

5.2 软件测试

- 软件测试概述

- 软件缺陷术语

- 术语解释

- 错误

- 软件生存期内不希望或不可接受的人为错误，导致软件缺陷的产生

- 缺陷

- 软件缺陷是存在于软件产品之中的不希望或不可接受的偏差，其结果是软件运行于某一特定条件时出现故障

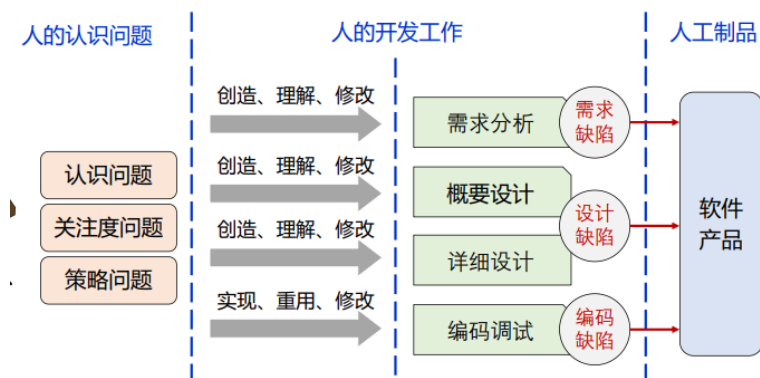
- 故障

- 软件运行过程中出现的一种不希望或不可接受的内部状态，若无适当措施加以处理，产生软件失效

- 失效

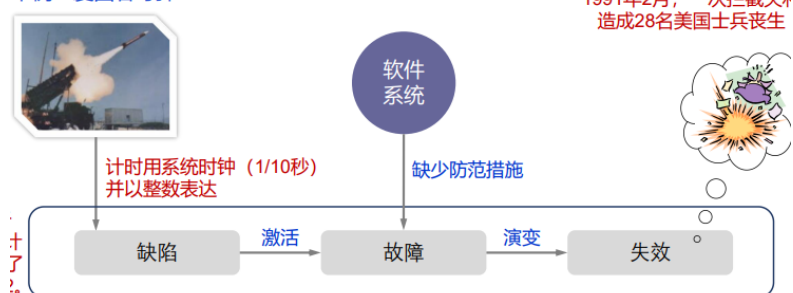
- 软件运行时产生的一种不希望或不可接受的外部行为结果

- 软件缺陷的产生



- 软件缺陷的演化

举例：爱国者导弹



- 软件测试概念

测试是使用**人工和自动手段**来运行或检测某个系统的过程，其目的在于检验系统**是否满足规定的需求**或**弄清预期结果与实际结果之间的差别**

- 两种思路

- 正向思维：验证软件正常工作

- 逆向思维：假设软件有缺陷

- 软件测试目的

- 直接目标：发现软件错误
- 期望目标：检查系统是否满足需求
- 附带目标：改进软件过程

- 测试的局限性

- 测试的不彻底性

测试只能说明错误存在，不能说明错误不存在

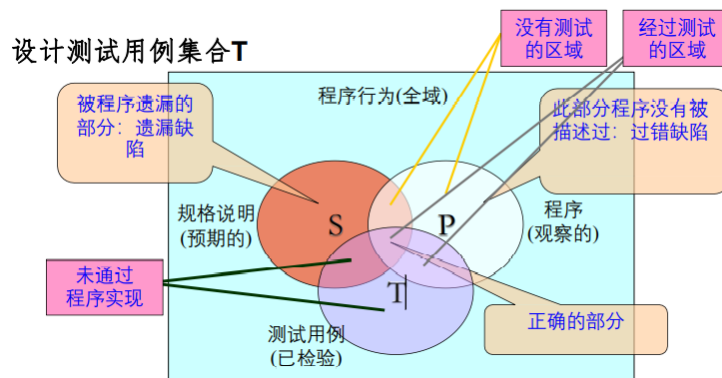
- 测试的不完备性

测试无法覆盖到每个应该测试的内容，不可能测试到全部输入、相应和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（软件质量依靠开发），测试通过早期发现缺陷并督促修正来间接提高软件质量

- 维恩图来理解测试



- 软件测试基本原则

- 必需有对预期输出或结果的定义
- 编程人员应避免测试自己编写的程序
- 编写软件组织不应测试自己编写的程序
- 彻底检查每个测试的执行结果
- 测试用例不仅根据有效和预期输入情况，也应根据无效和未预知的输入情况
- 检查程序是否未做到应该做的，以及做了不应该做的
- 避免测试用例用后即弃
- 计划测试工作时不应默认不会发现错误
- 程序某部分存在更多错误的可能性，与该部分已经发现错误的数量成正比
- 软件测试是一项极富创造力、极具智力挑战性的工作

- 缺陷的集群性

80/20原则：80%的软件错误存在于20%的代码行中

- 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。

- 测试用例需要定期评审和修改，同时不断增加新的不同测试用例来测试软件不同部分

- 软件测试团队

- 软件测试团队的任务

- 发现软件程序、系统或产品中所有的问题
 - 尽早发现问题
 - 督促开发人员尽快地解决程序中的缺陷
 - 帮助项目管理人员指定合理的开发计划
 - 对问题进行分析、分类总结和跟踪
 - 帮助改善开发流程，提高产品开发效率
 - 提高程序编写的规范性、易读性、可维护性等

- 测试人员角色

- 测试经理

- 建立和完善测试流程以及部门管理体制，审核测试项目并分配资源，监控和协调各项目的测试工作，负责与其他部门的协调和沟通工作。

- 测试组长

- 制定测试项目计划（包括人员、进度、软硬件环境和流程等），实施软件测试，跟踪和报告计划执行情况，负责测试用例质量，管理测试小组并提供技术指导

- 测试工程师

- 理解软件产品的要求，对其进行测试以便发现软件中的错误，验证软件是否满足规格说明所描述的需求，编写相应的测试方案和测试用例

- 测试工具工程师

- 编写软件测试工具，并利用所编写的测试工具对软件进行测试，或者为测试工程师开发测试工具

- 软件测试人员应具备素质

- 测试技术能力
 - 非技术能力
 - 专业领域经验

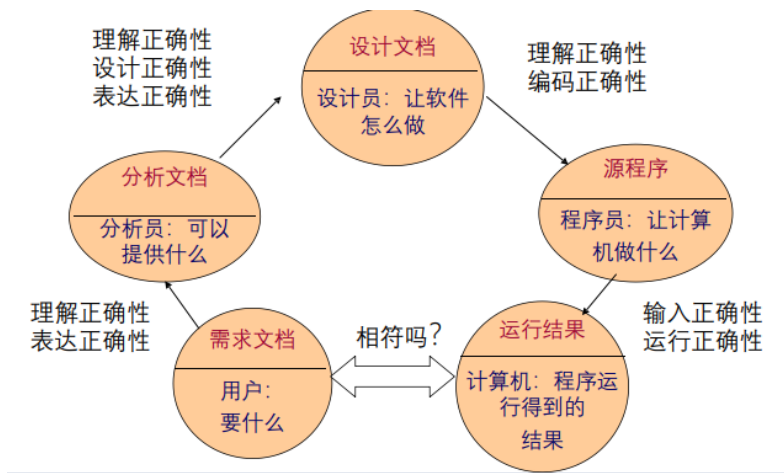
- 软件测试策略

- 软件测试对象

- 测试对象

- 需求规格说明
 - 设计规格说明
 - 源程序

- 软件测试对象之间的关系

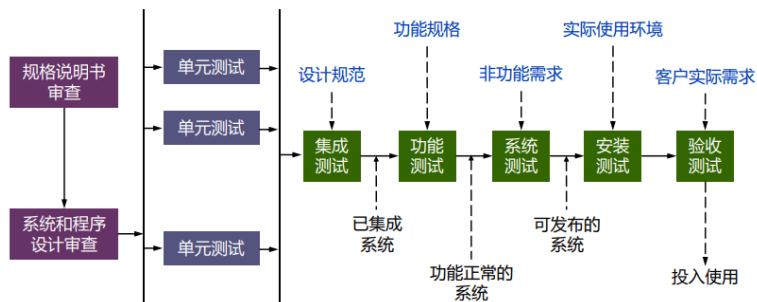


• 软件测试过程

• 测试过程



• 软件测试阶段



• 软件测试类型

• 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、白盒测试

• 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

对软件基本组成单元进行的测试。一般由编写该单元代码的开发人员执行。

• 目的

• 单元测试环境

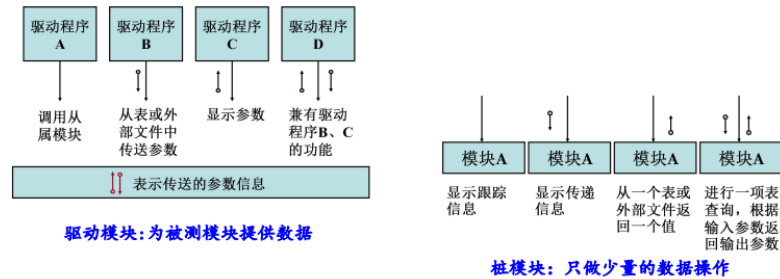
- 驱动模块

模拟被测模块的上一级模块，接收测试数据，将数据传送给所测模块

- 桩模块

模拟被测单元需调用的其他函数接口，模拟实现子函数的某些功能

- 图像表示



- 集成测试

在单元测试的基础上，将所有模块按照总设计的要求组装成为子系统或系统进行的测试

- 整体集成方式

把所有模块按照设计要求一次全部组装，然后进行整体测试

- 优点

效率高、人力少、用例数目少、简单

- 缺点

可能发现大量错误难以定位修改、可能遗漏较多错误、测试和修改过程中新旧错误混杂，调试困哪

- 渐增式集成方式

- 自顶向下的增量测试

从主控模块开始，按照软件控制层次结构，以深度优先或广度优先的策略，逐步集成各个模块

- 优点

尽早测试程序主要控制和决策机制，较早发现错误，较少需要驱动模块

- 缺点

桩模块数量较多，桩模块无法反映真实情况，重要数据不能及时返回上层模块，测试不充分

- 自底向上的继承测试

从软件结构最底层模块开始组装测试

- 优点

不用桩模块，测试用例设计简单

- 缺点

最后一个模块加入时才有整体形象，难以建立信心

- 系统测试

在实际运行环境或模拟实际运行环境，针对系统非功能特性进行的测试，包括负载测试、性能测试、压力测试、恢复测试、安全测试和可靠性测试

- 恢复测试

恢复测试是检验系统从软件或者硬件失败中恢复的能力，即采用各种人工干预方式使软件出错，而不能正常工作，从而检验系统的恢复能力

- 安全性测试

安全性测试检查系统对非法侵入的防范能力

- 压力测试

压力测试是检查系统在资源超负荷情况下的表现，特别是对系统的处理时间有什么影响

- 性能测试，常与压力测试一起进行

- 验收测试

软件产品完成了**功能测试和系统测试之后、产品发布之前**进行的软件测试活动验证软件的功能和性能能否满足用户期望的要求。**有 和 版本**

- 回归测试

在软件生命周期中的任何一个阶段，只要软件发生了改变，就可能给该软件带来问题。为了**验证修改的正确性及其影响**就需要进行回归测试。

- 黑盒测试

又称**功能测试**，它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

又称**结构测试**，它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

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**动态分析或程序测试**的方式来确认程序正确性。

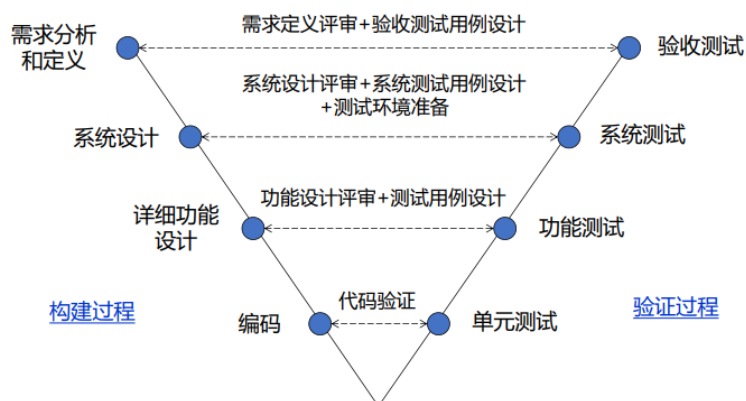
- 手动测试

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、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。

- V 模型



- 测试用例

- 定义

测试用例是为特定的目的而设计的一组**测试输入、执行条件和预期的结果**。

测试用例是执行的**最小测试实体**。

测试用例就是设计一个**场景**，使软件程序在这种场景下，必须能够正常运行并且达到程序所设计的执行结果。

- **特征**

最有可能**抓住错误**的；

不是重复的、多余的；

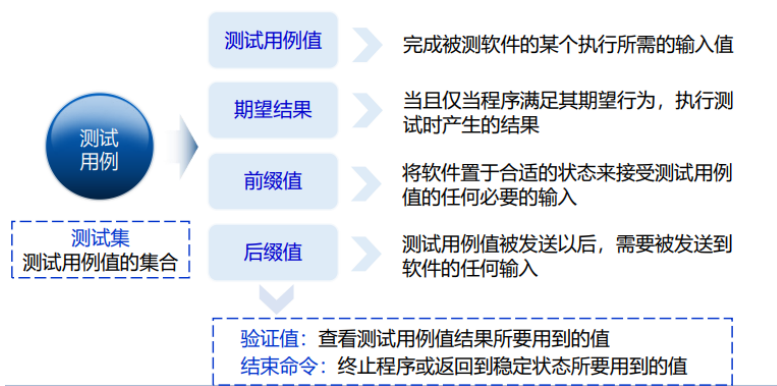
一组相似测试用例中**最有效的**；

既不是太简单，也不是太复杂

- **重要性**

- **指导人们系统地进行测试**
- **有效发现缺陷，提高测试效率**
- **作为评估和检验的度量标准**
- **积累和传递测试的经验与知识**

- **内容**



- **设计要求**

- **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**