

如何保证软件测试的质量

14126075 陈檬檬

软件测试伴随着软件的产生而产生，它是保证软件质量的重要方法。只有通过测试才可以确保其品质，才能够保证可靠性好，同时它还能够验证软件能不能实现预估的目的。目前测试已经不单纯的存在于开发的工作中，它已经贯穿到总的研发步骤中，开展测试的时间越靠前 总体的开发费用就会越少。通过很多的统计资料我们得知，测试的活动量在总的开发工作中占据了大约 40%，。对某些特殊行业的软件(如飞机控制、核反应堆等)的测试所花费的时间往往是其他软件的 3 ~ 5 倍。之所以耗费如此多的财力和时间来开展该项工作目的就是提升软件的品质以及稳定性。

1. 何为软件测试

简单地说，软件测试就是为了发现错误而执行程序的过程，它的目的是为了分析存在的错误。在 IEEE 提出的软件工程标准术语中，软件测试被定义为：“使用人工和自动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的需求或弄清楚预期结果与实际结果之间的差别。” 换句话讲，它是结合开发时期的要求以及程序的结构而设计得到的一个测试案例，然后结合该案例的运作情况来得知存在的错误的一个步骤。软件测试是与软件质量密切联系在一起的，归根结底，软件测试是为保证软件质量。所以软件测试是软件质量保证的重要环节，提高软件测试的质量也是保证软件质量的重要手段。

2. 保证软件测试质量的方法

2.1 测试人员

首先，要保证软件测试团队的人员能力。测试人员最好应具备相应系统的开发经验、掌握测试技术和方法，熟悉至少某一方面的系统业务知识并具备良好的沟通合作能力和怀疑精神。测试人员除了测试能力外，最好有一定的开发技能。开发能力在测试过程中，可以帮助测试

人员深入理解系统的实现，找到更好的测试方向，也可以帮助测试人员理解、

分析既有系统的业务逻辑，解决项目文档对已有系统的背景交代模糊的问题，同时也有助于测试人员深入分析、定位测试过程中发现的异常，提升与开发人员沟通的效率和效果，特别是在核心系统的变更维护类项目上，开发能力对测试人员提高测试工作的质量和效率尤为重要。一定的业务知识在系统测试过程非常重要，尤其对功能测试来说，丰富的业务知识可以帮助测试人员深入、全面地理解需求、设计合理的测试场景和测试案例。许多项目需求中，对功能点隐含的、背景性的规则描述不多，熟悉业务的测试人员在测试外汇相关项目时，就会设计案例验证其处理的正确性。测试人员需要具备良好的沟通协作能力，良好的沟通协作能力来自于对业务和系统的充分了解、有较高的语言表达能力和团队精神，以及有效的沟通合作技巧。优秀的测试人员还要有擅于发现问题的精神，对有可能出现问题的各个方面，如文档、架构、应用系统软硬件、测试环境等，都会主动从业务场景的角度去设计案例，并及时查错和纠错，坚持用事实和结果来说明问题。

2.2 测试技术

2.2.1 采用测试性设计技术

软件测试是目前用来验证软件是否能够完成所期望的功能的唯一有效的方法。但是在测试的实施过程中，由于种种原因导致测试的难度相当大，甚至出现了无法测试的情形。为了提高软件的可测试性，我们在软件设计时应当遵循测试性设计原则，通过改变设计或代码、为软件增加专门测试结构等方法来提高软件的可测试性。

在设计的时候要切实的遵照测试性理念，经由变幻代码或是其他的一些措施来尽可能的提升它的可测试特征。(1)测试驱动设计。这种设计就是直接把软件需求变成测试代码。当明确了测试性能规定之后，要进行代码编订工作。要先进行验收测试，然后开展单元的测试，最主要的是在开发的时候积极的修正处理。(2)所有的操作都要对应具体的措施，确保措施的可行性好。通常都是一些小规模的措施，使用这些措施能够确保调用的时候更为便利。(3)显示与控制分离。把代码移到 GUI 视图的外面，各种 GUI 动作就能成了模型上的简单方法调用。此时在改动程序的时候不会对试图产生负面效果,而且这样更加的易于被人们认可。(4)针对那些或许会成为参数的类应该设置接口。以此来阐述外在程序组合获释在应该变换接口的时候得到生成空类，这样就可以当成是参数输入了。

2.2.2 选择合适的测试管理模型

基于模型的测试主要考虑系统的功能，可以认为是功能测试的一种。测试模型体现了被测试系统的最本质的功能关系。其较之于系统要更加容易研发。任何能够运作的模型都要确保其可以提供充足的信息。因此要保证模型合乎如下一些规定才可以。第一，规定其应该是一个具体测试的十分精准的体现，要体现出检查用到的全部特点。第二，应该能抽象的显示出细节内容。第三，能够体现出全部事件以及全部的活动。第四，能够体现出系统的所有状态，只有这样才能够通过可行的措施来明确已经达到或是尚未达到的状态。

模型是系统功能的形式化或半形式化的表示，支持输入状态组合的系统枚举。基于模型的测试主要考虑系统的功能，可以认为是功能测试的一种。测试模型体现了被测试系统的最本质的功能关系。而且要比系统本身更易于开发和分析。一个可测试的模型要能提供足够的信息用来产生测试用例。所以可测试的模型必须满足以下要求：(1)必须是某种测试实现的完全准确的反映，模型必须表示要检查的所有特征；(2)是对细节的抽象；(3)可以表示所有事件和所有的动作；(4)可以表示系统的各种状态，以便由可行的方法来确定已达到或没有达到什么状态。

2.2.3 使用恰当的测试科技

目前的测试措施种类非常多，都有着一定的优缺点。但任何一种测试方法都无法覆盖所有测试的需求。所以要积极的分析这些措施，明确其存在的优点和缺陷，充分掌握软件测试的各种方法，熟悉其优缺点，根据系统需求精心设计组合。通常可从如下两个层次展开：

(一)从代码的特性角度出发展开测试：(1)单元测试：按照代码的单元组成逐个进行测试。(2)功能测试：按照软件的功能或特性逐个进行测试。(3)系统测试：对完整的代码进行编译和连接，以检查程序的主要功能能否达到预期目标。(4)回归测试：对以前修复过的 Bug 重新进行测试，看该 Bug 是否会重新出现。值得注意的是，回归测试并不是软件测试的一个独立阶段。

(二)从用户的角度出发展开测试：(1)配置测试：从用户的使用出发进行多方面的测试。(2)兼容性测试：主要考虑软件和操作系统的兼容性问题。(3)压力测试：在各种极限情况下对产品进行测试，以检查产品的长期稳定性。(4)性能测试：测试是保证程序具有良好的性能，能否达到预期的性能指标。(5)文档和帮

助文件测试：对文档和帮助文件进行检测，保证用户可以通过学习文档和帮助文件正常使用产品。(6)Alpha 和 Beta 试：在正式发布产品之前将软件测试版发送给用户，让用户在使用中找到能够存在的 Bug 或者反馈相关信息，以便在正式版中得到解决。

2.2.4 建立可复用的测试用例库

在测试的时候得到的那些用例，对于提升软件的品质有着非常关键的意义，其复用的意义更是突出在测试中应当构造出基于复用的测试用例，并采用测试用例库管理的方法来实现，这样可以大大提高软件测试的工作效率。

(1)基于复用的目的，对所使用的测试用例进行统一的建模组织，有效地将测试用例收集到测试用例库中，并按照行业项目等进行多级合理的分类、组织、存储。对采用不同方式描述的测试用例，将分别实现不同程度的复用。(2)对数据库中存有的测试用例实行有效管理，通过提供有助于复用的多种查询方式，确保测试用例的复用程度，对不同类别的测试用例进行复用频率的统计，为测试人员提供有效的复用信息，在保证软件测试质量的前提下，提高了测试的效率。(3)对数据库中的测试用例具体实现复用，通过查询出所需的测试用例，得到测试用例信息，针对不同描述方式的测试用例采用不同的复用方式进行复用。

2.3 其他

2.3.1 尽早地、不断地进行软件测试，制定完整的测试计划

缺陷与错误可能在软件周期的任何一个阶段出现，所以，要尽早地、不断地进行测试，以便尽早发现软件的缺陷。并严格执行测试计划，排除随意性。测试工作不应有时间就多测，没时间就少测而应该有一个完整的计划，并严格执行它，这样才能保证测试的效果，从而保证软件的质量。

2.3.2 测试用例设计要严密

不仅要有合理的输入数据，还要有不合理的输入数据以及预期的输出结果。测试时，人往往只注意输入预期合理的数据进行测试，而忽略了对不合理数据输入的测试。实际中，以特殊的方式运行程序时，常会出现一些错误，使用预期不合理的数据进行测试比用合理的数据进行测试收获更大。测试用例不仅要有确定的输入数据，还要有确定的输出结果。否则，很可能会把实际上是错误的结果作为正确的输出结果。

3. 总结

软件测试的目的不是为了仅仅找出错误，而是通过它发现错误分析错误，找到错误的分布特征和规律，从而帮助项目管理人员发现当前所采用的软件开发过程的缺陷，以便改进。同时也能够通过设计有针对性的检测方法，改善软件测试的有效性。即使测试没有发现任何错误，也是十分有价值的，因为完整的测试不仅可以给软件质量进行一个正确的评价，而且是提高软件质量的重要方法之一。提高软件测试的质量也是提高软件质量的重要手段。