂起，或造成数据丢失、主要功能完全丧失。
严重的（2级）	系统功能或特性没有实现、主要功能部分丧失、次要功能完全丧失或者致命的错误声明。
一般的（3级）	缺陷虽不影响系统的基本使用，但没有很好地实现功能，没有达到预期效果，如次要功能丧失、提示信息不太明确、用户界面差、操作时间长等。
微小的（4级）	对功能几乎没有影响，产品及其属性仍可使用，如存在个别错别字、文字排列不整齐等。



# 缺陷报告内容

缺陷的**优先级**是指缺陷应该被修复的紧急程度。

缺陷优先级	描述
立即解决 (P1)	缺陷导致系统几乎不能使用或测试不能继续, 需要立即修复
高优先级 (P2)	缺陷严重, 影响测试, 需要优先考虑
正常排队 (P3)	缺陷需要正常排队等待修复
低优先级 (P4)	缺陷可以在开发人员有时间的时候被纠正



# 功能测试

功能测试 (Functional Testing) 是在已知产品所应具有的功能基础上，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。



# 功能测试

界面

数据

操作

逻辑

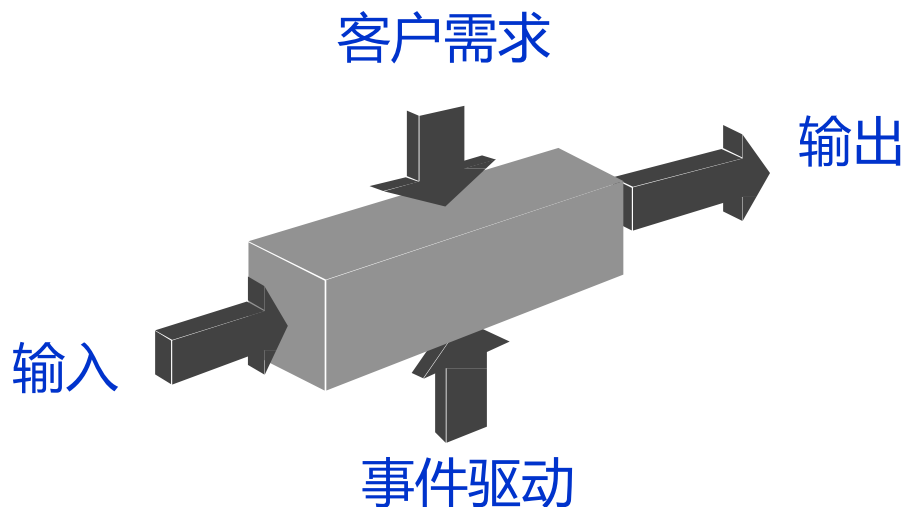
接口

- 程序安装、启动正常，有相应的提示框、错误提示等
- 每项功能符合实际要求
- 系统的界面清晰、美观
- 菜单、按钮操作正常、灵活，能处理一些异常操作
- 能接受正确的数据输入，对异常数据的输入有提示、容错处理等
- 数据的输出结果准确，格式清晰，可以保存和读取
- 功能逻辑清楚，符合使用者习惯
- 系统的各种状态按照业务流程而变化，并保持稳定
- 支持各种应用的环境
- 配合多种硬件周边设备
- 软件升级后，能继续支持旧版本的数据
- 与外部应用系统的接口有效



# 功能测试方法

## 黑盒测试方法



具有代表性和典型性

寻求系统设计和功能设计的弱点

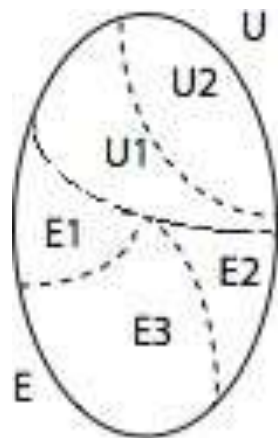
既有正确输入也有错误或异常输入

考虑用户实际的诸多使用场景



# 回顾：等价类划分

等价类划分是一种典型的、重要的黑盒测试方法，它将程序所有可能的输入数据（有效的和无效的）划分成若干个等价类，然后从每个部分中选取具有代表性的数据作为测试用例，测试用例由有效等价类和无效等价类的代表组成。

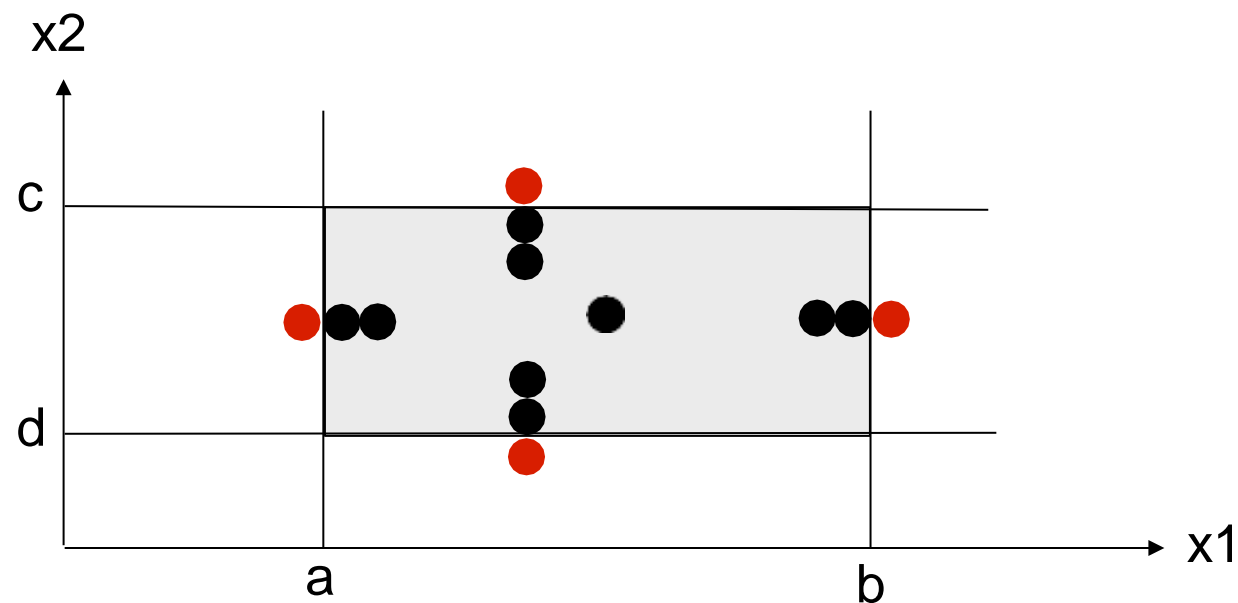


- E 表示所有正常和合法的输入
- U 表示所有异常和非法的输入



# 回顾：边界值分析

边界值分析是对输入或输出的边界值进行测试的一种方法，它通常作为等价类划分法的补充，这种情况下的测试用例来自等价类的边界。





# 场景法

**场景法**是通过运用场景对系统的功能点或业务流程进行描述，对应不同的业务场景生成相应的测试用例，从而发现需求和实现中存在的问题。



场景1	基本流
场景2	基本流、备选流1
场景3	基本流、备选流1、备选流2
场景4	基本流、备选流3
场景5	基本流、备选流3、备选流1
场景6	基本流、备选流3、备选流1、备选流2
场景7	基本流、备选流4
场景8	基本流、备选流3、备选流4



## 举例：ATM取款



# 举例：ATM取款

用例描述	
基本流	成功提款
备选2	ATM内没有现金
备选3	ATM内现金不足
备选4	PIN有误（还有输入机会）
备选5	PIN有误（没有输入机会）
备选6	账户不存在 / 账户类型有误
备选7	账户余额不足



# 举例：ATM取款

## 场景设计

场景1：成功提款	基本流
场景2：ATM内没有现金	基本流、备选流2
场景3：ATM内现金不足	基本流、备选流3
场景4：PIN有误（还有输入机会）	基本流、备选流4
场景5：PIN有误（没有输入机会）	基本流、备选流5
场景6：账户不存在 / 账户类型有误	基本流、备选流6
场景7：账户余额不足	基本流、备选流7



# 举例：ATM取款

## 测试用例设计

序号	场景	PIN	账号	取款金额	账面金额	ATM 现金	预期结果
1	场景1：成功提款	V	V	V	V	V	成功提款
2	场景2：ATM里没有现金	V	V	V	V	X	提款选项不可用，用例结束
3	场景3：ATM里现金不足	V	V	V	V	X	警告信息，返回基本流相应步骤，重新输入金额
4	场景4：PIN有误 (还有不止一次机会)	X	V	n/a	V	V	警告信息，返回基本流相应步骤，重新输入PIN
5	场景4：PIN有误 (还有一次机会)	X	V	n/a	V	V	警告信息，返回基本流相应步骤，重新输入PIN
6	场景5：PIN有误 (不再有输入机会)	X	V	n/a	V	V	警告信息，卡被没收，用例结束

.....



# 讨论思考



当当网  
dangdang.com



如何使用场景法  
设计测试用例



# 错误推测法

**错误推测法**是人们根据经验或直觉推测程序中可能存在的各种错误，从而有针对性地编写检查这些错误的测试用例的方法。

- 软件缺陷具有空间聚集性，80%的缺陷常常存在于20%的代码中。因此，应当记住常常光临代码的高危多发“地段”，这样发现缺陷的可能性会大得多。
- 列举程序中所有可能的错误和容易发生错误的特殊情况，根据可能出现的错误情况选择测试用例。

80%  
20



# Web应用功能测试





# Web应用功能测试

Web应用功能测试主要用来测试Web应用软件是否履行了预期功能，包括内容测试、链接测试、表单测试、Cookies测试、设计语言测试、数据库测试等。

网页测试



网站测试



# 内容测试

内容测试用于检测Web应用系统提供信息的正确性、准确性和相关性。

- **正确性**是指信息是真实可靠的还是胡乱编造的。例如，一条虚假的新闻报道可能引起不良的社会影响，甚至会让公司陷入麻烦之中，也可能惹上法律方面的问题。
- **准确性**是指网页文字表述是否符合语法逻辑或者是否有拼写错误。在Web应用系统开发的过程中，开发人员可能不是特别注重文字表达，因此测试人员需要检查页面内容的文字表达是否恰当以及页面样式是否看起来更舒服。
- **相关性**是指能否在当前页面可以找到与当前浏览信息相关的信息列表或入口，也就是一般Web站点中所谓的“相关文章列表”。测试人员需要确定是否列出了相关内容的站点链接。



# 内容测试



1	搜索某种类别的商品	关键字=软件工程	列出含有指定关键字名称的图书
2	让鼠标滑过每一个商品	所有商品图片	鼠标滑过每一个图片时显示相应的商品名称



# 链接测试

链接使用户可以从一个页面浏览到另一个页面，是Web应用系统的一个主要特征，它是在页面之间切换和指导用户去一些不知道地址的页面的主要手段。

链接测试需要验证三个方面的问题：

- 用户点击链接是否可以顺利地打开所要浏览的内容，即链接是否按照指示的那样确实链接到了要链接的页面。
- 所要链接的页面是否存在。实际上，好多不规范的小型站点，其内部链接都是空的，让用户感觉很差。
- 保证Web应用系统上没有孤立的页面，所谓孤立页面是指没有链接指向该页面，只有知道正确的URL地址才能访问。



# 表单测试

- 当用户给Web应用系统提交信息时，就需要使用表单操作。表单测试主要是模拟表单 提交过程，检测其准确性，确保提交信息在整个过程中是正确的。
- 表单测试主要考虑以下方面：
  - 表单提交应当模拟用户提交，验证是否完成功能。
  - 测试提交操作的完整性，以校验提交给服务器的信息的正确性。
  - 验证数据的正确性和异常情况的处理能力，注意是否符合易用性要求。
  - 在测试表单时，会涉及到数据校验问题。如果根据给定规则需要对用户输入进行 校验，需要保证这些校验功能正常工作。





注册

您是凤凰知音会员？ 是 否 ☐

网站注册用户信息

用户名

密码

确认密码

姓

名

电子邮箱

☒ 我希望通过此邮箱收到国航最新促销信息

手机号码

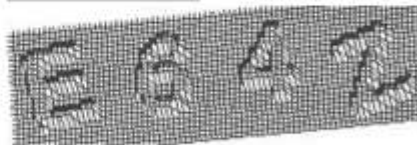
获取手机验证码

暂不支持173、177、1701、1702号段的手机号码

手机验证码

验证码

请输入验证码



换一张？

— 注册凤凰知音会员

使用条款和条件

进入本网站的访问者接受本协议条款，注册成为国航电子客票用户，并遵守本协议所述之条款使用本网站所提供之服务。如果不接受本声明之条款，请勿使用本网站。接受本声明之条款，您将遵守本协议之规定。

☒ 我已仔细阅读并同意“服务协议”

注册



# Cookies测试

Cookies通常用来存储用户信息和用户在某些应用系统上的操作序列，当一个用户使用Cookies访问了某一应用时，Web服务器将发送有关用户信息，并把该信息以Cookies形式存储在客户端计算机上，可用来创建动态和自定义页面或者存储登录等信息。





# Cookies测试





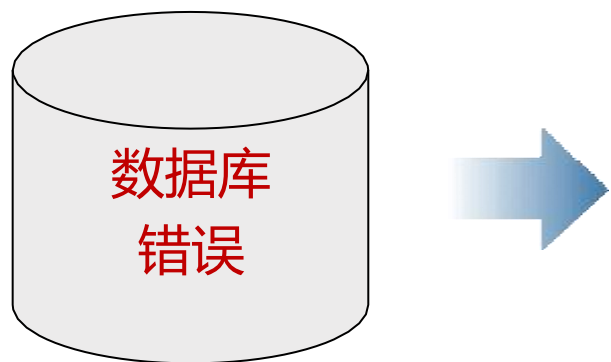
# 特定功能测试

测试人员需要对Web网站特定的功能需求进行验证，需要强调的是应该从客户或用户的角度，而不是从产品本身呈现的进行评判。



# 数据库测试

在Web应用技术中，数据库起着很重要的作用，它为Web应用系统的管理、运行、查询和实现用户对数据存储的请求等提供空间。在Web应用中，最常用的数据库类型是关系型数据库，可以使用SQL对信息进行处理。



- **数据一致性错误**：主要是由于用户提交的表单信息不正确而造成的
- **输出错误**：主要是由于网络速度或程序设计问题等引起的



# 设计语言测试

Web设计语言版本的不同会引起客户端或服务器端比较严重的问题，例如使用哪种版本的HTML等。此外，使用Java、JavaScript、ActiveX、VBScript或Perl等开发的应用程序也要在不同的版本上进行验证。



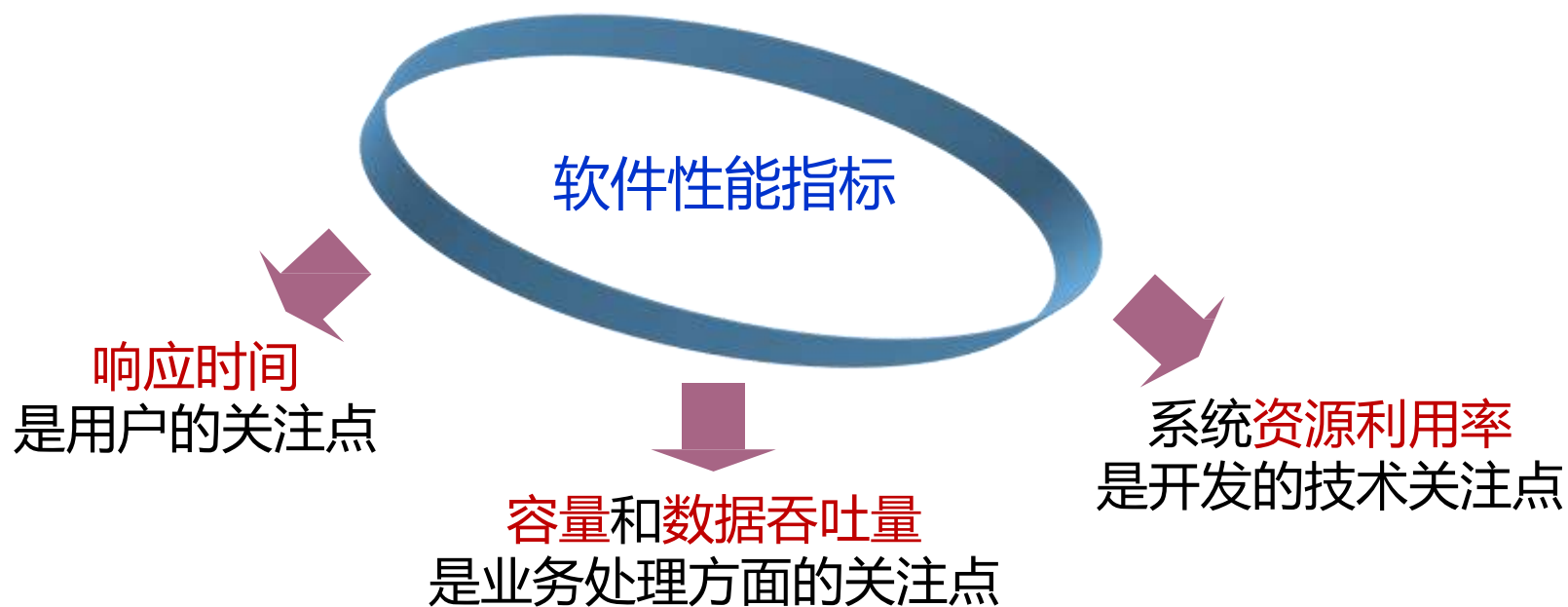
# 兼容性测试

- **平台测试：**市场上有很多不同的操作系统，常见的有Windows、Unix、Mac OS、Linux等，同一个应用可能在某些操作系统下能正常运行，但在另一些操作系统下可能会运行失败。
- **浏览器测试：**浏览器是Web系统客户端最核心的软件，来自不同厂商的浏览器对Java、JavaScript、ActiveX、plug-ins或不同的HTML有不同的支持，并且有些 HTML 命令或脚本只能在某些特定的浏览器上运行。
- **分辨率测试：**应检查页面版式在不同的分辨率模式下是否显示正常、字体大小是否合适于浏览、文本和图片是否对齐等。
- **打印机测试：**用户可能会将网页打印下来，需要验证网页打印是否正常。有时在屏幕上显示的图片 and 文本的对齐方式可能与打印出来的东西不一样，测试人员至少需要验证订单确认页面打印是正常的。



# 软件性能

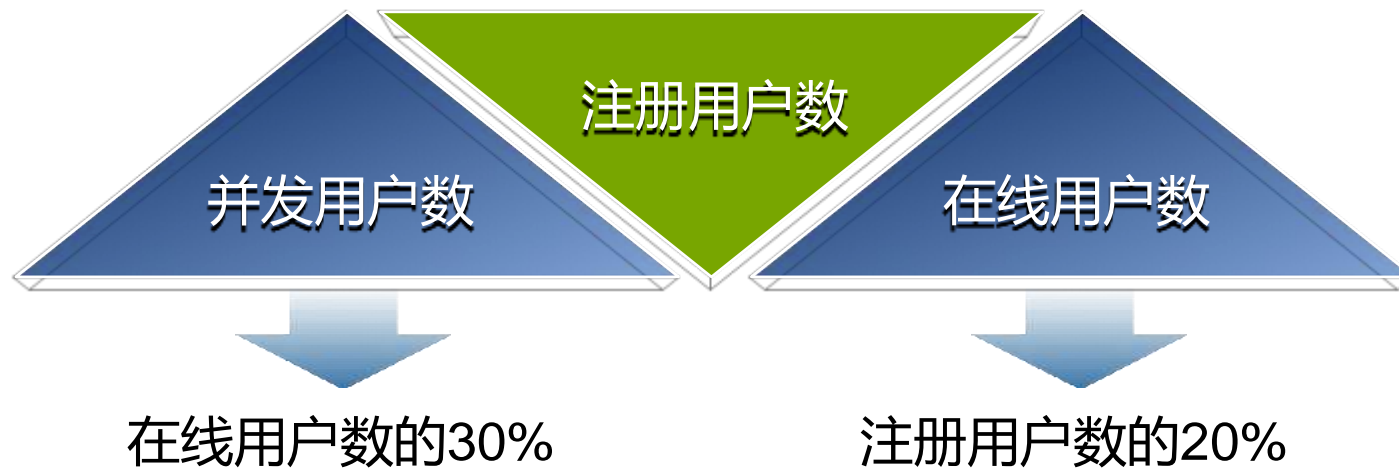
软件性能是一种非功能特性，它关注的不是软件是否能完成特定功能，而是完成功能时展示出的及时性、资源占用、稳定性、安全性、兼容性、可扩展性、可靠性等。



# 性能指标

## 用户数

- 注册用户数：所有在系统注册的用户数目
- 在线用户数：所有正在访问系统的用户（不一定做操作）数目
- 并发用户数：在某一给定时间内某特定时刻进行会话操作的用户数



# 性能指标



新浪微博  
weibo.com

- 假设用户只有注册后才能使用，但注册用户并不是每时每刻都在使用该网站，因此具体一个时刻只有部分注册用户同时在线。
- 在线用户在浏览网站时会花很多时间阅读网站的信息，因而具体一个时刻只有部分在线用户同时向系统发出请求。
- 由于注册用户可能长时间不登陆网站，使用注册用户数作为性能指标会造成很大的误差。在线用户数和并发用户数都可作为性能指标，但性能测试更多关心并发用户数。

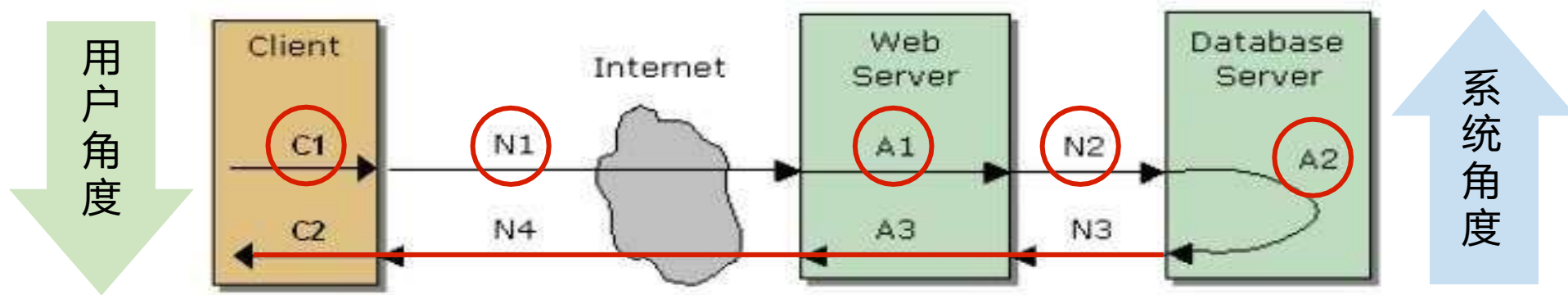


# 性能指标

## 响应时间

- 从客户端发出请求到获得响应的整个过程所经历的时间

$$\text{响应时间} = (A1+A2+A3)+(N1+N2+N3+N4)$$



$$\text{响应时间} = (C1+C2)+(A1+A2+A3)+(N1+N2+N3+N4)$$





# 性能指标

## 吞吐量

- 吞吐量 (Throughout) 是指单位时间内系统处理的客户请求的数量，直接体现软件系统的性能承载能力。
- 从业务的角度，吞吐量可以用请求数 / 秒、页面数 / 秒、访问数 / 秒或处理业务数 / 小时等来衡量。

## 资源利用率

- 资源利用率是指系统资源的使用程度，例如服务器CPU利用率、内存利用率、磁盘利用率、网络带宽利用率等。
- 一般由“资源实际使用 / 总的资源可用量”表示，如CPU利用率68%、内存利用率55%等。



# 性能测试

性能测试是通过自动化测试工具或手段模拟多种正常、峰值以及异常负载条件对系统的各项性能指标进行的一种测试。

## 负载测试

通过逐渐增加系统负载，测试系统性能的变化，最终确定在满足性能指标的情况下，系统能承受的最大负载量。

目标：在特定的运行条件下验证系统的能力状况。

## 压力测试

通过逐步增加系统负载，测试系统性能的变化，最终确定在什么负载条件下系统性能处于失效状态。

目标：发现在什么条件下应用程序的性能会变得不可接受。



# 性能测试

## 压力测试类型

- **稳定性压力测试**：高负载下持续运行24小时以上。
- **破坏性压力测试**：通过不断加载的手段快速造成系统崩溃，让问题尽快暴露出来。
- **渗入测试**：通过长时间运行，使问题逐渐渗透出来，从而发现内存泄漏、垃圾收集或系统的其他问题，以检验系统的健壮性。
- **峰谷测试**：采用高低突变加载方式进行，先加载到高水平的负载，然后急剧降低负载，稍微平息一段时间，再加载到高水平的负载，重复这样过程，容易发现问题的蛛丝马迹，最终找到问题的根源。



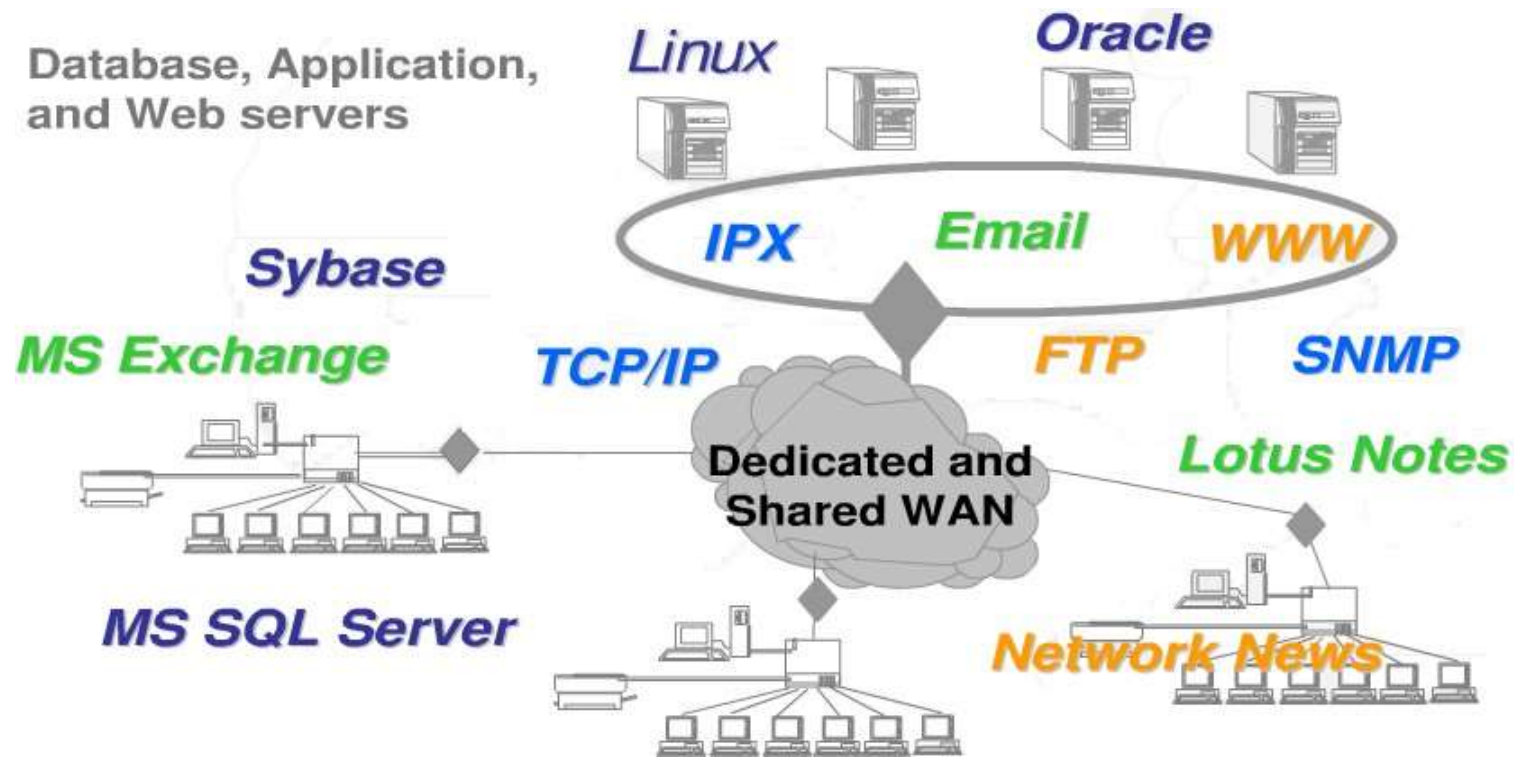
# 性能测试

## 大数据量测试

- 独立数据量测试是对某些系统存储、传输、统计、查询等进行的大数据量测试。
- 综合数据量测试是系统在具备一定数据量时，在负载压力测试下考察业务是否 能够正常进行的测试。
- **疲劳强度测试** 据量较大时系统的性能状况。
  - 采用系统稳定运行情况下，长时间运行系统的测试。
  - 目标：通过综合分析交易执行指标和资源监控指标来测试系统长时间无故障稳定运行的能力。



# 性能测试策略

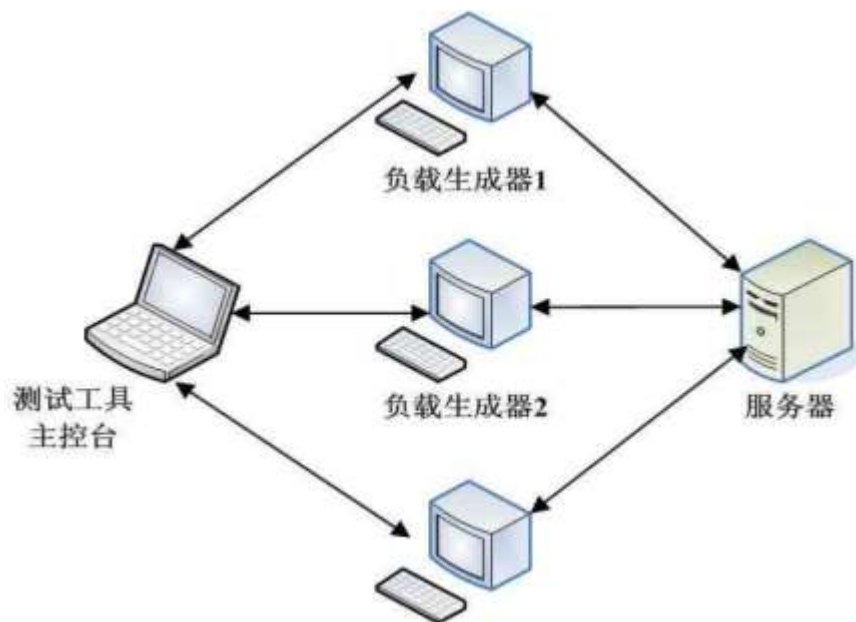


问题：如何找出问题的根源？



# 性能测试策略

**应用在客户端性能的测试：**逐渐增加并发虚拟用户数，直到系统瓶颈或者不能接收的性能点，通过综合分析交易执行指标、资源监控指标等来确定系统并发性能。



## 自动化性能测试工具

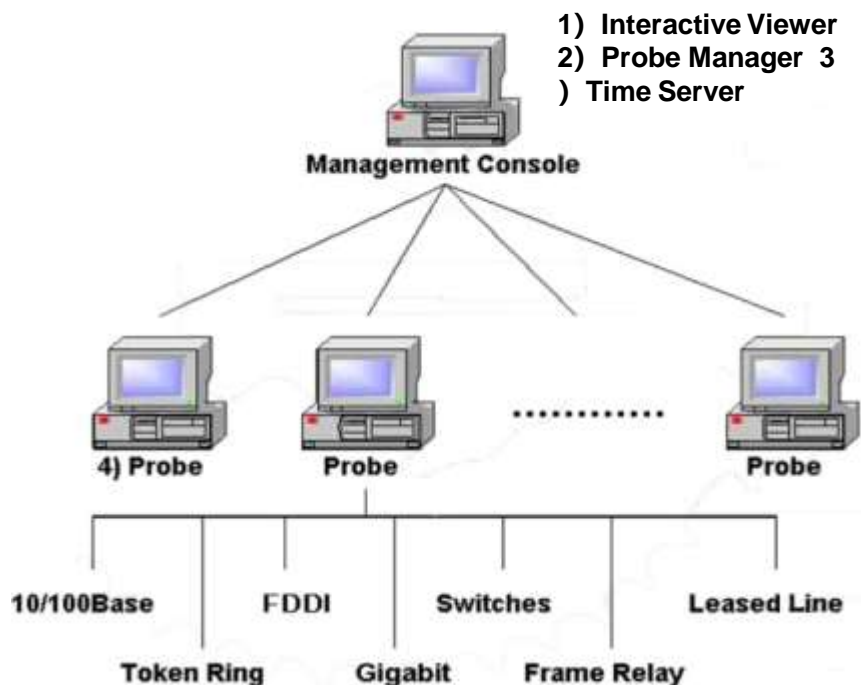
**实现机制：**通过在一台或几台PC机上模拟成百上千的虚拟用户同时执行业务的情景，对应用程序进行测试，从而度量系统性能并确定问题所在。

**步骤：**录制—改进—执行



# 性能测试策略

应用在网络性能测试：测试网络带宽、延时、负载和TCP端口的变化如何影响用户的响应时间。



## 网络测试监控工具

多个捕捉点，一个分析

**捕捉点：** Agent 被动监听数据包实现实时数据采集。

**分析：** Agent Manager完成对所跟踪到的数据的分析。



# 性能测试策略

**应用在服务器性能的测试：**测试关键点是资源占用情况、数据库性能和故障报警  
对于服务器性能的测试，可以采用工具监控，也可以使用系统本身的监控命令。

## 监控服务器 操作系统

- CPU
- 内存 & SWAP
- 磁盘管理
- 网络
- 文件系统
- 活动的进程

## 监控中间件

## 监控数据库

- 数据库中的关键资源
- 读写页面的使用情况
- 超出共享内存缓冲区的操作数
- 上一轮询期间作业等待缓冲区的时间
- 共享内存中物理日志和逻辑日志缓冲区的使用率
- 磁盘的数据块使用情况以及被频繁读写的热点区域
- 用户事务或者表空间监控事务日志
- 数据库锁资源
- 关键业务的数据表的表空间增长
- SQL执行情况





# 性能测试工具

HP LoadRunner是一种预测系统行为和性能的负载测试工具，它通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题。LoadRunner能够对整个企业的架构进行测试，通过使用该工具，企业能最大限度地缩短测试时间，优化性能和加速应用系统的发布周期。



***LoadRunner***



# 性能测试工具

**JMeter**是Apache组织开发的基于Java的压力测试工具，不仅用于Web服务器的性能测试，也涵盖数据库、FTP、LDAP服务器等各种性能测试，可与Junit、Ant等工具集成。它可以针对服务器、网络或其他被测试对象等大量并发负载进行强度测试，分析在不同压力负载下系统的整体性能。



# 谢谢大家！

---

