

软件测试组织与方法

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。为了尽可能多地找出程序中的错误，生产出高质量的软件产品，加强对测试工作的组织和管理就显得尤为重要。

一、 软件生存周期

一个软件从开始计划起，到废弃不用止，称为软件生存周期。一般来说，软件生存周包括计划、开发、运行三个时期，每一时期又可分为若干更小的阶段。计划时期的主要任务是分析用户要求，分析新系统的主要目标以及开发该系统的可行性。开发时期要完成设计和实现两大任务具体。具体分为需求分析、概要设计、详细设计、编码、测试。其中编码和测试是软件开发期的最后两个阶段。运行时期是软件生存周期的最后一个时期，软件人员在这一时期的工作，主要是做好软件维护。

统计表明，开发较大规模的软件，有 40% 以上的精力是耗费在测试上的，即使富有经验的程序员，也难免在编码中发生错误，何况，有写错误在设计甚至分析阶段早已埋下祸根，无论是早期潜伏下来的错误或编码中新引入的错误，若不及时排除，轻者降低软件的可靠性，重者导致整个系统的失败。为防患于未然，强调软件测试的重要性是必要的。

二、 测试的过程与方法

1、 测试的目的

在 G.J.Myers 的经典著作《软件测试技巧》中，给出了测试的定义：“程序测试是为了发现错误而执行程序的过程”。测试的目的是发现程序中的错误，是为了证明程序有错，而不是证明程序无错。在软件开发过程中，分析、设计与编码等工作都是建设性的，惟独测试是带有“破坏性”，测试可视为分析、设计和编码 3 个阶段的“最终复审”，在软件质量保证中具有重要地位。为了确保软件的质量，较理想的做法应该是对软件的开发过程，按软件工程各阶段形成的结果，分别进行严格的审查。

2、测试的过程

当设计工作完成以后，就应该着手测试的准备工作了，一般来讲，由一位对整个系统设计熟悉的设计人员编写测试大纲，明确测试的内容和测试通过的准则，设计完整合理的测试用例，以便系统实现后进行全面测试。

在实现组将所开发的程序经验证后，提交测试组，由测试负责人组织测试，测试一般可按下列方式组织：

(1)首先，测试人员要仔细阅读有关资料，包括规格说明、设计文档、使用说明书及在设计过程中形成的测试大纲、测试内容及测试的通过准则，全面熟悉系统，编写测试计划，设计测试用例，作好测试前的准备工作。

(2)为了保证测试的质量，将测试过程分成几个阶段，即：代码审查、单元测试、集成测试和验收测试。

(3)代码会审：

代码会审是由一组人通过阅读、讨论和争议对程序进行静态分析的过程。会审小组由组长，2~3名程序设计和测试人员及程序员组成。会审小组在充分阅读待审程序文本、控制流程图及有关要求、规范等文件基础上，召开代码会审会，程序员逐句讲解程序的逻辑，并展开热烈的讨论甚至争议，以揭示错误的关键所在。实践表明，程序员在讲解过程中能发现许多自己原来没有发现的错误，而讨论和争议则进一步促使了问题的暴露。例如，对某个局部性小问题修改方法的讨论，可能发现与之有牵连的甚至能涉及到模块的功说明、模块间接口和系统总结构的大问题，导致对需求定义的重定义、重设计验证，大大改善了软件的质量。

(4)单元测试：

单元测试集中在检查软件设计的最小单位-模块上，通过测试发现实现该模块的实际功能与定义该模块的功能说明不符合的情况，以及编码的错误。由于模块规模小、功能单一、逻辑简单，测试人员有可能通过模块说明书和源程序，清楚地了解该模块的 I/O 条件和模块的逻辑结构，采用结构测试（白盒法）的用例，尽可能达到彻底测试，然后辅之以功能测试（黑盒法）的用例，使之对任何合理和不合理的输入都能鉴别和响应。高可靠性的模块是组成可靠系统的坚实基础。

(5)集成测试：

集成测试是将模块按照设计要求组装起来同时进行测试，主要目标是发现与接口有关的问题。如数据穿过接口时可能丢失；一个模块与另一个模块可能有由于疏忽的问题而造成有害影响；把子功能组合起来可能不产生预期的主功能；个别看起来是可以接受的误差可能积累到不能接受的程度；全程数据结构可能有错误等。

(6)验收测试：

验收测试的目的是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。

经过上述的测试过程对软件进行测试后，软件基本满足开发的要求，测试宣告结束，经验收后，将软件提交用户。

3. 测试方法分析

集成测试及其后的测试阶段，一般采用黑盒方法。

(1)用边值分析法和（或）等价分类法提出基本的测试用例；

(2)用猜测法补充新的测试用例；

(3)如果在程序的功能说明中含有输入条件的组合,宜在一开始就用因果图法,然后再按以上(1)、(2)两步进行。

单元测试的设计策略稍有不同。因为在为模块设计程序用例时,可以直接参考模块的源程序。所以单元测试的策略,总是把白盒法和黑盒法结合运用。具体做法有两种:

a、先仿照上述步骤用黑盒法提出一组基本的测试用例,然后用白盒法作验证。如果发现用黑盒法产生的测试用例未能满足所需的覆盖标准,就用白盒法增补新的测试用例来满足它们。覆盖的标准应该根据模块的具体情况确定。对可靠性要求较高的模块,通常要满足条件组合覆盖或路径覆盖标准。

b、先用白盒法分析模块的逻辑结构,提出一批测试用例,然后根据模块的功能用黑盒法进行补充。

三、测试人员组织

人是测试工作中最有价值也是最重要的资源,没有一个合格的、积极的测试小组,测试就不可能实现。为高质高效地完成测试任务,好的测试工程师应具有如下能力:

1、沟通能力

一名理想的测试者必须能够同测试涉及到的所有人进行沟通,具有与技术(开发者)和非技术人员(客户,管理人员)的交流能力。既要可以和用户谈得来,又能同开发人员说得上话,不幸的是这两类人没有共同语言。和用户谈话的重点必须放在系统可以正确地处理什么和不可以处理什么上。而和开发者谈相同的信息时,就必须将这些活重新组织以另一种方式表达出来,测试小组的成员必须能够同等地同用户和开发者沟通。

2、技术能力

就总体言,开发人员对那些不懂技术的人持一种轻视的态度。一旦测试小组的某个成员作出了一个错误的断定,那么他们的可信度就会立刻被传扬了出去。一个测试者必须既明白被测软件系统的概念又要会使用工程中的那些工具。要做到这一点需要有几年以上的编程经验,前期的开发经验可以帮助对软件开发过程有较深入的理解,从开发人员的角度正确的评价测试者,简化自动测试工具编程的学习曲线。

3、自信心

开发者指责测试者出了错是常有的事,测试者必须对自己的观点有足够的自信心。如果容许别人对自己指东指西,就不能完成什么更多的事情了。

4、外交能力

当你告诉某人他出了错时,就必须使用一些外交方法。机智老练和外交手法有助于维护与开发人员的协作关系,测试者在告诉开发者他的软件有错误时,也同样需要一定的外交手腕。如果采取的方法

法过于强硬，对测试者来说，在以后和开发部门的合作方面就相当于"赢了战争却输了战役"。

5、幽默感

在遇到狡辩的情况下，一个幽默的批评将是很有帮助的。

6、很强的记忆力

一个理想的测试者应该有能力将以前曾经遇到过的类似的错误从记忆深处挖掘出来，这一能力在测试过程中的价值是无法衡量的。因为许多新出现的问题和我们已经发现的问题相差无几。

7、怀疑精神

可以预料，开发者会尽他们最大的努力将所有的错误解释过去。测试者必须听每个人的说明，但他必须保持怀疑直到他自己看过以后。

8、自我督促

干测试工作很容易使你变得懒散。只有那些具有自我督促能力的人才能够使自己每天正常地工作。

9、洞察力

一个好的测试工程师具有"测试是为了破坏"的观点，捕获用户观点的能力，强烈的质量追求，对细节的关注能力。应用的高风险区的判断能力以便将有限的测试针对重点环节。

总之，测试是软件生存周期中的一个关键的阶段，也是保证软件质量的重要活动之一。无论怎样强调软件测试的重要性和它对软件可靠性的影响都不过分，面对软件开发规模的增大、复杂程度的增加，更应高度重视软件测试工作的组织与管理，以提高软件质量。