软件测试的定义

**使用人工或自动化手段，来运行或测试某个系统的过程，其目的在于检验它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。**

软件测试的目的

**根本目的：发现问题、 解决问题；提高用户满意度、优化软件品质**

测试的对象：**程序 文档 数据** (软件开发过程中所产生的需求规格说明、概要  
设计规格说明、详细设计规格说明以及源程序、用户文档都是软件测试的对象。 )

测试的原则

**1、尽早地和及时地进行测试（在需求阶段开始测试）**

**2、测试前应当准备好测试数据和与之对应的预期结果这两部分**

**3、测试输入数据应包括合理的输入条件和不合理输入条件**

**4、程序提交测试后，应当由专门的测试人员进行测试**

**5、严格执行测试计划，排除测试的随意性**

**6、测试用例的所有相关预期结果做全面的检查**

7、充分注意测试当中的群集现象（二八原则）

8、注意保存测试计划、测试用例、出错统计和最终分析报告，为维护工作提供充分的资料

9、缺陷具有免疫性（每修复3~4个缺陷就会出现一个新的缺陷）

测试的误区（**避免误区**）

1、软件测试技术要求不高，至少比编程容易多了

2、若发布的软件有质量问题，那是软件测试人员的错

3、软件测试是测试人员的事，与开发人员无关

4、根据软件开发瀑布模型，软件测试是开发后期的一个阶段

5、有时间就多测试一些，来不及就少测试一些

6、软件测试是非建设性的工作，甚至是破坏性的

7、测试中发现错误是对责任人工作的一种否定

软件开发模型

1. 大棒开发法：思路简单，可能是开发者的突发奇想，结果不可预知
2. 边写边改法：

优点：简单考虑到了软件的需求，产品周期短

缺点：没有计划和文档的编制，后续维护难度大

测试工作： 由于新的版本不断产生，测试工作长期循环

1. 快速原型法：根据客户需求在较短的时间内解决用户最迫切解决的问题，完成可演示的产品。这个产品只实现最重要功能，在得到用户的更加明确的需求之后，原型将丢弃。
2. **瀑布模型：**

阶段：需求分析-系统设计-程序设计-编码-测试-运行及维护

特点：依线性顺序联接

优点：易理解 阶段性 强调需求分析 明确测试阶段 提供了一套模板

缺点：

线性严格——成果晚出——风险

阶段固定——反复&迭代——灵活性

单次需求——需求变更——适应性

测试滞后——缺陷晚查——代价

适用场合：功能性能明确完整 需求固定 无重大变动

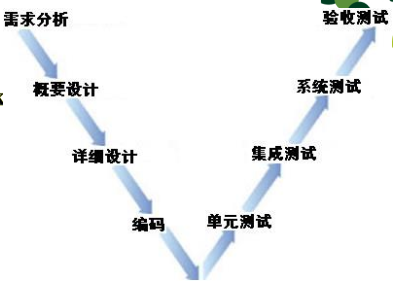
1. 螺旋模型法

步骤：评估可选方案--计划下一阶段--确定下一阶段方法--确定目标、可选方案和限制条件--明确并化解风险--当前阶段开发和测试

优点：严格的全过程风险管理；强调各开发阶段的质量；提供机会评估项目是否有价值继续下去。(发现问题早)

软件测试模型

1. V模型（是最具有代表意义的测试模型）



优点：

测试级别清晰（底层：单元测试 高层：系统测试）

测试对应开发

缺点：

线性执行——测试滞后编码 关注程序——忽略需求、设计 需求变更——实用性差

1. W模型（尽早和不断的进行软件测试）

优点：测试伴随着整个软件开发周期，而且测试的对象不仅仅是程序，需求、 功能、设计同样要测试。

缺点：1、在小的项目里，W模型不适用 2、不支持迭代，应对需求变化方面不适用

软件测试流程

1. 制定测试流程 2、设计和生成测试用例 3、搭建测试环境

4、实施测试 5、测试评估 6、测试总结

测试用例相关

1、解决的问题：要测什么 怎么测 如何衡量

2、静态测试：不运行程序，只对程序进行检查和审核 动态测试：使用和运行程序进行检查

3、黑盒测试

等价类划分

边界值分析（通常作为等价类划分的补充）原则：刚刚等于 刚刚大于 刚刚小于

常用控件法：复选框 组合列表框 文本框 按钮 单选按钮 列表框 滚动条（重点参考2-5PPT）

错误推测法 借助经验开展测试，推测软件中可能存在或容易产生的缺陷的功能模块

（经验分享 在2-6PPT）

例题：测试一个对线性表（比如数组）进行排序的程序，可推测列出以下几项需要特别测试的情况：、

1. 输入的线性表为空表；

2. 表中只含有一个元素；

3. 输入表中所有元素已排好序；

4. 输入表已按逆序排好；

5. 输入表中部分或全部元素相同

4、白盒测试

·静态测试

1. 代码检查法

a.桌面检查(程序员自己检查自己编的代码，目的是发现错误)

b.代码走查

C.代码审查（逐条讲解）

1. 静态结构分析法

使用静态结构分析工具分析程序源代码的内部结构（如：系统结构、数据结构、数据接口、内部控制逻辑）

1. 代码质量度量
2. 代码行度量法（**千行bug率**）
3. McCabe度量法 **V(G)=m-n+2（弧数减节点数加二）或等于判断语句数+1**
4. Halstead度量法

程序=运算符号+运算对象结构度量

**N=N1+N2=n1log2(n1)+n2log2(n2)**（n1=不同运算符的个数 也即– + - \* / = if else for …… n2=不同运算对象的个数）

1. 结构度量法

扇入：调用该模块的模块计数

扇出：该模块所调用的模块计数

**具有大扇入和大扇出的模块可能是不良设计。程序复杂性与扇出的平方成正比**

·动态测试（重点PPT2-9）

1. 逻辑覆盖
2. 语句覆盖
3. 判定覆盖
4. 条件覆盖
5. 单元测试

单元：一个最小的单元应有明确的功能、性能定义、接口定义而且可以清晰地与其他单元区

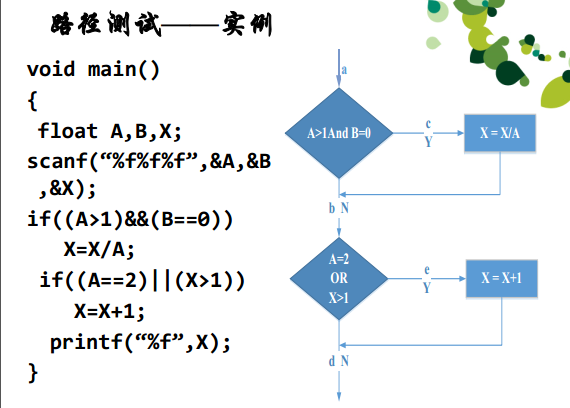
分开来。

单元测试：

1. 定义：单元测试又称为模块测试, 是针对程序模块（软件设计的最小单位）来进行正确性检验的测试工作。程序单元是应用的最小可测试部件
2. 目标：确保每个模块都能正常工作
3. 时间：编码--编译--单元测试
4. 依据：详细设计说明
5. 单元测试的内容：（详见PPT 4.1 page 24）
6. 模块接口 ：检查进出程序单元的数据流是否准确（在其他测试之前进行）也即检查输入参数与模块的形式参数在个数、属性、顺序上是否匹配
7. 局部数据结构：如类型说明是否一致，初始化或默认值是否一致、溢出检测、拼写错误的变量名等
8. 边界条件：可能与边界值有关的数据类型，如字符、数值、位置等如：运算或判断中取最大值、最小值时是否有错误。在n次循环的第0次、 1次、 n次是否有错误。数据流、控制流中刚好等于、大于、小于确定的比较值是否出现错误
9. 出错处理：关注模块在工作中发生错误时，出错处理设施是否有效。如：HTTP500错误：web服务器不能执行此请求 HTTP404 找不到您所请求的文本或脚本。

路径测试：

1. 定义：针对程序路径进行测试，关注由于计算错误、判定错误或不正常的控制流二产生的错误。
2. 实例：



软件缺陷相关

1、什么是缺陷

从产品内部看：软件缺陷是软件产品开发或维护过程中所存在的错误、毛病等各种问题；

从产品外部看：软件缺陷是系统所需要实现的某种功能的失效或违背。

**2、如何编写缺陷报告**（一个缺陷一个报告）

**举例**：使用“记事本”仅保存“联通”二字后再打开该文件，出现乱码。

•描述步骤：

–点击“开始” ->“程序” ->“附件” ->“记事本”打开记事本软件；

–仅输入“联通”二字后，点击“文件” ->“保存”；

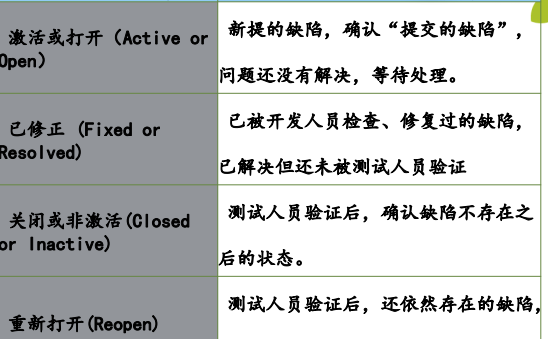
–在打开的“另存为”对话框中保存文件， 退出（文件名、保存位置任意）；

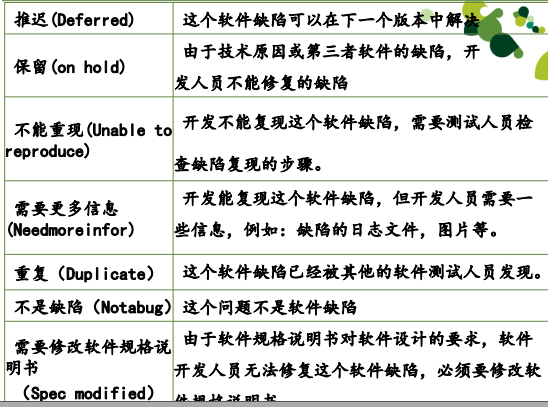
–打开保存的文件，出现乱码，不是“联通”二字。

1. 缺陷的严重性和优先级
2. Fault 致命的错误 造成系统或应用程序崩溃，主要功能完全丧失
3. Critical：严重错误，主要指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
4. Major：主要错误，这样的缺陷虽然不影响系统的使用，但没有很好地实现功能。
5. Minor: 一些小问题，不影响功能
6. Suggestion：一些友好的建议

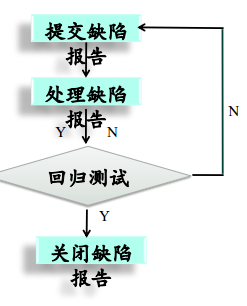
**一般情况下，严重程度高的缺陷优先级高，班没有必然联系，需结合实际综合思考**

1. 缺陷状态
2. 定义：缺陷通过一个跟踪修复过程的进展情况，也就是在缺陷生命周期中的状态基本定义
3. 缺陷状态：





1. 缺陷状态流转



易用性测试：易用性是交互的适应性、功能性和有效性的集中体现

1. 界面（UI）

标准和规范

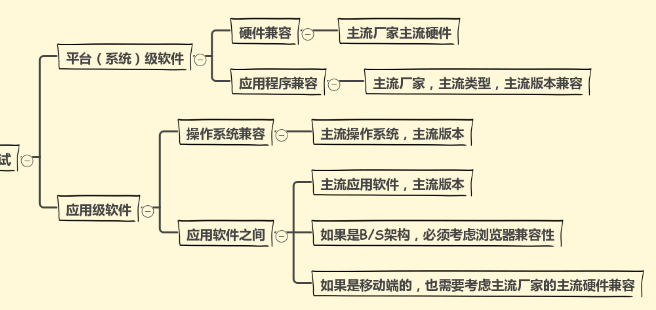
直观 正确 一致（如果软件或者平台有统一标准，遵循标准。如果没有，就注意软件的特性，确保相似操作以相似的方式进行） 灵活（状态终止和跳转、数据输入和输出） 舒适 实用

1. 辅助功能选项（为有残疾障碍的人测试）
2. 操作系统内置
3. 平台
4. 为什么要进行易用性测试：软件难以理解，不易使用，运行缓慢，或者从测试人员或用户的角度认为不好。

兼容性测试：指软件之间能否正确的交互和共享信息。

1. 兼容性测试标准
2. 向前兼容（可以使用软件以前的版本 如：Office 2007/2013 可以打开Office 2003下存储的文档）
3. 向后兼容（可以使用软件以后的版本 如：Office 2003可以打开Office2013存储的文档）

2、



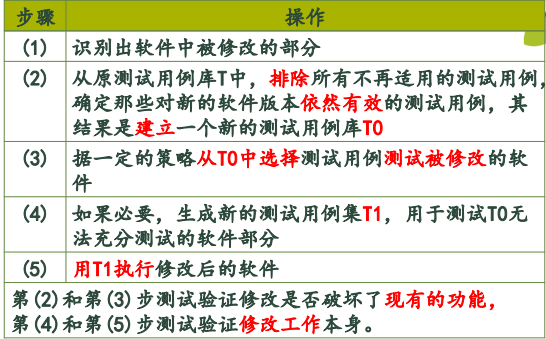
1. 数据兼容性指要在应用程序之间共享数据

安全性测试

1. 定义：检查系统对非法侵入的防范能力
2. 原因：验证安装在系统内的保护机制能否在实际中保护系统且不受非法入侵， 不受非法干扰。（应对入侵）
3. 如何做安全性测试：
4. 代码角度
5. 使用安全函数
6. 防止缓冲区溢出
7. 外部攻击角度
8. 跨站脚本攻击
9. SQL注入（通过把SQL命令插入到Web表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令）如：穿过防火墙，入侵检测系统；执行增、删、改、查脚本；尝试输入特殊字符

回归测试

1. 定义：验证被修复软件缺陷是否已经被解决，并且保证以前运行的功能依旧保持正常，而不受修改的影响
2. 测试点
3. 测试修改的模块
4. 有可能会受到影响的模块



验收测试

1. 定义：对软件的新的版本测试时，对新版本进行重新测试，这时的测试不仅是验证被修复的软件缺陷是否被解决了，且要保证以前所有运行正常的功能依旧保持正常，而不要受到这次修改的影响
2. 验收测试依据：软件需求说明文档
3. 谁来做验收测试
4. 产品软件验收
5. 项目软件验收
6. 验收测试的实施
7. α测试：开发公司组织内部人员进行测试
8. β测试：内测之后的公测，组织外部典型性用户进行测试

软件质量相关（见）

1. 质量定义范围-3A
2. 可说明性
3. 有效性
4. 易用性

2、软件质量标准

1. 产品质量模型（人们实践产物的属性和行为）：McCall ISO9126 Boehm
2. 过程质量模型（人们产生产品的过程的好坏）: CMM ISO9000
3. SQA

QA职责：（过程管理和过程改进） QC职责（发现软件问题）