

如何保证软件测试的质量

14126074 陈曼曼

软件产品的质量取决于软件开发过程，软件测试是提高软件质量最直接、最快捷的手段，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。测试不仅仅局限于软件开发中的一个阶段，而是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，其目的就是尽可能的提高软件产品。

1. 影响软件测试的因素

软件测试是发现软件项目中的错误和缺陷并且对项目质量进行评估的过程。测试工作从软件项目需求开始进行，并一直持续到产品发布以及产品的后期维护，这个工作过程中涉及到的很多因素都有可能影响测试工作的质量，其中包括技术因素，主观因素和客观因素。主观因素是指软件测试中涉及到的人员以及这些人员对软件测试的主观认识，他们的工作态度，对测试工作的理解等；客观因素是指测试时间是否充分，测试环境是否符合测试需要，投入资金能否支持软件工作的正常进行，被测软件项目是否属于保密范畴，项目信息是否对测试人员公开等；技术因素是从测试的专业角度分析，测试项目的组织、测试计划的建立、测试的设计和和实施等是否影响测试质量。

2. 提高软件测试质量的方法

2.1 测试原则

(1)测试人员从项目开始时介入项目，测试是贯穿产品整个生命周期的活动；(2)所有的测试部应追溯到用户需求，并且验证需求文档是否真正符合用户的要求；(3)当需求明确时，测试团队就应该针对需求设计相应的测试计划和方案，把“尽早地和不断地进行测试”作为测试人员的座右铭；(4)完全测试是不可能的，测试需要终止；(5)测试无法显示软件所有的缺陷；(6)充分重视测试中的集群现象。

2.2 项目的组织

软件测试是软件开发过程中独立的项目，应有专门人员负责测试，并配备管理人员对测试进行组织管理。由于思维定势，人们难于发现自己的错误，应避免系统开发人员出现在测试团队中，从而出现开发人员测试自己程序的现象，负责测试的人员除了掌握必需的测试技术和方法外，应具有相关专业的知识背景，能够透彻地理解被测对象及其工程背景，可以针对实际情况解决问题，应选择有工程管理能力和技术管理能力并且具有测试管理经验的技术人员负责项目管理。除此之外，应在测试团队中指定专人负责与软件项目中的其他人员沟通交流。

2.3 测试计划

软件测试是有组织、有计划、有步骤的活动，为了避免测试的随意性，应由测试负责人制定测试计划，项目测试团队按照此计划进行测试工作。测试计划应包括测试目的，测试对象，测试内容，测试方法，资源配置，进度安排，风险评估，人员分工。

2.3.1 测试目的

每个测试项目都有自己的针对性，是侧重于软件符合用户需求的程度，还是在软件发布之前尽可能多的找出软件缺陷，或者是对软件质量进行评估，这些都需要在制定测试计划时明确指出。

2.3.2 测试对象

“软件=程序+数据+文档”已被业界所公认，那么对软件进行测试，就不仅仅是对程序的测试，需求规格说明书、设计说明书以及源程序都应成为软件测试的对象。软件测试存在于软件各个开发阶段中，每个阶段性产品质量合乎要求，软件项目的最终质量才有保证。

2.3.3 测试范围

测试范围描述了测试工作涉及的范围包括软件的功能测试，性能测试，或负载压力测试、兼容性测试等。

2.3.4 测试内容

测试内容包括软件或项目的具体测试点，如用户登录，数据库连接等。

2.3.5 测试方法

根据测试项目的内容，测试的目标以及项目实际的工程条件选择测试的方法。根据软件开发阶段的不同，测试可分为单元测试、集成测试、确认测试和系统测试等；按照测试技术来划分，又分为黑盒测试和白盒测试两种测试方法。黑盒测试是验证软件需求的满足度和功能正确性的最有效的方法，其中包括等价类测

试、边界值测试、基于决策表的测试等。针对变量的不同形式选择不同的测试方法。(1)如果变量引用的是物理量或者变量是独立的，可采用边界值测试和等价类测试；(2)如果变量不是独立的，可采用决策表测试；(3)如果可保证是单缺陷假设，可采用边界值和健壮性测试；(4)如果是多缺陷假设，可采用最坏情况测试，健壮最坏情况测试和决策表测试；(5)如果程序包含大量的例外处理，可采用健壮性测试和决策表测试；(6)如果变量引用的是逻辑量，可采用等价类测试和决策表测试。如果时间等资源允许，还应采用自盒测试对程序进行测试，从程序的内部结构出发，检验程序中的逻辑路径是否都能按照预定要求正确工作。

2.3.6 资源配置。

资源配置包括软、硬件设备以及网络运行环境，尽可能地使测试环境与开发环境相同，最好避免测试与开发共用一个资源环境，以免在进行测试时对已开发项目造成影响。

2.3.7 进度安排。

根据测试项目的规模和测试的工作量，对需求分析、测试设计、测试执行及总结等做出时间规划。

2.3.8 风险评估。

风险评估中标明可能出现的风险，包括进度滞后或出现非计划事件等。在测试工作进行的过程中采取有效的办法降低被标明的风险发生的可能性。

测试计划完成之后，应组织人员对计划进行评审确保测试计

划明确了测试的内容、范围以及方案，测试工作的起始时间，测试人员安排等，其中测试方法是否合理有效，是评审的重点。

2.4 测试设计和执行

2.4.1 测试设计

测试设计主要指测试用例的设计与开发。测试人员应以软件需求说明、产品说明书等文档为依据，根据所选择的测试方案的来开发测试用例。测试用例应包括以下信息：用例编号、待测试的功能、测试环境及条件、测试输入、执行步骤、预期输入、实际输出以及评价输出结果的准则。测试人员在进行用例设计和开发时，应注意以下几点：①充分理解用户需求，使测试用例覆盖需求文档中的所有功能，并且包括代表软件工作任务的功能组合；②开发有代表性的测试用例，使其可以反映各种合理和不合理、正常和越界等情况；③测试结果必须是可判定的，即执行了测试用例执行后结果的正确性是可以判定的或可以评估的；④测试结果必须是可再现的，即面对同样的测试用例，其执行结果应当是相同的；⑤对不大可能出现缺陷的功能进行用例设计和开发是没有意义的。设计开发完成之后，应对测试用例进行评审，检查用例是否正确，设计是否充分，有无遗漏或是冗余。评审之后执行测试用例，进入测试执行阶段。

2.4.2 测试执行

(1) 使用恰当的测试技术

目前的测试措施种类非常多，都有着一定的优缺点。不过任何一

个措施都不能够体现出全部的测试规定，所以要积极的分析这些措施，明确其存在的优点和缺陷，结合体系的规定来细致的组合，通常可从如下两个层次中入手：

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。

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)测试过程

测试执行时应合理安排测试用例的执行顺序，对测试过程进行记录，全面观察测试用例的执行结果，即使实际输出结果与预期输入结果一致，也要查看软件的操作日志、系统运行日志及系统资源使用情况

况等来判断用例是否执行成功。若实际输出结果与预期结果不同，则应立即确认发现的问题，详细描述缺陷发生时的具体情况，包括软硬件配置、网络环境、测试用例输入、操作步骤、程序输出、当前输出设备的输出以及相关日志等信息。提供缺陷的详细信息便于重现问题，有利于开发人员分析缺陷产生的原因，定位错误位置，然后改正错误。随着软件中的缺陷和错误被测试到并改进，测试用例也应随着软件的实际情况而更新。测试人员应随时删除过时的和冗余的测试用例，对不受控制的用例进行改进，并增添新的用例。将有效的测试用例保存，可以在以后测试相同功能时直接利用，节省时间和资源。

(3)建立可复用的测试用例库

在整个软件测试的执行过程中，测试用例应该被保存并加以管理。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。在测试的时候要设置以复用为前提的用例，而且通过有效的管理措施，来提升其功效。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)对库里的用例合理的管控，经由提供的查询措施,来保证复用性,对于不一样的类型的用例开展复用次数的记载,这样能够帮助工作者获取有用的信息内容,在确保品质的背景之下,提升了测试的功效。3)对库里的用例进行合理的复用，经由查询用例,获取具体的信息，对于不同的用例使用不一样的复用模式来开展复用活动。

3. 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法。如何进行充分而有效的软件测试，成为软件测试领域关注的重要问题，只有明确测试对象，将测试与开发过程的各个阶段紧密结合起来，才能保证开发出高质量的软件。