**一、软件质量**

**1.软件 = 程序(数据) + 文档 + 服务**

**软件产品组成部分**：程序代码 、 帮助文件 、 用户手册、 样本和示例 、 标签 、产品支持信息、图表和标志、错误信息、广告与宣传材料、软件的安装、软件说明文件、测试错误提示信息。

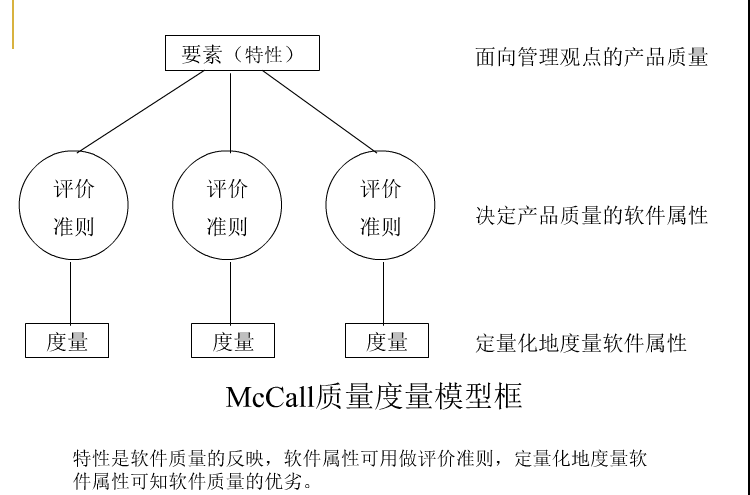
**2.软件开发过程**:需求分析(可行性报告， 项目初步开发计划，需求规格说明，用户手册概要，测试计划);设计(**概要**：建立系统总体结构，划分功能模块；定义各功能模块接口；数据库设计；指定组装测试计划.**详细**:设计各模块具体实现算法;确定模块间的详细接口;指定模块测试方案)[设计说明书,测试计划]**编码**(编程, 进行模块调试和测试,编写用户手册)[调试报告, 用户手册]**测试**、**维护**(纠错、适应、增强、预防)。

**3.软件测试与软件开发过程的关系：**

1）项目规划阶段：负责从单元测试到系统测试的整个测试阶段的监控。2）需求分析阶段：确定测试需求分析、 系统测试计划的制定， 评审后成为管理项目。3）详细设计和概要设计阶段：确保集成测试计划和单元测试计划完成。4）编码阶段：由开发人员进行自己负责部分的测试代码。 在项目较大时， 由专人进行编码阶段的测试任务。5）测试阶段（ 单元、 集成、 系统测试） ：依据测试代码进行测试，并提交相应的测试状态报告和测试结束报告。

**4.软件质量的定义:** 与软件产品满足规定的和隐含需求的能力有关的特性

**5.McCall质量模型**包括质量要素、准则、度量三层次:

**1)** **McCall质量模型软件质量要素（11个分为3类）**

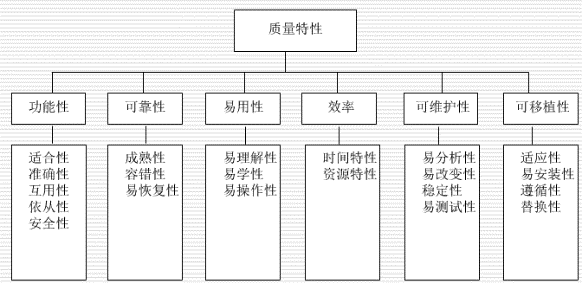
1.产品修正：可维护性、可测试性、灵活性

2.产品转移：互联性、可移植性、复用性

3.产品运行：正确性、可使用性、完整性、可靠性、效率

**2) 质量要素评价准则** 1）可审查性 2）准确性 3）通信通用性 4)完全性 5）简明性 6）一致性 7）数据通用性 8）容错性 9）执行效率 10）可扩充性 11）通用性 12）硬件独立性 13）检测性 14）模块化 15）可操作性 16）安全性 17）子文档化 18）简单性 19）软件系统独立性 20）可追踪性 21）易培训性**3)软件质量的度量:** 1）确定软件质量需求 2）确定度量 3）分析度量结果 4）确认质量度量

**6.ISO-9126软件质量模型：**包括SQRC(软件质量需求评价准则)**、**SQDC(软件质量设计评价准则)**、**SQMC(软件质量度量评价准则)，对应于McCall质量模型的质量要素、准则、度量。

**1) 8个要素**：正确性、可容性、效率、安全性、可用性、可维护性、适应性、连接性 **2）23个评价准则**：和McCall一样

**3）6个质量特性(左图)**  **4) 质量特性使用**：定义软件质量需求、评价软件产品。软件质量观点（用户、开发者、管理者）**5）质量评价。目的：**1.确定产品是否通过验收 2.与其他类似产品相比较选择 3.评估产品正面和负面影响 4.确定何时优化。 **步骤：**1.质量要求定义：据软件需求定义软件质量特性和可能的子特性，将用户的质量要求转化为软件开发不同阶段的质量要求，并及时分解为软件产品组成部分的质量要求。2.评价准备：（1）选择质量度量；（2）定义等级； （3）定义评估准则；3.评价过程：（1）测量：把选定的度量应用到软件产品上（2）评级 （3）评估。

**7.软件质量管理:**

**1）ISO9000：2000：**本标准表述了质量管理体系的基础。1 质量管理体系的理论说明 2 质量管理体系要求与产品要求 3 质量管理体系方法 4 过程方法 5 [质量方针](http://wiki.mbalib.com/wiki/%E8%B4%A8%E9%87%8F%E6%96%B9%E9%92%88)和质量目标 6 最高管理者在质量管理体系中的作用 7 文件 8 质量管理体系评价 9 持续改进 10 统计技术的作用 11 质量管理体系与其他管理体系的关注点 12 质量管理体系与优秀模式之间的关系

**2）CMM(能力成熟度模型)：**

**二、软件测试**

**1.软件缺陷：**

**1）定义：**软件出错机理可描述为:软件错误，软件缺陷，软件故障，软件失效。

1. 软件错误(error) ：是指在软件生存期内的不希望或不可接受的人为错误，其结果是导致软件缺陷的产生。2. 软件缺陷(bug) ：是存在于软件（文档、数据、程序）之中的那些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软件故障，这时称软件缺陷被激活。3. 软件故障(fault) ：是指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。此时若无适当措施（容错）加以及时处理，便产生软件失效。4. 软件失效(failure) ：是指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。

**2）分类**：可从不同角度对软件缺陷进行分类：

（1）按错误的影响和后果分类 ：小、中等、较严重、非常严重、最严重

（2）按错误的性质和范围分类：功能错误、系统错误、加工错误、数据错误、代码错误

（3）按软件生存期阶段分类：问题定义（需求分析）错误、规格说明错误、设计错误、编码错误

**3）产生**：1用户的计算机知识较少 2要开发产品的特性不够清晰3需求变化的不一致 4对需求说明书不重视5项目组成员间缺少沟通

**2.软件测试的定义：**

IEEE/ANSI标准： 就是在既定的状况条件下，运行一个系统或组建，观察记录结果，并对其某些方面进行评价的过程。

《软件测试技巧》： 软件测试是为了发现错误而运行程序的过程。

广义软件测试的定义：由确认、验证、测试3方面组成。

**3. 软件测试的分类：**

1）按测试过程 （开发阶段）：单元测试、集成测试、确认测试、系统测试、验收测试

（1）单元测试的主要目的是针对编码过程中可能存在的各种错误，例如用户输入验证过程中的边界值的错误。

（2）集成测试主要目的是针对详细设计中可能存在的问题尤其是检查各单元与其它程序部分之间的接口上可能存在的错误。

（3）系统测试主要针对概要设计，检查了系统作为一个整体是否有效地得到运行，例如在产品设置中是否达到了预期的高性能。

（4）验收测试通常由业务专家或用户进行，以确认产品能真正符合用户业务上的需要。

2）按测试用例设计方法：白盒测试、黑盒测试、灰盒测试

3）按实施对象：(1)Alpha测试（企业内部测试）：是由用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的受控测试。(2)Beta测试（最终用户测试）：是软件的多个用户在实际使用环境下进行的测试。(3)第三方测试（独立测试）

4）按执行方式：(1)人工测试：手工执行的测试；(2)自动化测试：希望能够通过自动化测试工具或其他手段,按照测试工程师的预定计划进行自动的测试。如：负载测试、性能测试、可靠性测试等。

5）按测试方式划分：(1)静态测试:静态测试方法的主要特征是在用计算机测试源程序时，计算机并不真正运行被测试的程序，只对被测程序进行特性分析。静态测试常称为“分析”，静态分析是对被测程序进行特性分析的一些方法的总称。(2)动态测试:

动态测试方法的主要特征是计算机必须真正运行被测试的程序，通过输入测试用例，对其运行情况(输入/输出的对应关系)进行分析。

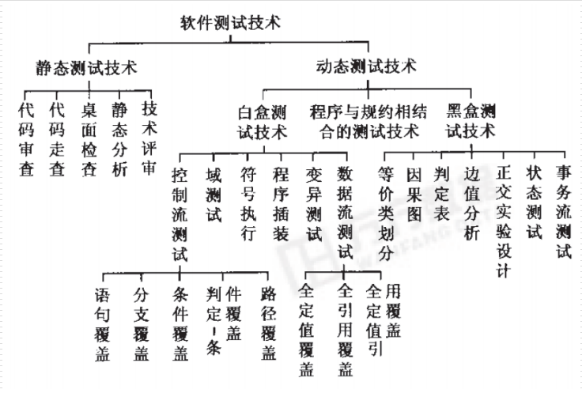
6）按测试形态(Testing Types)：(1) 建构性测试(Construction Testing)：当程序还是处于建设阶段时所进行的测试；是属于前置性的测试,它主要是偏重于程序端的功能测试,以确保程序执行运行正常。(2) 系统测试(System Testing) :是针对系统的行为来做测试；是属于中后期的整合测试,所进行的测试是以使用者的观点为主,也就是模拟外界世界的使用者会如何的使用产品。(3) 特殊测试(Special Testing) :根据产品的本质特性来安排或剔除特殊测试

**3.软件测试的目的**：决定了如何去组织测试。① 软件测试是程序的执行过程，目的在于发现错误；② 测试是为了证明程序有错，而不是证明程序无错误。③ 一个好的测试用例是在于它能发现至今未发现的错误；④ 一个成功的测试是发现了至今未发现的错误的测试。

**4.软件测试的原则：**

1软件开发者的座右铭：“尽早地和不断地进行软件测试” 。2测试用例应由测试输入和与之对应的预期输出结果两部分组成。

3程序员应避免检查自己的程序。（注意不是指对程序的调试）。4在设计测试用例时，应当包括合理的输入条件和不合理的输入条件。不合理的输入条件是指异常的，临界的，可能引起问题异变的输入条件。5妥善保存测试计划，测试用例，出错统计和最终分析报告，为维护提供方便。6测试无法显示潜在的软件缺陷和故障。7通过测试只能报告已被发现的缺陷和故障，无法报告隐藏的软件故障。8完全测试是不可能的，测试需要终止。9软件测试是有风险的行为，要针对风险做出抉择。10充分注意测试中的群集现象。11严格执行测试计划，排除测试的随意性。12应当对每一个测试结果做全面的检查。13对测试结果要有一个确认过程。14所有测试的标准都应建立在用户的需求上。15软件测试必须基于“质量第一”的思想去开展各项工作，当时间和质量冲突时，时间服从质量。16并非所有软件缺陷都要修复。

**5.软件测试技术：**1）静态方法2） 动态方法

**静态测试**:只对被测程序进行特性分析(文档评审、代码检查、代码度量) 包括检查单和静态分析方法 测试的内容与选择的测试方法有关

**动态测试**：必须真正运行被测试的程序(主要是白盒为主和黑盒为辅) 通常对软件单元的功能、性能、接口、局部数据结构、独立路径、出错处理、 边界条件和内存使用情况进行测试

**6.软件测试模型--V模型：**

需求分析 验收测试

概要测试 (开发之后测试)系统测试

详细设计 集成测试

编码 单元测试

**V模型价值**：非常明确标明测试过程中的不同级别,并且描述了这些测试阶段和开发过程各阶段的对应关系

**V模型问题**：测试是开发之后的一个阶段、测试的对象是程序本身、易导致需求阶段的错误一直到最后系统测试阶段才被发现

**7.软件测试模型--W模型**

是V模型测试的改进，在概要设计、详细设计和编码每个步骤都要进行检测。主要思想: 尽早地和不断地进行软件测试。

优点：强调了测试计划等工作的先行和对系统需求和设计的测试。

缺点：没有对软件测试流程予以说明

**8.软件测试模型--H模型**

H模型将测试作为一个独立流程，贯穿整个开发周期，与其他流程并行，同时测试准备和测试执行分离。

**特性**：1 测试不仅仅指测试的执行，还包括许多其他活动; 2 测试是一个独立流程，贯穿产品整个生命周期；与其它流程并发进行 3 测试要尽早准备，尽早执行4 测试是根据被测对象的不同而分层进行。

**意义**：1 测试准备和测试执行分离，有利于资源调配，降低成本，提高效率；2 充分体现测试过程（不是技术）的复杂性；3 有组织、有结构化的独立流程，有助于跟踪测试投入的流向

9.**测试用例定义**：（1）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。（2）测试用例是执行的最小实体。

**10.测试用例编写标准**：标识符、测试项、测试环境要求、输入标准、输出标准、测试用例之间的关联。

**11.黑盒测试**：等价类划分方法、边界值分析方法、错误推测方法、因果图方法、判定表驱动分析方法

**12.白盒测试**：逻辑驱动测试（语句覆盖、判定覆盖（分支覆盖）、条件覆盖、判定/条件覆盖、条件组合覆盖）、基本路径测试

**13.常用软件测试技术：**

（1） 单元测试、 集成测试、 确认测试、系统测试、验收测试。

（2） 常用测试方法： 文档测试、 配置测试、 外国语言测试、 易用性测试、 网站测试、 对Web进行压力测试、测试和β测试、 Client/Server 测试、 实时系统测试、面向对象的软件测试。

（3） 调试与测试的区别：目的不同：（1）软件测试的目的是发现错误，至于找出错误的原因和错误发生的地方是调试的任务；调试的目的是为了证明程序的正确。（2）任务不同：测试贯穿于整个开发过程，而调试是编码活动的一部分，主要任务就是排错。（3）指导原则和方法不同。（4）操作者不同

自动化测试：

**14.自动化测试的定义**：一般是指软件测试的自动化。自动化测试可理解为测试过程自动化和测试结果分析自动化，包括测试活动的管理与实施、测试脚本的开发与执行。

15. **自动化测试的意义：**软件测试工作量大，重复性高

16. **自动化测试的原理和方法：**软件测试自动化实现的基础、软件测试自动化实现的原理和方法（主要有：直接对代码进行静态和动态分析、测试过程的捕获和回放、测试脚本技术、虚拟用户技术和测试管理技术。）

**1）代码分析：**代码分析类似于高级语言编译系统，一般针对不同的高级语言去构造分析工具，在工具中定义类、对象、函数、变量等定义规则、语法规则；在分析时对代码进行语法扫描，找出不符合编码规范的地方；根据某种质量模型评价代码质量，生成系统的调用关系图等。

2）**捕获和回放**（代码分析是一种白盒测试的自动化方法，捕获和回放则是一种黑盒测试的自动化方法）：**捕获时**将用户每一步操作都记录下来所有的记录转换为一种脚本语言所描述的过程，以模拟用户的操作；**回放时**，将脚本语言所描述的过程转换为屏幕上的操作，然后将被测系统的输出记录下来同预先给定的标准结果比较。

**3）脚本技术**：脚本是一组测试工具执行的指令集，也是计算机程序的一种形式。可通过录制测试的操作产生，然后再做修改，这样可减少脚本编程的工作量。包括：线性脚本、结构化脚本、共享脚本、数据驱动脚本、关键字驱动脚本。

**4）自动比较**: 1)静态比较(测试过程中不比较，将结果存入数据库)和动态比较 2) 简单比较(允许误差)和复杂比较

**17. 自动化测试存在的一些问题和限制：**(1)不能取代手工测试 (2)手工测试比自动测试发现的缺陷更多 (3)对测试质量的依赖性极大 (4)测试自动化不能提高有效性 (5)测试自动化可能会制约软件开发。(6)工具本身并无想象力。