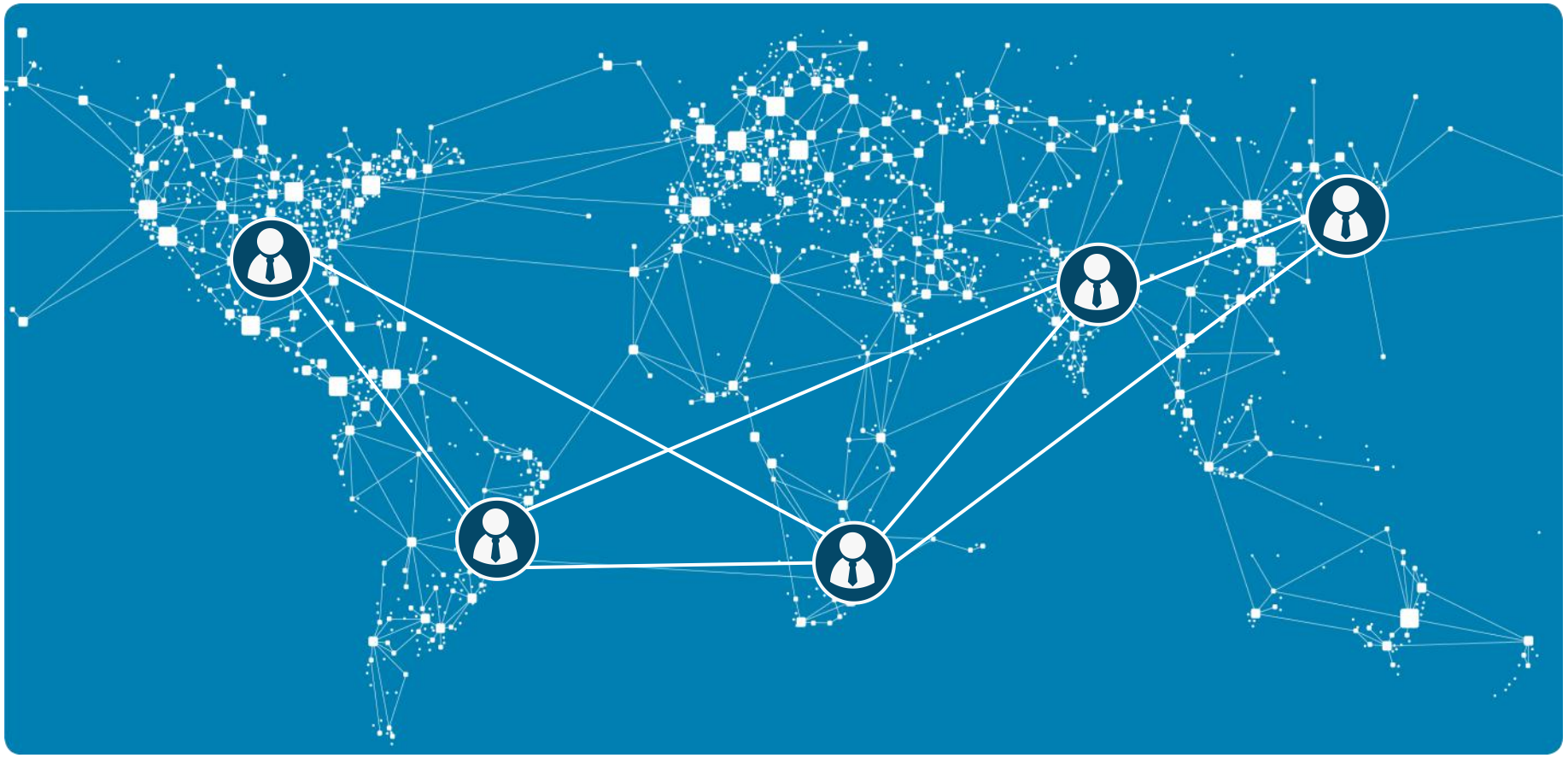


软件测试

软件系统测试

软件测试概念

软件测试类型

软件功能测试

软件性能测试

软件质量

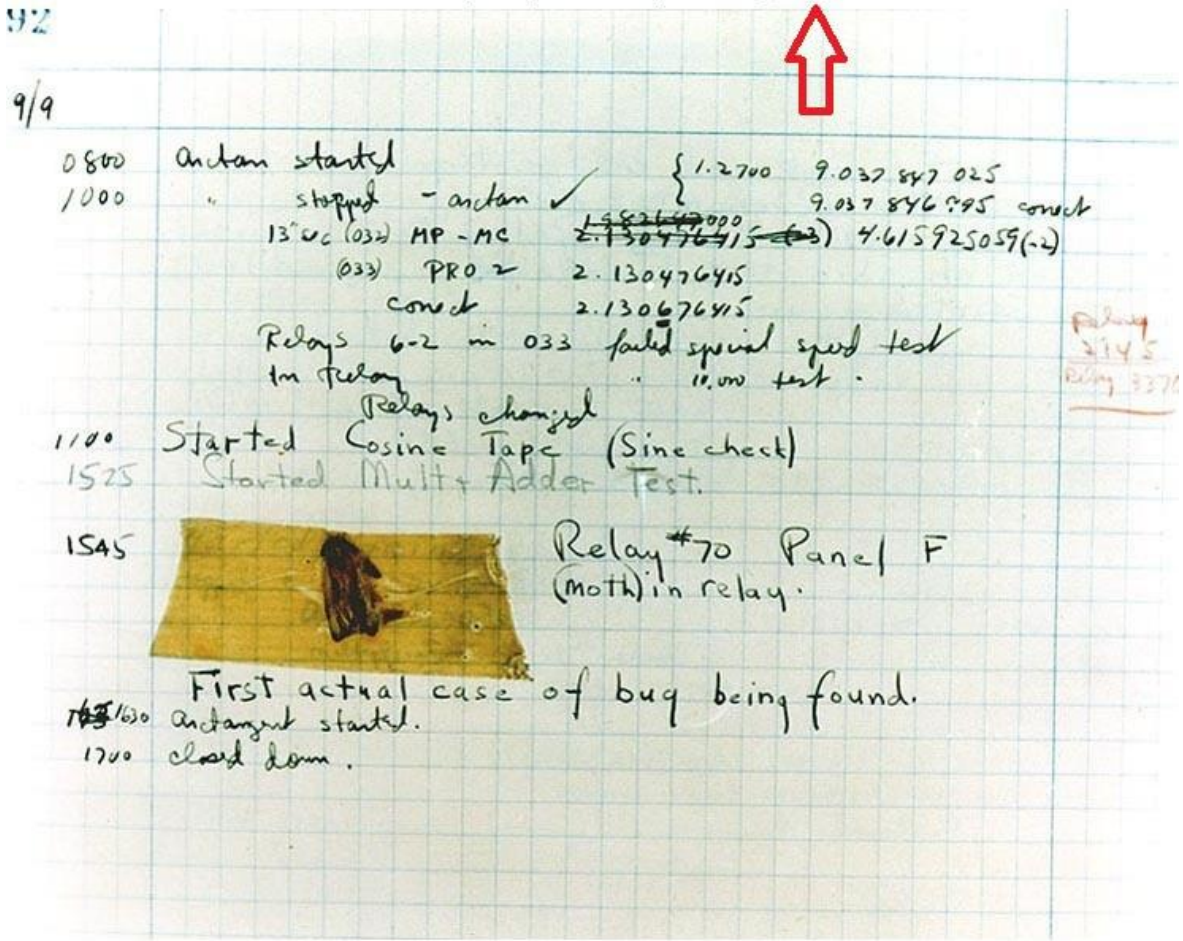
The average software product released on the market is not error free.

Win98 发布日的尴尬

“Bug”的由来

“Bug”的由来

Photo # NH 96566-KN (Color) First Computer “Bug”, 1947



软件缺陷的产生

人的认识问题

- 认识问题
- 关注度问题
- 策略问题

人的开发工作

- 创造、理解、修改
- 创造、理解、修改
- 创造、理解、修改
- 实现、重用、修改

需求分析	需求缺陷
概要设计	设计缺陷
详细设计	
编码调试	编码缺陷

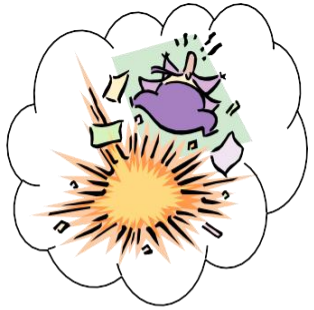
人工制品

软件产品

软件缺陷的演化

举例：爱国者导弹

1991年2月，一次拦截失利 造成28名美国士兵丧生



软件系统

计时用系统时钟（1/10秒）并以整数表达

缺少防范措施

缺陷 —激活→ 故障 —演变→ 失效

寄存器存储导致误差（0.000000095）$_{10}$

0.000000095×100h×60×60×10 = 0.34s

软件测试

测试 —— 发现问题 —— 修复

软件测试的定义

验证软件正常工作

假定软件有缺陷

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

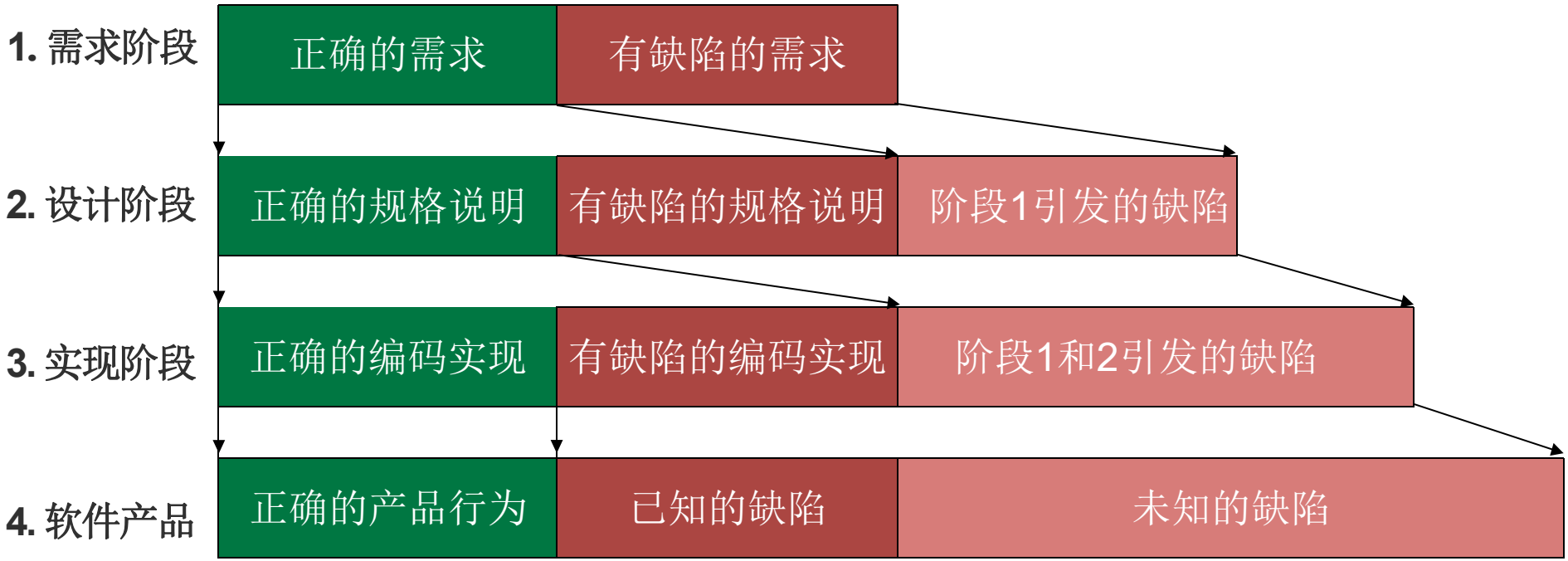
- 测试无法覆盖到每个应该测试的内容
- 不可能测试到软件的全部输入与响应
- 不可能测试到全部的程序分支的执行路径

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

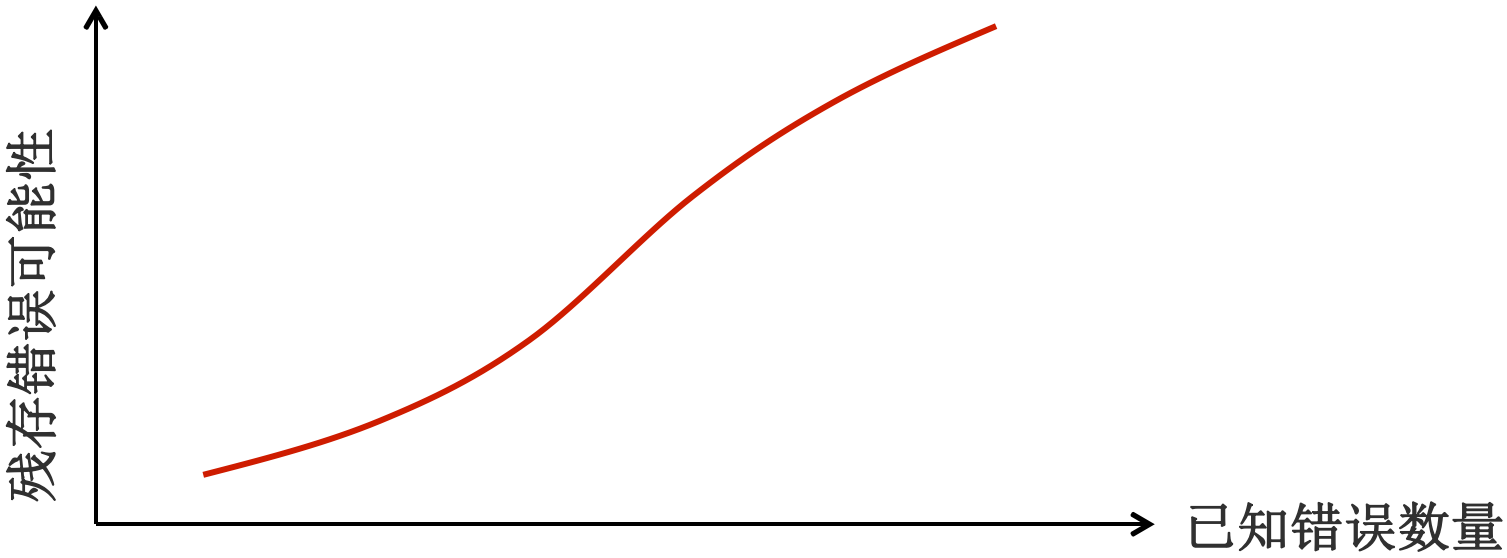
尽早地在缺陷刚引入时就发现和修复，可以有效地避免缺陷的雪崩效应。



缺陷的集群性

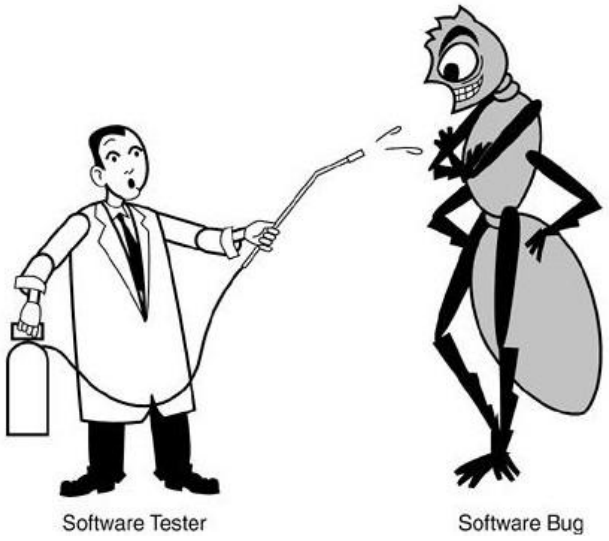
软件错误具有聚集性，对存在错误的部分应重点测试。

- **80/20**原则：**80%**的软件错误存在于**20%**的代码行中
- 经验表明：测试后程序中残留的错误数目与该程序中已检出的错误成正比。



杀虫剂悖论

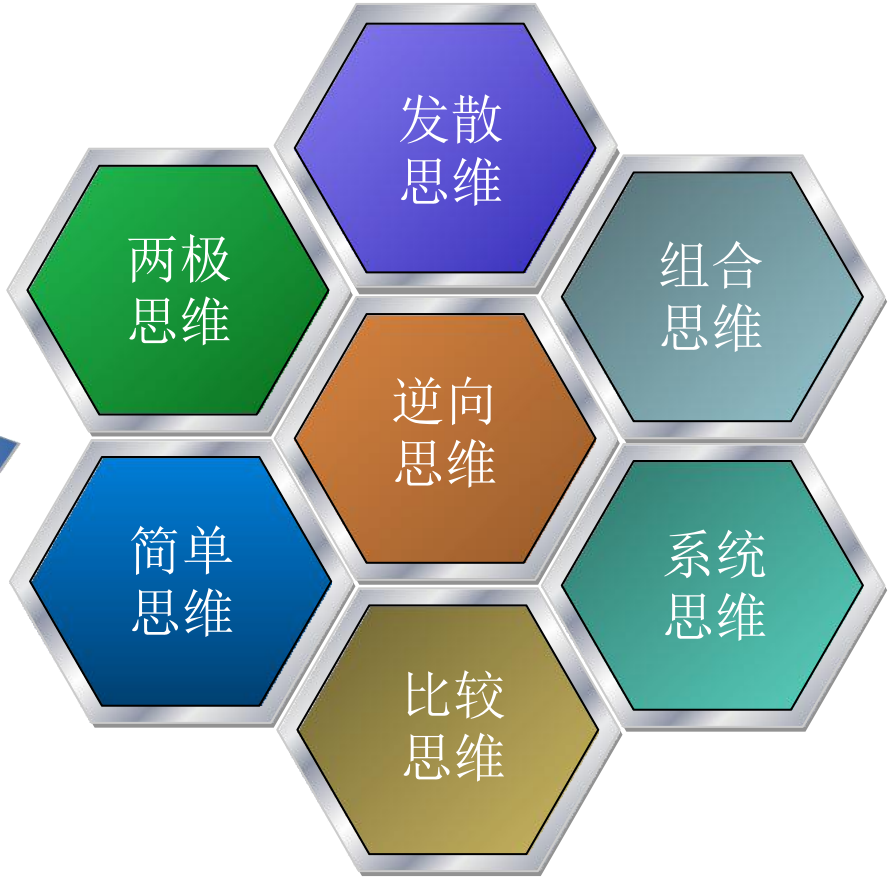
用同样的测试用例多次重复进行测试，最后将不再能够发现新的缺陷。



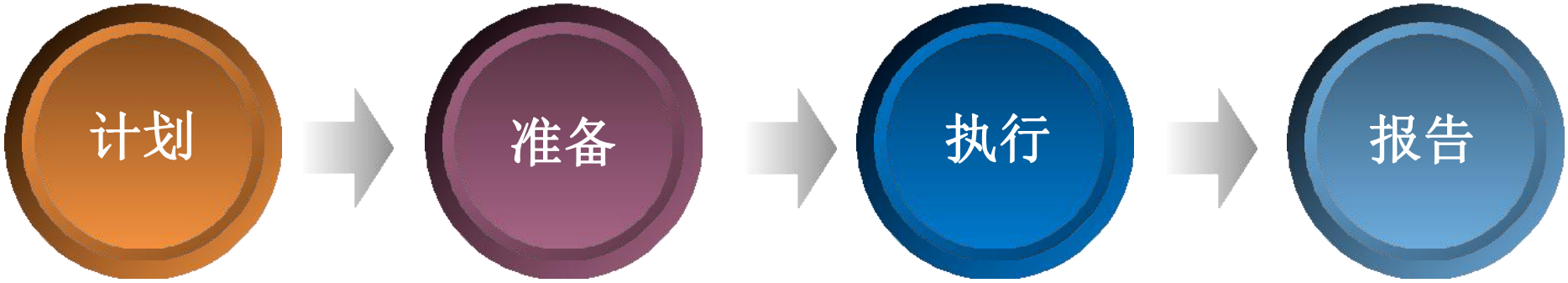
如同给果树喷撒农药，为了杀灭害虫只打一种杀虫药，虫子就会有抗体而变得适应，于是杀虫剂将不再发挥作用。

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



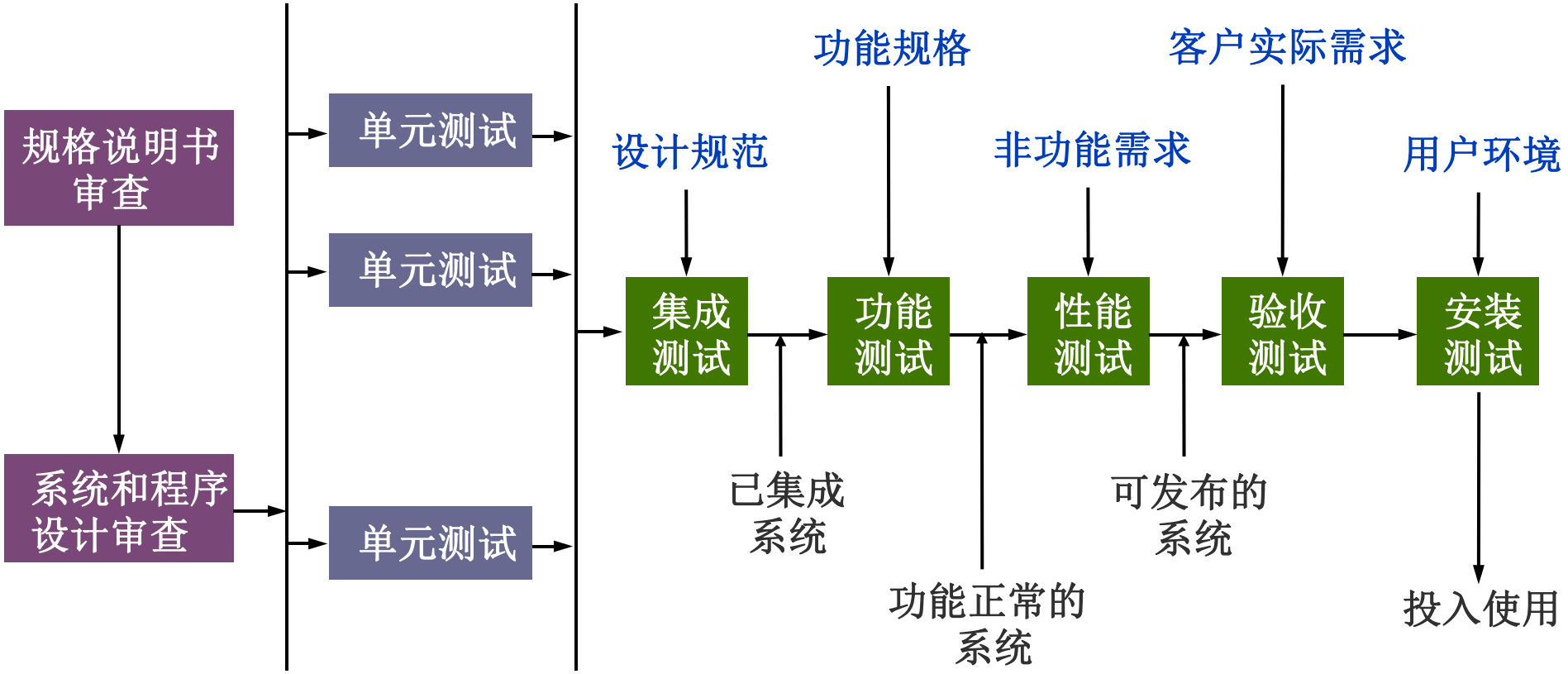
- 识别测试需求
- 分析质量风险
- 拟定测试方案
- 制定测试计划

- 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


规格说明书审查
系统和程序设计审查
单元测试
单元测试
单元测试
设计规范
功能规格
非功能需求
客户实际需求
用户环境
集成测试
功能测试
性能测试
验收测试
安装测试
已集成系统
功能正常的系统
可发布的系统
投入使用

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

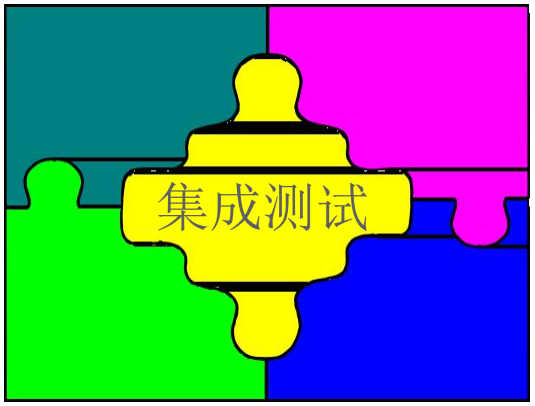
单元测试

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。



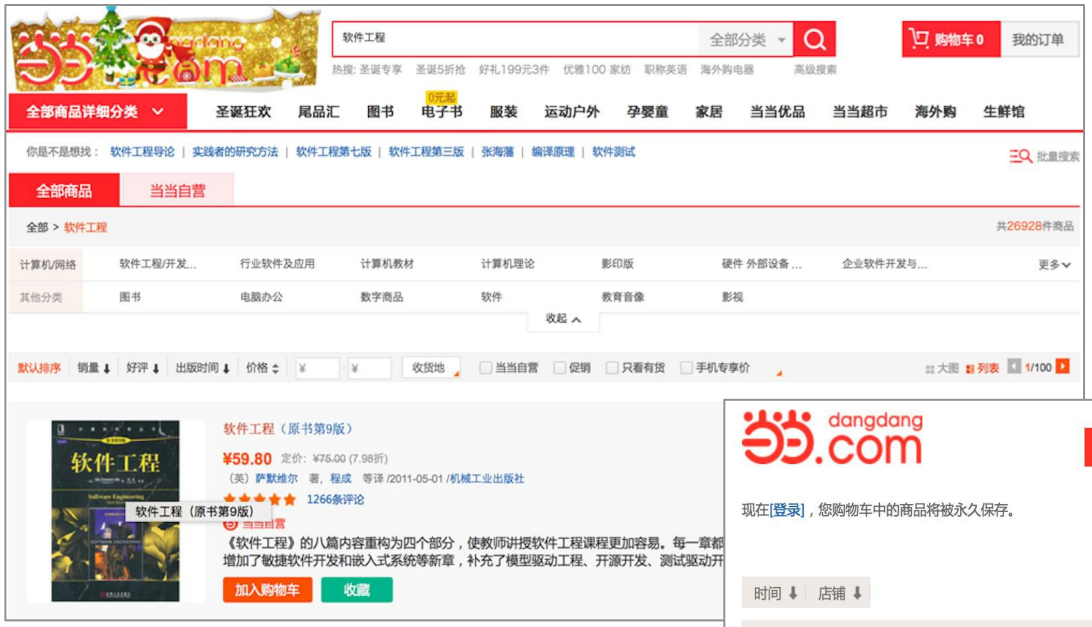
- **一次性集成方式：**分别测试每个单元，再一次性将所有单元组装在一起进行测试。
- **渐增式集成方式：**先对某几个单元进行测试，然后将这些单元逐步组装成较大的系统，在组装过程中边连接边测试。

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。

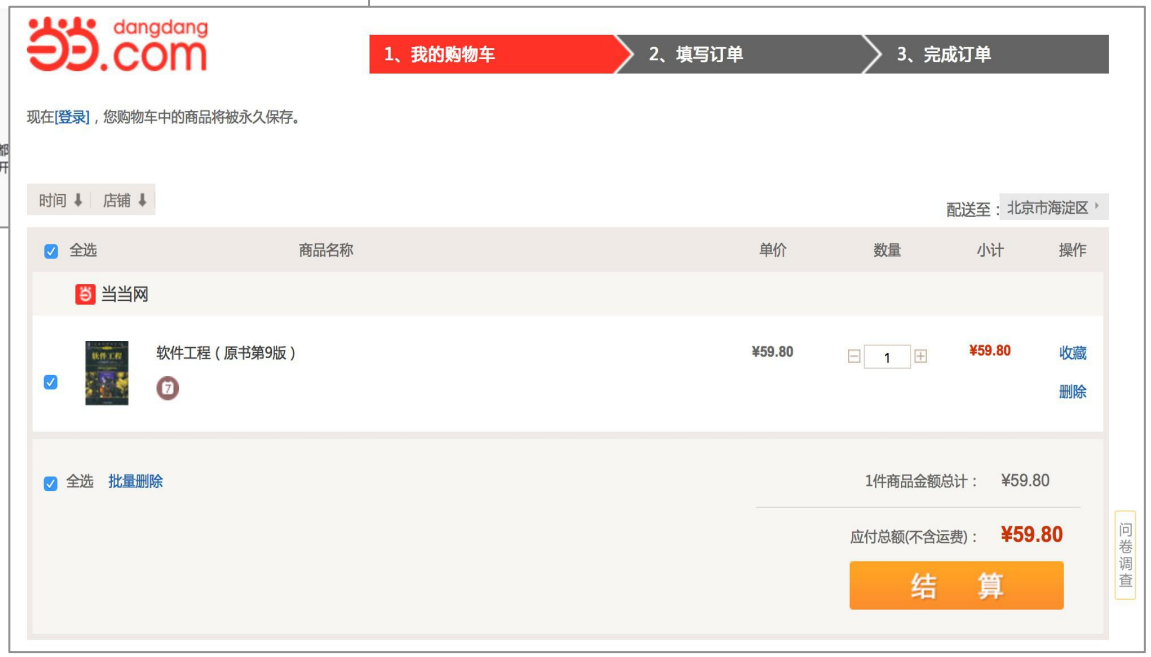
功能测试

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



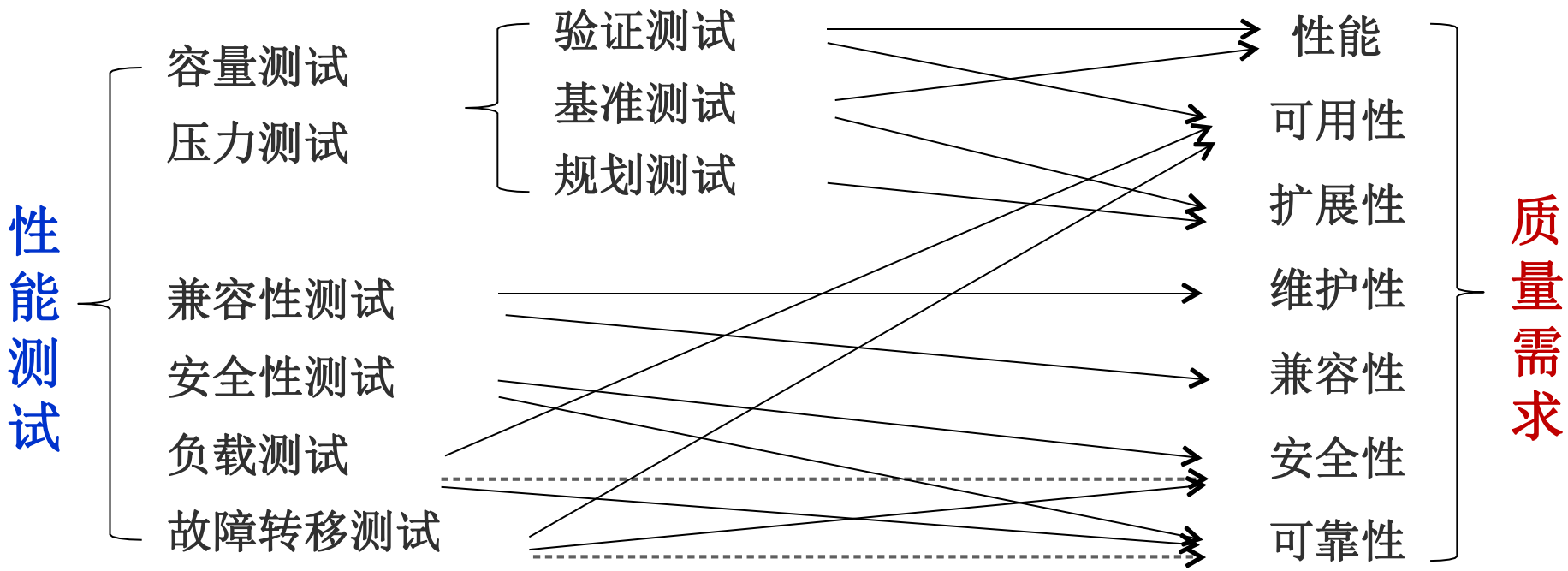
网上搜索并选购图书



当当网
dangdang.com

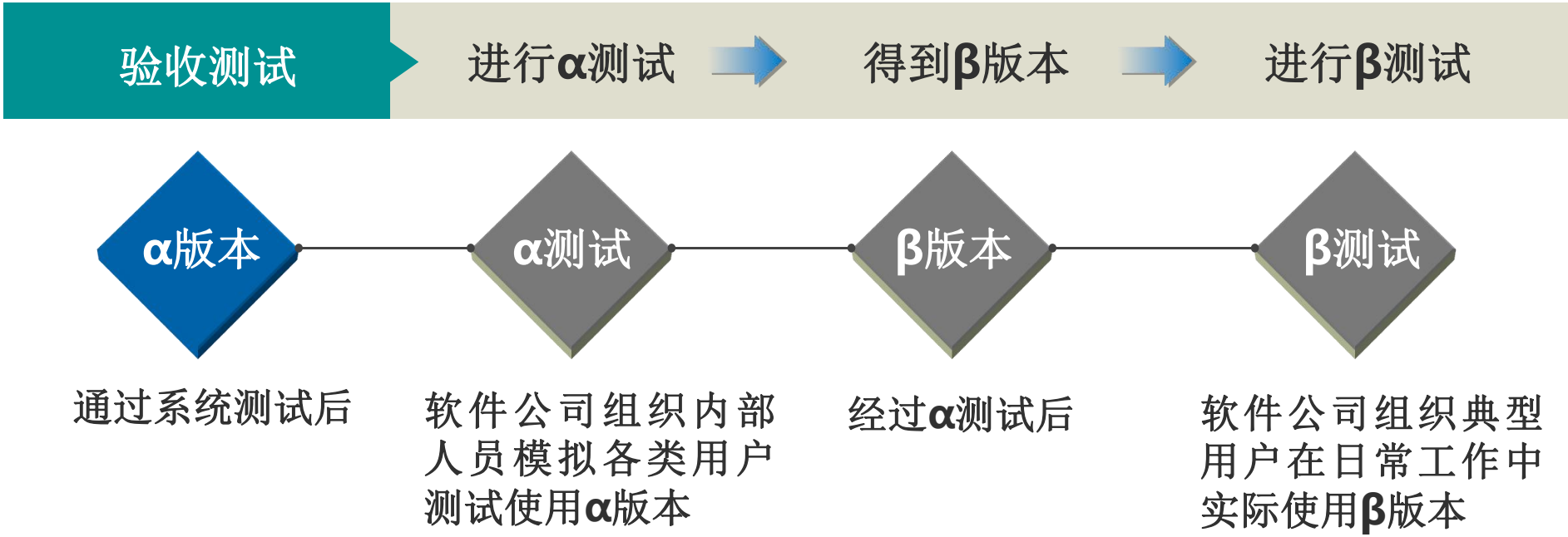
性能测试

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。



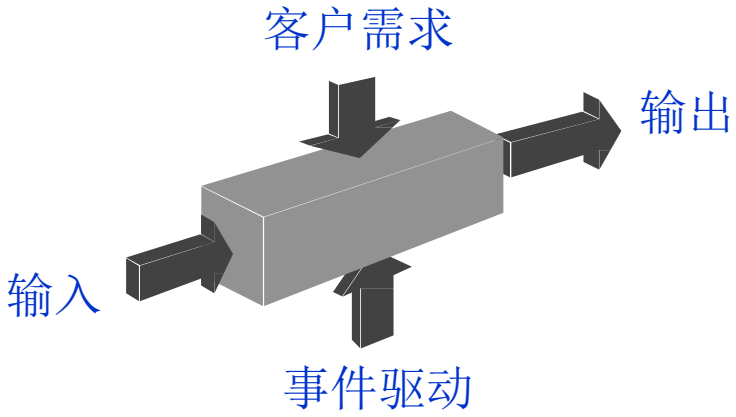
安装测试

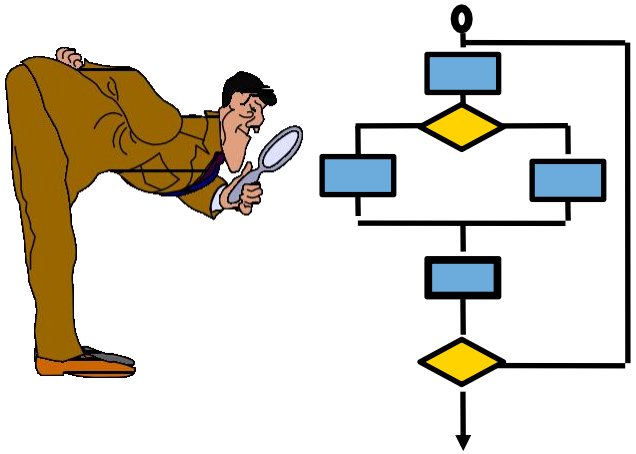
安装测试是系统验收之后，需要在目标环境中进行安装，其目的是保证应用程序能够被成功地安装。

- 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？

黑盒测试和白盒测试

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。

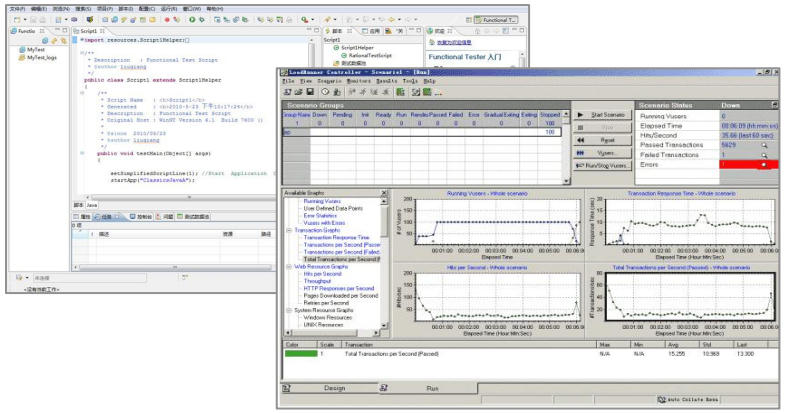
非正式 ——— 正式

互审　　走查　　审查会议

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

测试计划

测试用例

测试

结果

缺陷

缺陷报告

测试报告

软件测试计划


软件测试计划编写要素
Why：为什么要做这些测试
What：测试内容，不同阶段的工作内容
When：不同阶段的起止时间
Where：相应文档，缺陷的存放位置，测试环境等
Who：测试人员及安排
How：如何去做，需要用到的工具及测试方法

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