

如何保证软件测试的质量

14126135 宋静

0 引言

随着计算机技术的普及，计算机软件已渗透到社会生活的方方面面。软件测试是伴随着软件的产生而产生的，有了软件生产和运行就必然有软件测试。随着软件产业的蓬勃发展和对软件质量的重视，软件测试也越来越被人们看重。软件测试是为软件项目服务的，其根本目的是为了^①提高软件质量，降低软件项目的风险，而软件测试的质量保证是保障软件测试顺利进行完成的必要手段。

1 软件测试

随着计算机硬件成本的不断下降，软件在整个计算机系统的成本中占有越来越高的比例，如何提高软件质量是整个计算机软件行业的重大课题。软件测试作为软件开发的一个重要环节，日益受到人们的重视。软件测试是为了尽可能多地找出程序中的错误，保证软件产品的高质量和可靠性，在分析、设计等各个开发阶段结束前，对客户的软件产品进行严谨的技术评审，因此，软件测试的质量尤为重要。

1.1 软件测试的概念理解

大量统计资料表明，软件测试的工作量往往占软件开发总工作量的 40%以上，对于一些特殊的软件研发费的成本可能是该软件其他开发阶段成本的三倍甚至更多。仅就软件测试而言，它的目标是发现软件中的错误，但是，发现错误不是软件工程的最终目标，而是要开发出符合用户需要的软件。因此在软件测试中，从开发者的角度出发，就希望该软件产品不存在什么错误，通过测试已经表明可以满足用户的需求，但是从用户的角度的出发，则希望软件测试中发现更多隐藏的缺陷和错误。所以对于软件测试的工作人员来说，就应把着眼点放在如何尽可能地发现软件错误这个基础上，不让这些问题遗留到用户的使用阶段中去。

对于测试，我认为更为合适的定义应该是：测试是为发现错误而执行程序的过程，它表明软件测试是一个破坏性的过程，甚至是一个“疯狂找茬”的过程。为确保测试工作的顺利进行，就要对其进行有效管理。在一个软件工程项目中，不同的管理者对软件测试管理有不同的理解，但无论怎样，软件测试管理都要求测试人员掌握软件测试技术和软件测试工具，它是提高软件质量的一种重要手段。

1.2 当前软件测试中存在的问题分析

1) 软件开发透明度差，测试人员未尽早参与单元测试

由于软件生产单位大多采用“手工作坊”式的开发方式，由开发者自己设计、自己编程、自己测试、自己包维护，即所谓的“三自一包”。未能加强对软件开

发过程的管理，使得软件出现故障后只能由开发者自己去维护。同时单元测试是对软件基本组成单元的测试，也就是对软件模块的测试。在实际开发项目的过程中，考虑到开发人员对软件本身的了解以及节省项目开发时间，单元测试基本上都是由开发人员自行测试，这样做的结果是在单元测试的过程中很难发现潜在的问题，而当这些潜在的问题出现时往往是在的最后综合测试阶段，这样就增加了解决问题的难度，影响了软件开发的进度。

2) 缺少测试工具的使用，手工测试过多

传统的测试大多还是以手工为主，测试人员根据需求规格说明书与测试对象进行比对，人工测试过程中的手工成本高、测试人员的随意性、缺乏测试相关知识以及人工操作的低效率、误差大等因素都会直接影响到测试的结果和软件的质量，而且随着软件项目规模的不断扩大，在对测试案例进行统一配置管理时，这种人工测试的难度也会逐渐加大。

3) 测试在软件基本开发完成后才启动，未能对软件进行充分的检测

在传统的瀑布型的软件开发模式中，软件测试阶段位于编码阶段之后进行，就是说再将所有的编码完成之后才开始对其进行测试。这样的测试在软件开发过程中是十分不利的。随着软件项目规模越来越大，软件的缺陷发现的越晚，修改、补救所耗费的成本就越高，同时也会延长项目的开发时间，对最终的测试也增加了难度。除此之外，软件的测试软件、测试工具缺乏，标准、规范和制度不健全。目前绝大多数的软件检测都与硬件一同进行，在软件的验收中，所谓的检测也是对预先制定的几个用例进行检测，而不是按照软件开发的各个阶段进行单元测试、部件测试、配置项测试、系统测试等，起不到软件测试的应有作用，因而造成软件的缺陷多、故障多。

综上所述，为了确保软件的质量，测试不能在代码编写完成之后再开展工作，应该将测试与开发作为两个相互关联、相互结合的并行过程，在开发活动的前期就开展测试活动。本文将分别从软件测试技术，软件测试工具以及软件测试过程管理这三个角度论述如何在软件开发过程中保证软件测试的质量。

2 基于软件测试技术保证软件测试的质量

测试目的是发现错误，为了达到这一目的，需要计划和执行一系列必要的测试步骤，而每一个测试步骤都是通过一系列的系统化测试技术来完成的，以保证测试设计和用例的质量，在此总结了四种设计测试用例的思路，以保证测试的质量。在实际项目中结合需求，综合考虑，选择最佳方案。

2.1 基于规范的测试技术设计测试用例

在该类测试中，除了早已熟悉的边界值分析、等价类划分等方法外，还有决策表、有限自动机、形式化规范等方法。决策表用来表示条件和行为之间的逻辑关系，从条件和行为的所有可能组合中系统地选取测试用例，可用因果图来表示

这样的关系。有限自动机用来对测试系统进行建模，并采用相关的技术，选取测试用例覆盖其中的状态及其转换。形式化规范是用形式化语言给出规范，进而实现自动地由规范生成测试用例，同时提供预示输出，用以检验测试结果。

2.2 基于代码的测试技术设计测试用例

在该类测试中，主要包括参考模型建立、控制流测试和数据流测试。主要用图形的形式来表示程序的控制结构，其中的结点和边对应程序中的元素，结点表示语句或语句的顺序流，边表示结点之间的控制转换；控制流测试旨在覆盖程序中的所有语句和语句块，可以用决策/条件覆盖等准则来度量覆盖率；数据流测试中，使用控制流程图作为注释来解释如何定义和使用程序变量，沿着流程图的不同流向看出同一个变量的值的变化，可以用一个三元组来表示，其中 V 是变量， d 是定义 V 的结点， u 是使用 V 的结点，这样，就可以看出三者之间的关系。

2.3 基于故障的测试技术设计测试用例

在该类测试中，主要包括错误猜测、故障植入、修改测试三种，其中错误猜测是由测试人员根据给定的程序，依据自身经验和以前类似项目的记录大致猜测最有可能的故障；故障植入是目前非常通用的测试方法，通过人工地将故障植入程序并对其后的结果进行监控，找出程序中隐藏的缺陷；修改测试是指对程序稍作修改，即作一些语法上的修改(区别于原始版本)，让所有的测试用例在原始版本和修改后的版本上运行，如果测试用例能够分清程序与修改后程序之间的差别，就将后者删去。大量实践表明：修改测试可以找出程序中的语法故障，有的甚至是很复杂的故障。当然，在具体应用时，修改的程度要根据实际需要而定。

2.4 基于使用的测试技术设计测试用例

可以使用模拟仿真技术重新生成产品实际操作的环境，即通过列举程序可能的输入状态和它们发生的概率来建立测试模型，生成的测试模型是关于如何使用系统的一种定量特征描述。目前在软件可靠性工程测试技术里已经大量使用该技术。

3 基于软件测试工具保证软件测试的质量

3.1 软件测试工具简介

软件测试工具是提高软件测试质量必不可少的方面，可以充分弥补手工测试的先天性不足，给测试质量带来质的飞跃，常用的软件测试工具包括：功能测试工具、性能自动化测试工具、白盒测试工具以及软件测试管理工具，以性能自动化测试工具为例进行介绍。性能自动化测试工具包括两类：（1）负载性能软件测试工具。负载性能软件测试工具包括 QA Load、Silk Performer、Load Runner 与 WebRunner 四种：QA Load 主要包括服务器系统、客户系统、电商应用和资源配置等测试工具；Silk Performer 主要应用于工情况下工作；Load Runner 是适应性

较高，而且规模较大的负载测试工具，具有优化系统性能的作用；WebRunner 工具具有良好的软件测试性能和测试分析能力，能够使其自动进行压力测试。(2) 资源监控工具。资源监控是软件压力测试的一个关键步骤，多数软件测试工具是集成的，例如，Load runner 工具主要由反应时间、执行状况、图表分析、场景情况、Load、TPS、CPU 等集成。Nmon 这种资源监控工具在测试性能时，其信息记录较为全面，可以广泛应用于 Linux 和 AIX 操作系统之中，能够在操作过程中对系统资源的具体分布与使用情况进行定位、捕捉，还能将结果输入到文件之中，最终形成图形与数据文件。

3.2 如何利用测试工具保证软件测试质量

首先：强化测试人员的培训。软件测试工作者首先需要熟练掌握各类软件测试过程，了解软件的特点和应用范围、对象，熟悉软件缺陷及测试的故障多发位置，对高风险部位重点审核。同时，软件测试工作人员需要及时沟通交流技能，了解各类软件测试工具操作的注意事项。为了提高测试质量，测试人员必须接受相应的培训，深入了解检测过程、工具、方法，通过互相协调配合与有效交流而促进测试工作的顺利开展。

其次：提供自动化管理效果。测试质量的保证需要花费大量的资金与人力，特别是对系统复杂、信息量较大、分布地域广、平台异构的多组织协调进行测试时，保证质量较为困难，而跟踪测试流程的工具开发将大大提高软件测试质量。自动化软件测试工具进行检测时，需要制定各阶段工作计划和测试的量化标准，并在各部分匹配适合的软件测试工具，严格依据制度表执行，定期评审与测试。若软件未在预期时间内得到工作的审计与测试报告，就会向对应的实体部门传输消息提醒，并可以提醒工作人员检查延迟的原因。对于延迟时间较长的，软件就会自动发送警告信号，让测试工作人员具体分析产生延迟现象的原因。其原因可能是工作人员有限、经验缺乏、进度表设置不合理等，需要对员工及时加以培训。对于评审、测试过程存在的软件密度大或密度不一致问题，一般出错于开发过程之中，软件需要重新进行设计，当软件密度偏小时，说明缺陷测试效力较弱，同时加强员工培训工作。当模块或测试过程存在缺陷密度比平均值高的情况时，软件会自动发送警告，要求软件测试工作人员查找出错原因。在软件测试时间明显超过进度安排的情况下，系统也会发出警告信息，要求操作人员及时处理和调整，从而避免项目工程和延迟。

4 基于软件测试过程管理保证软件测试的质量

软件测试并不仅仅是软件生存周期中软件编码完成后的一个或几个阶段，它已成为一个全过程的验证和确认活动，贯穿于整个开发生命周期的始末。因此，对于一个软件企业来说，建立管理完善的软件测试过程并使之不断改进是非常重要的同时值得注意的还有软件测试过程中伴随的风险分析和风险管理。

4.1 全程软件测试管理保证软件测试质量

采用全程软件测试管理，测试人员贯穿需求阶段、开发阶段、发布阶段和运营阶段这四个阶段，开展测试活动。

1) 需求阶段测试

在需求阶段，开发人员对用户的需求进行分析、整理、估时，测试人员应参与其中，检查功能性需求和非功能性需求是否描述清晰，其中可以将非功能性需求作为验收要点，测试人员要发挥作用，减少含混性需求留到开发阶段，同时协助开发人员做好时间估算。

2) 开发阶段测试

在此阶段，开发人员要进行架构评审、功能要点确认、编码开发、单元测试、开发自测、Bug Fix 等工作，而作为测试人员通常要先进行功能确认，即在开发人员进行编码前，针对需求处理的用户需求，与开发人员进行确认，修正理解偏差，确保需求理解一致。接下来进行测试用例设计、用例评审、测试探索 包括功能测试、Bug Tracking、回归测试、系统测试、验收测试)，另外，测试人员应每天发布时间进度表，让团队了解当前进度情况，总结问题，寻求耗时超过预期时间任务的解决办法。每个项目在开发/测试阶段中，最好能构建缺陷经验库，避免多走弯路，测试人员还可提供相关 checklist 帮助开发人员在编码过程中关注开发自测的要点，从而提升质量。

3) 发布阶段测试

当一个项目处于发布阶段，开发人员进行上线申请、上线部署、服务监控时，作为测试人员应完成验收测试，提供测试报告，给出测试数据度量。另外，测试人员还有一项重要工作，就是对当前版本的缺陷进行统计分析。总之测试人员在此阶段应当持续反馈、改进、总结每个版本发生的问题，并对缺陷进行分析，总结出一些规律，帮助开发人员建立良好的习惯，改进代码的质量。

4) 运营阶段测试

运营阶段并不是终止阶段，此时开发人员进行故障登记，而测试人员则要进行版本问题反馈和改进提议：对软件运营发生的问题，总结反馈，提出改进建议，并且跟踪实施；生产故障分析：协助开发排查生产故障，避免测试场景的遗漏。全程软件测试实践，强调的是贯穿每个阶段的测试活动，不论是开发、还是测试，要理解双方的活动价值，什么时候该做什么事情，什么事情该做到什么程度才算好，保证每个环节的质量，才能够保证产品的全程质量。另外产品质量不是测试出来的，而是构建过程中沉淀下来的，开发人员的素养、测试人员的素养，以及团队对开发测试过程的重视程度，决定了产品质量。

4.2 测试风险管理保证软件测试质量

风险是一种不确定的事件或条件，一旦发生，会对至少一个项目目标造成影

响。引起风险的原因有很多，要根据不同的类型来确定最佳的方法以解决风险造成的不良影响。

风险管理包括风险管理规划、风险识别、风险分析、风险应对规划和风险监控等各个过程。软件测试中的风险管理的目标在于提高测试中积极事件的概率和影响，降低测试中消极事件的概率和影响。对于已识别出的风险，需要分析其发生概率和影响程度，并进行优先级排序，优先处理高概率和高影响风险。其中，风险识别是系统化地识别已知的和可预测的风险，才能提前采取措施，尽可能避免这些风险的发生，最重要的是降低风险可能造成损失的程度；风险评估是对已识别风险的影响和可能性大小的分析过程。从经验来看，许多最终导致项目失败、延期、客户投诉的风险都是从不起眼的小风险开始，由于这些小风险长时间得不到重视和解决，最终导致了项目的交付。

5 总结

软件质量是软件产品的生命力，而软件测试是保证软件质量的重要方面。本文从当前软件测试中存在的问题出发，从软件测试技术，软件测试工具以及软件测试过程管理这三个角度阐述了如何在软件开发过程中保证软件测试的质量。此外，我本人也正在从事软件测试的实习工作，根据我近两个月的实习经历以及在学习工作中遇到相关问题，我认为在时间紧，任务重的条件下，保证软件测试的质量应该更注意以下几点：

1) 项目成员明确需求，需求按优先级排序，评审之后少做变更。要做到需求按优先级排序，把需求分解成具体的最小级别的功能点，先实现高优先级的需求。三方评审通过后，项目中冻结需求，尽量少做变更。

2) 制定合理的测试计划，明确里程碑时间和负责人。测试计划是指导测试行动的总纲领，规划好测试设计，用例编写，测试执行的时间，测试负责人每天关注进展，及时调配资源，将问题解决在萌芽状态。

3) 保证测试设计和用例的质量。资深的测试工程师负责测试设计；按测试组成员能力水平分配任务，完成用例设计。完成之后，进行测试组和项目组的评审，查漏补缺。

4) 提高测试介入的标准。时间紧张，需要开发保证代码质量，测试介入的标准肯定是必须通过冒烟测试。冒烟用例评审时一定找开发确认，开发自己先执行成功冒烟用例，保证测试介入后能顺利走下面的流程。

5) 迭代测试。开发迭代提交模块，测试针对性进行测试。迭代测试增加了测试时间但是并没有延误整个的时间进度，因为在每一个迭代过程中测试过程都是提前开始的。

6) 每天召开晨会，沟通项目进度，解决问题。介入测试时，开发，测试，PM等团队成员每天花半小时召开晨会，沟通各自的进展，列出项目中的问题，确认

解决人和解决时间。问题及时解决，会加深团队伙伴的信任，激发工作热情。

当然，测试工作最终还是基于代码质量的，当我们低估项目复杂度的时候，增加测试时间才是明确的选择。欲速则不达，着急冒进，项目的质量很难得到保障，压缩的时间迟早会补偿回去。总之，软件测试的工作模式正在随着软件开发模式的改进而发生变化，保证软件测试质量却是永恒不变的主题，作为测试工作人员，也应该与时俱进，掌握更多更先进的技术，实现更符合用户需求、更健壮的软件产品。