

北京交通大学

软件质量保证

——如何保证软件测试的质量

范玲珊 14126085

2015-5-20

如何保证软件测试的质量

1. 1、软件质量保证的重要性

软件质量保证工作是一个软件公司中最应该被重视的工作。因为高质量的软件是一个公司发展和壮大的最基本因素。诚然有些公司是准备有一定市场后就要卖出去，有些公司是依靠人脉关系销售软件，有些公司因为起步早而在某一行业处于近乎垄断的地位等等。但是中国现在已经加入 WTO，外国的很多软件公司已经进入中国，并且以后还会有更多的公司进入中国市场。我们可以问问自己：为什么优秀和一般的同类软件的销售价格会相差十几倍至几十倍？

1. 2、软件生命周期中如何保证软件质量

我认为高质量的软件产品是一个软件团队所有成员都负责任的完成自己任务以后的必然产物。

首先说说团队，这其中涉及的需求人员、设计人员、开发人员、测试人员都应该真切的视自己为团队的必不可少的力量，都应该为了项目或产品的成功竭尽所能的去工作，只有团队真正的拧成一股绳的时候才具备了产出高质量软件的基本条件。这是我要说的第一点：团队认同感、归属感。

高质量的需求调研文档是软件成功必不可少的条件，但是不同的人对同一句话的理解往往会有差异，因为立场不同。所以想要保证需求的质量，需求人员必须把自己置身到用户的立场去感受、去调研、去理解目标用户反馈的信息。对于不确认的信息要想尽办法搞清楚。所以需求调研人员最好是行业专家。需求文档整理出来后，必须经过客户方代表和公司设计、开发、测试的共同评审才能最终定稿，并最终进入软件设计流程。这是我要说的第二点：软件需求必须用“心”去做，并且监督评审必须到位。

接下来就进入了软件的生产流程，在设计阶段，设计人员是主角，开发人员、测试人员、需求人员要可以及时获得设计文档。设计人员必须在实现需求的情况下，站在用户的立场上去设计功能，实现最好的用户体验。在设计评审时，开发、测试、需求要从用户的角度去评判设计，根据需求从用户的角度去评审设计，这真的很重要。问题如果能在设计阶段就发掘出来会极大的减少资源的浪费，缩短产品或项目周期。这是我要说的第三点：设计要注重用户体验，同时监督评审也必须到位。

软件进入开发测试流程后，实际的开发人员应该站在用户的角度上去开发每

一个功能，如果有比设计更好的实现方法，应及时和设计、测试、需求人员沟通，共同 确认是否更改设计。每一个功能完成后，必须进行完整的自测，然后及时送测给测试人员，测试人员也要在用户的角度进行测试，发现问题或建议及时反馈、沟通和 处理。还有很重要的一点，测试必须要有测试用例。测试开始前，测试用例必须经过评审，当然评审粒度根据公司资源确定。这是我要说的第四点：开发是软件的制造者，测试是软件质量的保证者，两者相辅相成，荣辱与共。

高质量的软件是一个软件团队共同努力的结果，任意一个环节出问题都可能造成团队的灾难。团队领导者必须要想办法、尽全力将自己的团队凝结在一起，使大家 具有团队荣誉感和使命感。软件生命周期的各个阶段都有工作重点，团队领导必须把握好。团队领导不能轻视任何一个环节的工作，否则高质量的软件只能是一句空话。在这里，我们从软件测试的基本概念开始，对如何以软件测试性设计为中心、合理运用软件测试技术来提升软件测试质量提出了自己的看法。

1.3、软件测试是软件质量的保证者

软件产品的质量取决于软件开发过程，软件测试作为软件生存期中的一个重要阶段，受重视程度越来越高。软件测试是保证软件质量和可靠性的关键步骤，也是用来验证软件是否能够完成所期望功能的唯一有效的方法。测试已不仅仅局限于软件开发中的一个阶段，它已开始贯穿整个软件开发过程，进行测试的时间越早，整个软件开发成本下降就越多。大量统计表明，软件测试的工作量往往占到软件开发总量的 40%以上，在极端的情况下，甚至可能高达软件工程其它步骤成本总和的三至五倍，其目的是尽可能的提高软件产品的质量和可靠性。

2.1、软件测试相关概念

(1) 软件测试：软件测试是为了发现错误而执行程序的过程。或者说，软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计出一批测试用例，并利用这些测试用例的运行结果来发现程序错误的过程。

(2) 软件测试用例：测试用例实际上是对软件运行过程中所有可能存在的目标、运动、行动、环境和结果的描述。测试用例是测试组织的最小单位，指对一项特定的软件产品进行测试任务的描述，体现测试方案、方法、技术和策略。内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等，并最终形成文档。

软件测试的核心是设计和执行测试用例。而测试用例的选择问题可以看作是

从庞大的输入状态组合中，搜寻哪些可以发现错误的状态组合。因此需要用抽象的手段来尽量使测试更加有效。

(3) 测试用例库：完整的单元测试很少只执行一个测试用例，开发人员通常都需要编写多个测试用例才能对某一软件功能进行比较完整的测试，这些相关的测试用例称为一个测试用例集。将大量的测试用例收集到测试用例库中，合理的分类后供测试人员选择使用，能够极大地提高软件问题的发现率。

2、提高测试质量的方法

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。

(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。在确定软件测试性能要求的基础上优先编写测试代码。先写验收测试，再写单元测试，并在开发过程中不断修正。

(2) 每个操作对应一个方法，使方法小型化。使用小型化方法说明和重载带缺省方法参数的方法，使得测试中调用这些方法变的很容易。

(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。这样，在修改程序功能不会影响视图，同时通过方法调用测试功能也比间接地测试功能更容易。

(4) 对于可能要作为参数的类，做一个接口。用接口说明外部程序组件或在需要时改变接口形成一个空类作为参数传入。

2.2 选择合适的测试管理模型

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。

2.3 采用合理的测试技术

软件测试方法众多，各有优点，但任何一种测试方法都无法覆盖所有测试的需求。这就需要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。我们可以从以下两个方面展开：

2.3.1 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。

2.3.2 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 测试：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。

2.4 建立可复用的测试用例库

软件测试过程中产生的大量测试用例对保证软件的质量起到关键作用，测试用例的复用就显得尤其重要。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。

3、结语

本文从技术的角度提出了一种以软件测试性设计为中心，配合合理的软件测试方法来实现提高软件测试质量的方法，测试的目标是发现 BUG，如果只是黑盒功能性测试，可以借助一些开源的自动化测试工具来协助完成，专业的测试人员是会写代码、能看懂设计的，会根据模块的特性设计打桩模块，这个比用通用的测试工具更有针对性和高效。

即使是自动化测试，也很难达到 100%的覆盖率，所以设计测试用例很考验测试人员的水平，凭借经验在可能出错的边界设计用例，这些经验是长期积累出来的。性能测试和安全测试，要了解被测对象上线后面对的环境，不仅要了解项目需求，还要懂设计。比如：系统最大并发用户数、数据库读操作频度、写操作频度、哪些会成为瓶颈、系统是如何优化和做负载均衡的、排队和缓冲是怎么设计的、出现冲突的优先策略是什么、系统安全级别是什么、数据传输是否要加密、数据库存储的是明文还是加密过的、是否涉及到防火墙、入侵检测、VPN、U 盾等软硬件安全防护措施等等。只有全面了解设计，才能有针对性的保证软件质量。