

## 1、软件测试概述

**软件测试的IEEE定义：**使用人工或自动的手段来运行或测量软件系统的过程，目的是检验软件系统是否满足规定的需求，并找出与预期结果之间的差异。软件测试的发展趋势：

- ① 测试工作将进一步前移。软件测试不仅仅是单元测试、集成测试、系统测试和验收测试，还对需求的精确性和完整性的测试技术、对系统设计的测试技术将成为新的研究热点。
- ② 软件架构师，开发工程师，QA人员，测试工程师将进行更好地融合
- ③ 测试职业将得到更充分的尊重。
- ④ 设置独立的软件测试部门将成为未来软件公司的共识。
- ⑤ 测试外包服务将快速增长，和软件开发外包一样，软件测试外包将成为全球化的趋势。

### 软件测试工程师的素质：

责任心；沟通能力；团队合作精神；耐心、细心和信心；保持怀疑的态度，有缺陷预防的意识；不断学习的能力。

### 合格的测试工程师应具有的能力：

- ① 一般能力：包括表达、交流、协调、管理、质量意识、软件开发过程方法、软件工程等；
- ② 测试技能及方法：包括测试基本概念及方法、对测试工具的掌握、对专业测试标准的熟悉程度等；
- ③ 测试规划能力：包括风险分析及防范能力、测试目标及计划的制定能力等；
- ④ 测试执行能力：包括测试数据/脚本/用例的制定能力、测试比较及分析能力、缺陷记录及处理能力；
- ⑤ 测试分析、报告和改进能力：包括测试度量、统计技术、测试报告、过程监测及持续改进能力。

### 测试工程师的职责：

测试人员要了解项目需求内容，从用户的角度提出自己的测试看法；

测试人员要编写合理的测试计划并与项目整体计划有机地整合在一起；

测试人员要编写覆盖率高的测试用例；测试人员要认真仔细地实施测试工作，并提交测试报告以供项目参考；

测试人员要进行缺陷跟踪和分析。

## 2、软件测试基础软件的概念

软件是计算机系统中与硬件相互依存的一部分，包括程序、数据、与其相关文档的完整结合。软件 = 程序 + 数据 + 文档。

### 软件的特点：

- ① 软件是一种逻辑体，而不是具体的物理体，因而它具有抽象性；
- ② 软件的生产与硬件不同，它没有明显的制造过程，对软件质量的控制，必须在开发方面下功夫；
- ③ 在软件运行和使用期间，没有硬件那样的机械磨损和老化问题，然而它存在退化问题，必须进行多次的修改和维护；
- ④ 软件的开发和运行常常受计算机系统的制约，对计算机系统有着不同程度的依赖性，为了解除这种依赖性，在软件开发过程中提出了软件移植问题。

⑤ 软件本身是复杂的，软件的复杂性可能来自它所反映问题的复杂性，也可能来自程序逻辑结构的复杂性。

⑥ 软件成本的昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力，它的成本比较高。

## **软件的分类：**

### **按照功能划分：**

**系统软件：**如操作系统、数据库管理系统，各种驱动软件等

**应用软件：**如Office、金山词霸、QQ等

### **按照技术结构划分：**

**单机版本：**如Office，画图工具等

**C/S结构软件：**如QQ、MSN等

**B/S结构软件：**如新浪、搜狐、google等

### **按照用户划分：**

**产品软件：**Office、财务处理软件、金山毒霸等

**项目软件：**如为企业定制的OA系统等

### **按照开发规模划分：**

类别 参与人数 开发时间

小型 10人以下 1-4个月

中型 10-100人 1年以下

大型 100人以上 1年

## **软件测试的概念：**

软件测试就是为了发现错误而执行程序的过程(狭义观点)。

使用人工或自动的手段，来运行或测试软件系统的过程，目的是检验软件系统是否满足规定的需求，并找出与预期结果之间的差异。(标准定义IEEE)

软件测试就是为了证明程序有错，而不是证明程序无错误(辩证观点)。

测试被定义为“对软件系统中潜在的各种风险进行评估的活动”。(风险观点)软件测试就是“验证(Verification)”和“有效性确认(Validation)”活动构成的整体，即软件测试V&V。(标准观点)

要完整理解软件测试，就要从不同方面去审视软件测试，概括起来，软件测试就是贯穿整个软件开发生命周期，对软件产品(包括阶段性产品)进行验证和确认的活动过程，其目的是尽快尽早地发现在软件的缺陷。

## **软件测试的对象：**

① 源程序/目标代码

② 各开发阶段的文档(需求规格说明、概要设计说明、详细设计说明及其它相关文档)

## **软件测试的目的：**

从用户角度看的目的：通过软件测试发现隐藏的 errors 和缺陷，考虑是否可以接受该产品。从开发者角度看的目

的：表明软件产品不存在错误，验证软件实现了所有用户的要求。从测试人员角度看的目

的：发现错误，预测错误，提供软件可靠性错误，对软件做出评价。

- ① 帮助开发人员、测试工程师发现问题、分析问题。
- ② 减少软件的缺陷数目或者降低软件缺陷的密度。
- ③ 提高软件的可靠性
- ④ 评估软件的性能指标。
- ⑤ 增加用户对软件的信心。
- ⑥ 测试的最终目的是尽快尽早地发现在软件中的缺陷，通过修正各种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缺陷和错误造成的隐患所带来的商业风险。

### **软件测试的原则：**

- ① 所有测试的都应追溯到用户需求。
- ② 应当把“尽早地和不断地进行软件测试”作为软件测试者的座右铭。
- ③ 由于软件的复杂性和抽象性，在软件生命周期各个阶段都可能产生错误，所以不应把软件测试仅仅看作是软件开发的一个独立阶段的工作，而应当把它贯穿到软件开发的各个阶段中。在软件开发的需求和设计阶段就应开始测试工作，编写相应的测试文档。
- ④ 完全测试是不可能的，测试需要终止。
- ⑤ 想要进行完全的测试，在有限的时间和资源条件下，找出所有的软件缺陷和错误 使软件趋于完美是不可能的主要原因：① 输入量太大；② 输出结果太多；③ 路径组合太多。
- ⑥ 测试无法显示软件潜在的缺陷：进行测试是可以查找并报告发现的软件缺陷和错误，但不能保证软件缺陷和错误全部找到。
- ⑦ 充分注意集中群集现象（二八定理）：经验表明，在所测试程序段中，若发现的错误数目多，则残存的错误数目也较多。缺陷的二八定理指的是，一般情况下，80%软件缺陷出现在20%的功能区域，在测试过程中，投入主要的人力和精力重点测试这20%的功能区域。
- ⑧ 开发人员应避免检查自己的程序：基于心理因素，揭露自己程序中的问题总不是一件愉快的事，不愿否认自己的工作；由于思维定势，人们难于发现自己的错误。因此为达到测试的目的，应由客观、公正、严格的独立的测试部门或独立的第三方测试机构进行测试。
- ⑨ 尽量避免测试的随意性：应从工程的角度理解测试，它是有组织、有计划、有步骤的活动。

### **软件测试误区：**

**误区一：**如果发布出去的软件有质量问题，都是软件测试人员的错。

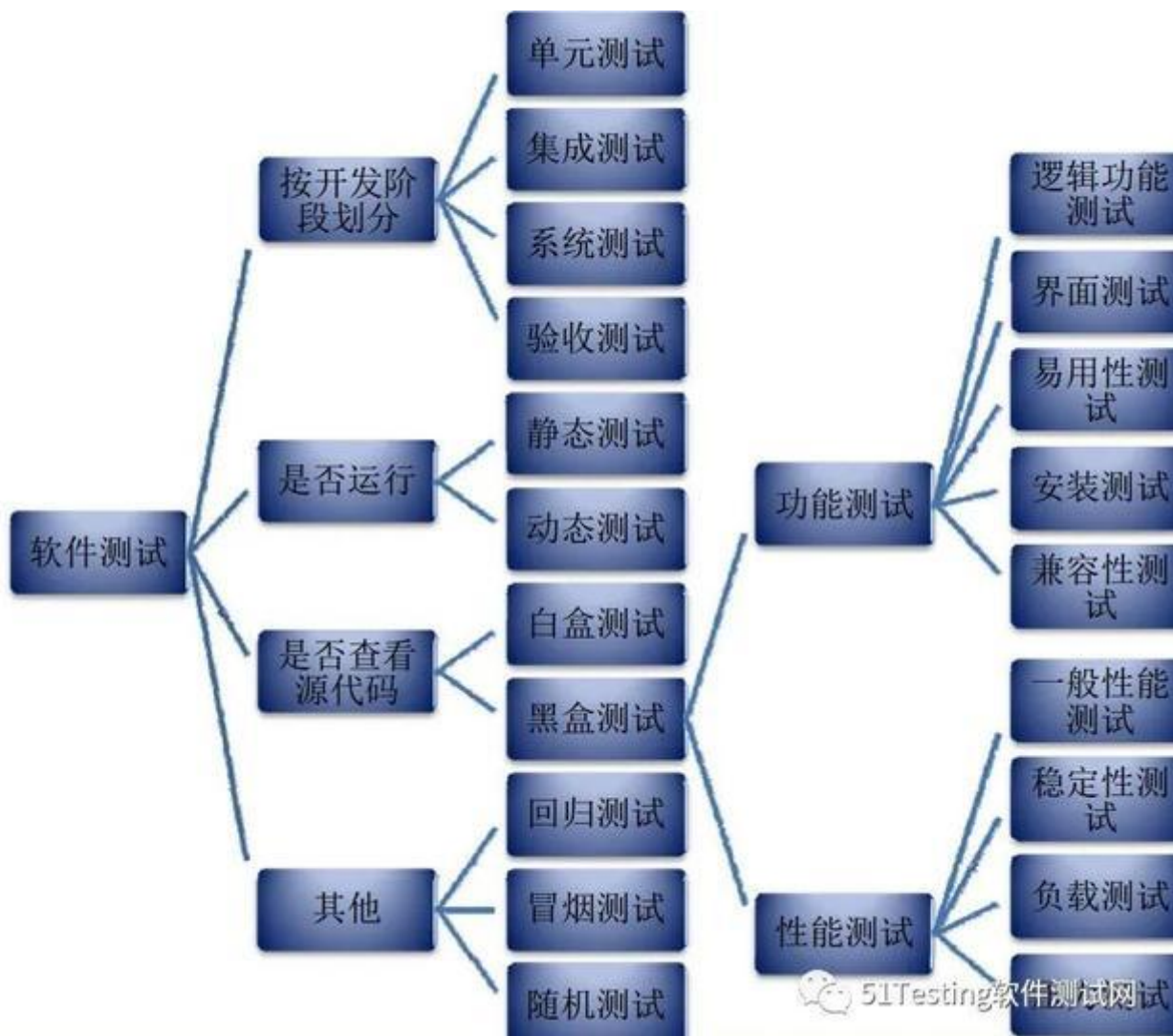
**误区二：**软件测试技术要求不高，至少比编程容易多了。

**误区三：**有时间就多测试一些，来不及就少测试一些。

**误区四：**软件测试是测试人员的事，与开发人员无关。

**误区五：**根据软件开发瀑布模型，软件测试是开发后期的一个阶段。

### **3、软件测试分类**



## 单元测试：

单元测试又称模块测试，针对软件设计中的最小单位——程序模块，进行正确性检查的测试工作。单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。

## 单元定义：

C中指一个函数，Java中指一个类，在图形化的软件中，单元一般指1个窗口，1个菜单。如何进行单元测试：单元测试主要用白盒测试，先静态地检查代码是否符合规范，然后动态运行代码，检查其实际运行结果，检查程序的运行结果是否正确是一个最基本的要求，还要关注容错处理，程序的边界值处理等。

## 集成测试：

集成测试又叫组装测试，通常在单元测试的基础上，将所有程序模块进行有序的、递增的测试。重点测试不同模块的接口部分。

## 系统测试：

指将整个软件系统看为一个整体进行测试，包括对功能、性能、以及软件所运行的软硬件环境进行测试。

## 验收测试：

验收测试指按照项目任务书或合同、供需双方约定的验收依据文档进行的对整个系统的测试与评审，决定是否接收或拒收系统。在系统测试的后期，以用户测试为主或有测试人员等质量保证人员共同参

与测试。

**α测试**：指的是指的是由用户，测试人员、开发人员等共同参与的内部测试。

**β测试**：指的是内测后的公测，即完全交给最终用户测试验收测试的重要性：验收签字,收钱。

测试名称	测试对象	测试依据	人员	测试方法	时间比例
单元测试	最小模块，如函数类等	《详细设计》	白盒测试工程师或开发工程师	主要采用白盒测试	1
集成测试	模块间的接口，如参数传递	《概要设计》	白盒测试工程师或开发工程师	黑盒&白盒测试相结合	2
系统测试	整个系统，包括软硬件	《需求规格说明文档》	黑盒测试工程师	黑盒测试	3
验收测试	整个系统，包括软硬件	《用户需求》，验收标准	主要为用户，还可能有测试工程师等	黑盒测试	4

**静态测试**：指不实际运行被测软件，而只是静态地检查程序代码、界面和文档中可能存在的错误的过程。

**动态测试**：指实际运行被测程序，输入相应的测试数据，检查实际输出结果与预期结果是否相符。

（动态测试方法为结构和正确性测试；动态测试工具Robot、QTP等）

**黑盒测试**：指的是把被测的软件看做一个黑盒子，我们不关心盒子里面的结构是什么样子的，只关心软件的输入数据和输出

**白盒测试**：指的是把盒子打开，去研究里面的源代码和程序结构。软件公司中，往往采用黑盒测试&白盒测试相结合的方式。

静态黑盒测试：看文档，看页面等静态白盒测试：开源代码等动态黑盒测试：使用软件等

动态白盒测试：运行源代码等**灰盒测试**：是介于白盒测试与黑盒测试之间的一种测试，灰盒测试多用于集成测试阶段，不仅关注输出、输入的正确性，同时也关注程序内部的情况。

**功能测试**：是黑盒测试的一方面，它检查实际软件的功能是否符合用户的需求。

逻辑功能测试(functiontesting)

界面测试(UItesting)

易用性测试(usability testing)

安装测试(installationtesting)

兼容性测试(compatibilitytesting)

**性能测试**：是软件测试的高端领域，通常我们所说的高级软件测试工程师一般就是指性能测试或是白盒测试工程师。时间性能（事务响应时间等）空间性能（系统资源消耗）一般性能测试可靠性测试负载测试压力测试

**回归测试**：指对软件的新版本测试时，重复执行上一个版本测试时的用例。

**冒烟测试**：是指在对一个新版本进行系统大规模的测试之前，先验证一下软件的基本功能是否实现，是否具备可测试性。

**随机测试**：是指测试中所有的输入数据都是随机生成的，其目的是模拟用户的真实操作，并发现一些边缘性的错误。

软件测试的过程：从软件开发的过程按阶段划分有：需求验证、单元测试、集成测试、确认测试、系统测试、验收测试

#### 4、白盒测试用例设计方法

测试用例（英文为TestCase,缩写为TC）：指的是在测试执行之前设计的一套详细的测试方案，包括测试环境、测试步骤、测试数据和预期结果。测试用例可以针对黑盒测试设计用例，也可以针对白盒测试设计用例。编写测试用例的唯一标准就是用户需求,具体的参考资料是《需求规格说明书》。

**设计测试用例的原因**：软件测试是一项有组织、有计划、有步骤的活动，为了将软件测试的行为转换为可管理的、具体量化的模式，需要创建和设计测试用例。

**测试用例的四性**：

**代表性**：能够代表并覆盖各种合理的和不合理合法的和不合法的、边界的和越界的以及极限的输入数据、操作等。

**针对性**：对程序中的可能存在的错误有针对性地测试。

**可判定性**：测试执行结果的正确性是可判定的，每一个测试用例都应有相应的期望结果。

**可重现性**：对同样的测试用例，系统的执行结果应当是相同的。

**测试用例的基本原则**：

利用成熟的测试用例设计方法来指导设计

测试用例的针对性

测试用例的代表性

测试用例的可判定性

测试用例的可重现性足够详细、准确和清晰的步骤

测试用例必须符合内部的规范的要求

**语句覆盖**：语句覆盖就是设计若干个测试用例，运行被测试程序，使得每一条可执行语句至少执行一次；

**判定覆盖（也称为分支覆盖）**：设计若干个测试用例运行所测程序使程序中每个判断的取真分支和取假分支至少执行一次；

**条件覆盖**：设计足够多的测试用例，运行所测程序，使程序中每个判断的每 条件覆盖设计足够多的测试用例 行所测程序使程序中每个判断的每个条件的每个可能取值至少执行一次；

**判定-条件覆盖**：设计足够多的测试用例，运行所测程序，使程序中每个判断的每个条件的所有可能取值至少执行一次，并且每个可能的判断结果也至少执行一次，换句话说，即是要求各个判断的所有可能的条件取值组合至少执行一次；

**条件组合测试：**设计足够多的测试用例，运行所测程序，使程序中每个判断的所有可能的条件取值组合至少执行一次；

**路径测试：**设计足够多的测试用例，运行所测程序，要覆盖程序中所有可能的路径。

主要测试技术：分支条件覆盖，基本路径测试5、黑盒测试用例设计方法

**主要测试技术：**等价类划分（边界值分析），因果图法，（正交实验法）

6、缺陷管理软件缺陷

**软件缺陷：**

是指存在于软件（程序、数据、文档）中的那些不符合用户需求的问题。

**软件缺陷的来源：**

**需求说明书：**需求说明书的错误或不清楚引起的错误，是缺陷第一大的来源。

**设计文档：**设计文档描述不准确、以及与需求说明书不一致，是缺陷的第二大来源。

**编码：**纯粹是由编码的问题引起。

**其它：**可能是系统集成、测试引起。

**软件缺陷的根源：**交流不充分（客户与开发人员、开发人员与测试人员等）软件的复杂性（功能复杂、开发复杂、测试复杂）开发人员的错误（对需求的理解、开发压力、能力与经验）需求的变化（需求说明书设计文档 程序的变更）进度压力（项目周期比较紧）

**软件缺陷的发现手段：**同行评审、测试、管理评审、QA发现、项目组内部发现、客户反馈为了便于缺陷的定位、跟踪和修改，要对所发现的缺陷，按照缺陷的严重程度、优先级、发现阶段、修复阶段、缺陷的性质、所属功能模块、系统环境等方面进行分类和统计。

**二八定理：**80%的软件问题总是发生在大约20%的功能模块中。

**缺陷密度：**基本的缺陷测量是以每千行代码的缺陷数(个/KLOC)来测量的，其测量单位是defects / KLOC。

**常见寻找bug的方法：**色彩、功能结构布局、图片、页面大小、字体、窗体大小、界面文字、容错处理（也为功能缺陷，所谓容错，就是容忍错误的能力。当用户在使用软件过程中发生错误后，软件应该能给出引导信息，指应用户进行正确的操作）、数据转换（增删改查）、性能缺陷（黑盒测试）。