# **测试基础理论**

## 一、软件测试基本概念

软件测试是一种用来促进鉴定软件的正确性、完整性、安全性和质量的过程。换句话说，软件测试是一种实际输出与预期输出之间的审核或者比较过程。在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。

通俗的理解就是：我要一个能吃饭的碗，而你不能给我一个盆子，也不能给我一个杯子，更不能给个有缺口的碗。而这一切就只是为了满足客户的要求。

## 二、软件测试的定义

（1）经典定义

1979年Myers提出，测试是为发现错误而执行程序的过程。

软件测试（Software Testing）是软件生存期的一个重要的阶段，是软件质量保证的一个关键步骤。通俗的来说，软件测试就是软件在投入运行前，对软件需求分析、设计规格说明和编码进行最终复审的活动。

（2）IEEE定义

使用人工或者自动的手段来运行或者是测量软件系统的过程，以检验软件系统是否满足规定的要求，并且找出与预期结果之间的差异。

## 三、软件测试的目的

（1）从用户的角度出发，希望通过软件测试暴露软件隐藏的错误和缺陷，从而考虑是否接受该产品

（2）从软件开发者的角度出发，希望表面软件产品不存在错误和缺陷，验证软件能正确实现用户的需求，使人们树立起对软件质量的信心

（3）从软件管理者的角度出发，希望在付出的有限资源（经费、时间等）的情况下能够达到软件的质量要求，

总而言之，测试的目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别。

## 四、软件测试的原则

**（1）测试应当基于用户的需求**

从用户角度来看，最严重的错误是那些导致程序无法满足需求的错误。应依照用户的需求配置环境并且依照用户的使用习惯进行测试并评价结果。假如系统不能完成客户的需求和期望，那么，这个系统的研发是失败。同时在系统中发现和修改缺陷也是没有任何意义的。

**（2）做好软件测试计划是做好软件测试工作的关键**

软件测试是有组织、有计划、有步骤的活动因此测试必须要有组织有计划，并且要严格执行测试计划避免测试的随意性。测试计划应包括：所测软件的功能，输入和输出，测试内容，各项测试的进度安排，资源要求，测试资料，测试工具，测试用例的选择，测试的控制方法和过程，系统的配置方式，跟踪规则，调试规则，以及回归测试的规定等以及评价标准。另外，回归测试的关联性一定要引起充分的注意，修改一个错误而引起更多错误出现的现象并不少见。

**（3）尽早的开始软件测试并且不断的进行软件测试**

由于软件的复杂性和抽象性，在软件生命周期各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，而应当把它贯穿到软件开发的各个阶段去。在需求分析和设计阶段就应开始进行测试工作，编写相应的测试计划及测试设计文档，同时坚持在开发各阶段进行技术评审和验证，这样才能尽早发现和预防错误，杜绝某些缺陷和错误，提高软件质量。

**（4）测试前必须明确定义好产品的质量标准**

必须用预先精确对应的输入数据和输出结果来对照检查当前的输出结果是否正确，做到有的放矢。系统的质量特征不仅仅是功能性要求，还包括了很多其他方面的要求比如稳定性、可用性、兼容性等等。

**（5）避免测试自己的软件**

由于心理因素的影响或者程序员本身错误的理解了需求或者规范导致程序中存在错误，应避免程序员或者编写软件的组织测试自己的软件。一般要求有专门的测试人员进行测试，并且还要求用户参与，特别是验收测试阶段，用户是主要的参与者。

**（6）应充分注意测试中的集群现象**

一般来说，一段程序中已发现的错误数越多，其中存在的错误概率也就越大。错误集中发生的现象，可能和程序员的编程水平和习惯有很大的关系。因此，对发现错误较多的程序段，应进行更深入的测试。

**（7）必须检查每个实际输出结果**

彻底检查每个测试的执行结果，避免因为疏忽或者对结果与预期结果的一致性主观臆断造成错误遗漏。

**（8）穷举测试是不可能的**

由于时间和资源有限，穷举测试是不可能的，软件测试不能无限进行下去，应适时终止。此外，应避免冗余测试。

**（9）测试设计决定了测试的有效性和效率**

根据测试的目的，采用相应的方法去设计测试用例，从而提高测试的效率，更多地发现错误，提高程序的可靠性。除了检查程序是否做了应该做的事，还要看程序是否做了不该做的事；另外，测试用例的编写不仅应当根据有效和预料的输入情况，也需要根据无效和未预料的输入情况。

**（10）注意保留测试设计和说明文档，并注意测试设计的可重用性**

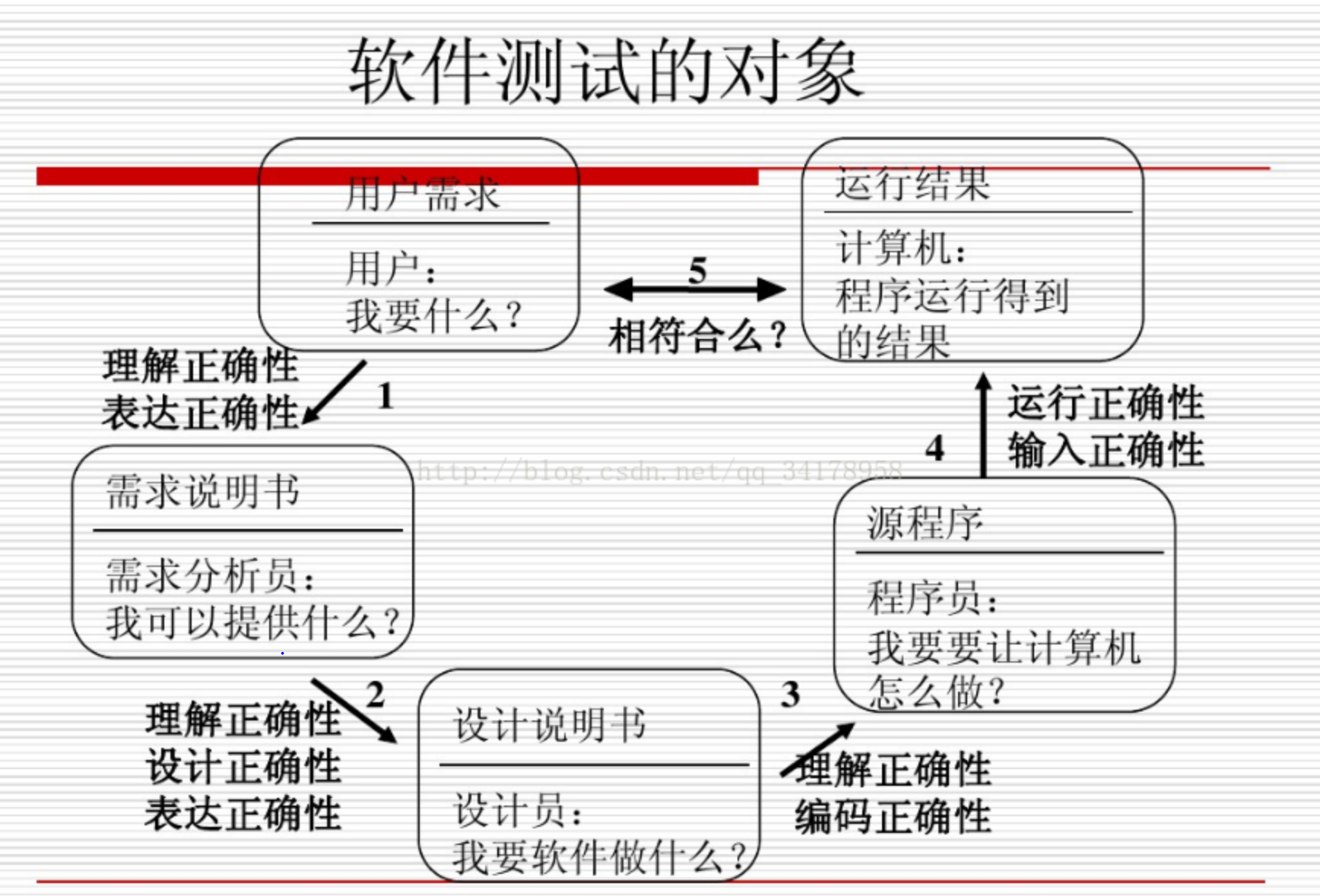
妥善保存测试计划、测试用例、出错统计和最终分析报告，并为维护等提供方便。

## 五、软件测试的对象

软件测试不等同于程序测试，软件测试贯穿于软件定义和开发的整个期间。需求分析，概要设计，详细设计以及程序编码等各个阶段所得到的文档，包括需求规格说明书、概要设计规格说明，详细设计规格说明以及源程序，都是软件测试的对象。

大可概括为：

1. 理解软件的需求
2. 软件概要设计
3. 软件源代码
4. 软件详细设计
5. 可运行程序
6. 软件运行环境（测试提交Bug时，应该注明当前环境）

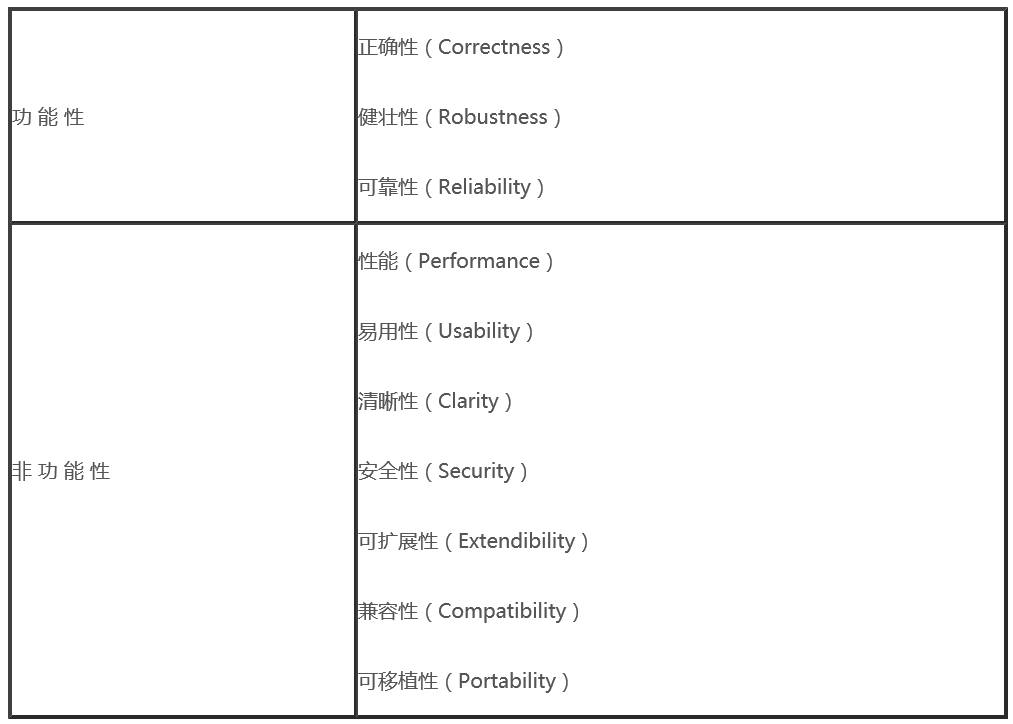


## 六、软件质量与软件测试

**（一）关于软件质量**

软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述的开发标准、以及所有专业开发的软件都应具有的和隐含特征相一致的程度。质量属性可分为两大类：“功能性”与“非功能性”。

“10大”软件质量属性：



如何提高软件质量呢?

提高软件质量的基本手段是消除软件缺陷，消除软件缺陷有三种基本方式：

（1）在开发过程中有效地防止工作成果产生缺陷，将高质量内建于开发过程之中。这就是“预防胜于治疗”的道理，无疑是最佳方式，但是要求开发人员必须懂得正确地做事（门槛比较高）。

（2）当工作成果刚刚产生时马上进行质量检查，及时找出并消除工作成果中的缺陷--这种方式效果比较好，人们一般都能学会。最常用的方法是技术评审、测试和质量保证等，已经被企业广泛采用并取得了成效。

（3）当软件交付给用户后，在使用过程中出错了，赶紧请开发者来补救，这种方式的代价最高。可笑的是，当软件系统在用户那里出故障了，那些现场补救成功的人倒成了英雄，好心用户甚至还寄来感谢信。

**（二）关于软件测试**

测试的目的：尽可能多的发现缺陷，比如功能的错误，性能低下，易用性差。

测试的思路：先假设程序存在什么缺陷，然后执行程序来发现缺陷。

测试类型：白盒测试、黑盒测试等（后续详细了解）

测试内容可分为：

**1、功能测试**

检查软件的功能是否符合要求。枚举方法：构造合理的输入，看是否有期望的输出。边界值方法：采用定义域的边界值进行测试。

**2、容错性测试**

检查软件在异常情况下的反应，容错性好的软件会确保系统不发生难以预料的崩溃。方法：构造一些不合理的数据看系统的反应（错误的数据类型或定义域外的值）。

**3、性能与效率测试**

测试软件的速度与对资源的利用率。极限测试：持续不停地给服务器发送请求看是否会死掉，给程序输入特别大的数据看是否能吃得消。获取测试的绝对值（如数据的传输率）：记录运行环境对软件的影响。获取测试的相对值（如该软件和其他软件相比快多少倍）：确保被测试的几个软件具有相同的软件和硬件环境中。

**4、易用性测试**

用户不用看用户手册，即具有好的易用性。

**5、文档测试**

检查文档的正确性，完备性，可理解性。