

如何保证软件测试的质量

14126103 焦继笑

摘要：随着计算机应用越来越广泛与深入,软件也越来越复杂,人们已清楚的认识到软件产品和其它工业产品一样,未经测试、试验是不能作为产品推向市场的。软件产业的发展,需要合格的、高质量的商品化软件产品。软件质量提高是一个庞大的系统工程,涉及到技术、过程和人员等综合因素,本文针对软件质量提高工作的关键环节——软件测试——进行探讨,着重讨论了软件测试和质量提高工作中可能面临的问题。

关键词： 软件测试 软件质量 质量保证 质量提高

1. 引言

软件质量作为参与国际竞争的必要条件,日益受到人们的关注。由于受到资源限制和环境影响,多数 IT 组织追求短期利益、放弃长远质量投资在所难免,陷入发展的恶性循环。显然,在合理借鉴国外成功经验的基础上,探寻切合国内实际情况的软件质量提高途径是当务之急。软件测试在软件生命周期中占据重要的地位,在传统的瀑布模型中,软件测试仅处于编码之后、运行维护阶段之前,是软件产品交付用户使用之前软件质量保证的最后手段。这是一种误导,软件生命周期每一阶段中都应包含测试,从静态测试到动态测试,要求检验每一个阶段的成果是否符合质量要求和达到定义的目标,尽可能早的发现错误并加以修正。如果不在早期阶段进行测试,错误的不断扩散、积累常常会导致最后成品测

试的巨大困难、开发周期的延长、开发成本的剧增等等。

2. 软件质量定义

软件质量，是贯穿软件生存期的一个极为重要的问题。是软件开发过程中所使用的各种开发技术和验证方法的最终体现。因此，在软件生存期中要特别重视质量的保证，以生成高质量的软件产品。软件质量是一个软件企业成功的必要条件，其重要性无论怎样强调都不过分。软件质量与传统意义上的质量概念并无本质差别，只是针对软件的某些特性进行了调整。软件质量由三部分构成：软件产品的质量，即满足使用要求的程度。软件开发过程的质量，即能否满足开发所带来的成本、时间和风险等要求。软件在其商业环境中所表现的质量。

总结起来，高品质软件应该是相对的无产品缺陷或只有极少量的缺陷，它能够准时递交给客户，所花费用都在预算内，并且满足客户需求，是可维护的。但是，有关质量好坏的最终评价依赖于用户的反馈。

3: 软件测试与软件质量保证之间的关系

软件质量保证与软件测试二者之间既存在包含又存有交叉的关系。软件测试能够找出软件缺陷，确保软件产品满足需求。但是测试不是质量保证，二者并不等同。测试可以查找错误并进行修改，从而提高软件产品的质量。软件质量保证则是避免错误以求高质量，并且还有其他方面的措施以保证质量问题。

从共同点的角度看，软件测试和软件质量保证的目的都是尽力确保软件产品满足需求，从而开发出高质量的软件产品。两个流程都是贯穿

整个软件开发生命周期中。正规的软件测试系统主要包括：制定测试计划、测试设计、实施测试、建立和更新测试文档。而软件质量保证的工作主要为：制定软件质量要求、组织正式审查、软件测试管理、对软件的变更进行控制、对软件质量进行度量、对软件质量情况及时记录和报告。软件质量保证的职能是向管理层提供正确的可行信息，从而促进和辅助设计流程的改进。软件质量保证的职能还包括监督测试流程，这样测试工作就可以被客观地审查和评估，同时也有助于测试流程的改进。

二者的不同之处在于软件质量保证工作侧重对软件开发流程中的各个过程进行管理与控制，杜绝软件缺陷的产生。而测试则是对已产生的软件缺陷进行修复。

4： 如何提升软件测试质量

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。

1、 提高测试质量的方法

1.1 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。

1.2 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，以便由可知的方法来确定已达到或没有达到什么状态。

1.3 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：

1.3.1 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。

1.3.2 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。

1.4 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。

2、测试人员素质

2.1、需要了解行业的背景，以及测试的需求，这需要对需求进行细化，转变为可以测试的测试需求。

2.2、参加里程碑的评审活动，多与需求人员、设计、开发人员进行沟通。

2.3、实时调整测试策略，分析测试过程中的风险。

总结：软件测试是软件质量保证的重要手段，有些研究数据显示国外软件开发机构，40%的工作量花在软件测试上，软件测试费用占软件开发总费用的 30%至 50%。对于一些要求高可靠、高安全的软件，测试费用可能相当于整个软件项目开发所有费用的 3 至 5 倍。由此可见，要成功开发出高质量的软件产品，必须重视并加强软件测试工作。