

北京交通大学软件质量保证作业



课程名称： 软件质量保证

任课老师： 田思源

学生： 周豪

学号： 14126197

联系电话： 15650782728

2015 年 5 月 23 日

如何保证软件测试的质量？

引言

随着社会的不断进步和计算机科学技术的飞速发展，计算机及软件在国民经济和社会生活等方面的应用越来越广泛和深入。作为计算机的灵魂，软件在其中起着举足轻重的作用。软件的失效有可能造成巨大的经济损失，甚至危及人的生命安全。例如，1996 年 Ariane 5 运载火箭的发射失败等都是由软件故障引起的。

软件开发的各个阶段都需要人的参与。因为人的工作和通信都不可能完美无缺，出现错误是难免的。与此同时，随着计算机所控制的对象的复杂程度不断提高和软件功能的不断增强，软件的规模也在不断增大。例如，Windows NT 操作系统的代码大约有 3 200 万行。这使得错误更可能发生。人们在软件的设计阶段所犯的错误是导致软件失效的主要原因。软件复杂性是产生软件缺陷的极其重要的根源。

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，从最初的代码调试开始，一路跟随着软件的发展不断具体和深化，已经成为软件产业中不可或缺的一部分。测试工作的执行过程，直接影响软件的质量，进而影响软件产品的生命力。现如今软件规模越来越大，功能越来越复杂，软件测试所涉及的领域也越来越广，需要投入的人力、物力、财力及技术等资源也越来越多，如何正确理解软件测试的重要性，如何充分而有效的进行测试，如何控制软件测试的质量来保证软件产品的质量和可靠性，是目前软件测试领域的值得探讨的重要问题之一。

软件测试简介

软件测试是使用人工或自动手段运行来测试软件或系统的过程，是以评价一个程序或者系统为目标的活动。软件测试保证被测对象做了用户所期望的事件，并且确认被对象以正确的方式来做了这些事情。

软件测试的目的和意义：1) 以最少的人力、物力、时间等资源，找出软件中各种潜在的错误和缺陷，通过修正各种错误和缺陷，规避软件发布后由于软件错误和缺陷造成的隐患所带来的风险；2) 对软件质量进行度量和评估，以验证软件的质量满足用户的需求的程度，为用户选择与接受软件提供有力的依据；3) 对软件生命周期进行质量控制，通过分析错误和缺陷的分布以及它们的产生原因，帮助开发人员提高软件开发质量，帮助项目管理人员改进软件过程。

软件测试是保证软件质量和可靠性的重要手段。目前许多项目的软件工程实践以结构化分析和设计为核心，在开发阶段的前期，包括需求分析和设计都是以技术评审和工程管理作为质量保证的手段，而技术评审和工程管理主观因素很大，很可能又引入错误并

扩展到后续开发阶段。

另一方面，软件测试确实能够发现软件中隐藏的许多缺陷。例如，在英国约克大学为英国海军开发的SHOLIS项目中，尽管采用形式化方法描述和证明软件规约，并且采用程序正确性证明方法排除了软件开发前期的许多缺陷，单元测试仍然发现了整个软件开发过程中很多的缺陷。

随着人们对软件测试重要性的认识越来越深刻，软件测试阶段在整个软件开发周期中所占的比重日益增大。现在有些软件开发机构将研制力量的40%以上投入到软件测试之中；对于某些性命攸关的软件，其测试费用甚至高达所有其他软件工程阶段费用总和的3到5倍。

尽管人们在软件开发过程中也采用形式化方法描述和证明软件规约，并采用程序正确性证明、模型检验等方法保证软件质量，但是这些方法都存在一定的局限性，尚未达到广泛实用阶段。因此，程序代码最终体现了软件的质量，无论是从软件开发方法学还是软件测试自身的效益看，软件测试在今后较长时间内仍将是保证软件质量的重要手段。

软件测试的原则主要有以下8点：(1)所有的软件测试都应追溯到用户需求；(2)尽早的和不断的进行软件测试；(3)执行测试计划，尽量避免测试的随意性；(4)应当对每一个测试结果做全面检查；(5)测试用例应由测试输入数据和对应的预期输出结果这两部分组成。在设计测试用例时，应当包括合理的输入条件和不合理的输入条件；(6)程序员应避免检查测试自己编写的程序；(7)充分注意群集现象；(8)妥善保存测试计划、测试用例、出错统计和最终分析报告，为维护提供方便。

影响软件测试工作质量的因素

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。这里讨论如下几个因素。主观因素，客观因素，项目管理人员，开发人员，测试人员。

主观因素：

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是测试活动中的主观因素。测试涉及的人员包括：项目管理人员，项目开发人员，项目测试人员，技术因素等。

客观因素：

客观因素指将人们对软件测试的主观认识排除之外测试工作面对的客观实际情况，包括测试时间是否充分，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。客观因素对软件测试的约束也影响着测试工作的展开。

项目管理人员：

项目管理人员作为软件项目的管理者，他们对软件测试的态度决定了测试在项目中所占的比重，从此可以推断出测试工作的质量层次。如果项目管理人员没有充分认识到测试的重要性，则会影响测试在整个软件项目过程中的被关注度和重视程度，导致测试需要的人力、物力、时间等资源不足，直接影响甚至阻碍测试工作的进行，无法获得高质量的测试。

开发人员：

软件开发人员对测试工作的理解也是影响测试工作质量的一个因素。如果开发人员

对测试工作的理解不符合真实的测试情况，或是对软件测试的认识有误区，便会影响开发人员与测试人员的沟通合作，使测试人员不能顺利的进行测试工作，影响工作效率和质量。

测试人员：

测试人员是软件测试工作的具体执行者，找出软件中的错误及缺陷并且监督其改正是软件测试人员的职责。他们的工作态度和工作方法直接影响软件测试的质量。

技术因素：

除了主、客观因素的存在，从测试的专业角度分析，测试项目的组织、测试计划的建立、测试的设计和实施、回归测试等是影响测试质量的关键要点。

控制测试质量的要点

项目组织和测试原则

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。

项目组织：

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。

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) 所有的测试都应追溯到用户需求，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。

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。测试计划应包括以下内容：

测试目的：

完全的测试是不可能的项目，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，还是力求在软件发布之前尽可能多的找出软件缺陷，又或是对软件质量进行评估，需要在制定测试计划时明确测试目的，以确定测试的停止。

测试对象：

被测软件或项目的介绍，包括项目背景，软件功能等。

测试范围：

描述测试工作涉及的范围，是软件的功能测试，性能测试，或负载压力测试、兼容性测试等。

测试依据：

测试的依据就是用户需求的书面表现,包括软件项目立项说明书,软件需求说明书,产品说明书,用户手册及软件设计说明书等。

测试内容:

软件或项目的具体测试点,如用户登录,数据库连接等。

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,还应采用白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。

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

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。

风险评估:

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。

人员分工:

按照测试工作的内容和进度,安排测试团队中每个测试人员的工作。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。

测试设计和执行

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。

测试执行：

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。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。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。

回归测试

“缺陷解决”步骤是另一个引入错误（以及新缺陷）的机会[2]，这就需要在对旧代码进行修改之后，重新测试以确认修改没有引入新的错误或导致其它代码产生错误，这就是回归测试。回归测试不需要执行以前使用过的全部测试用例，而是识别出软件被修改的部分，在原来的测试用例中选择对新代码依然有效的测试用例进行执行。如果需要，还应根据实际情况开发设计心得测试用例，用来测试原有用例无法充分测试的部分。

测试总结

测试执行完成，缺陷修改及回归测试结束之后，应对测试工作做出总结。1）总结测试结果，对被测项目质量进行评估；2）评价测试团队的工作效率及工作质量；3）对测试过程中的经验教训进行总结。

小结

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。这里从软件测试的流程出发，探讨了控制测试工作质量的方法。