

软件的测试与方法

陈翔鹰 陈英 彭一苇

(北京理工大学计算机系)

计算机硬件的飞速发展推动了软件开发,伴随着软件的发展,是软件的开发越来越困难,成本越来越高,管理越来越复杂,可靠性和质量越来越难于保证。六十年代末出现了所谓的“软件危机”。“危机”推动了人们从理论上对软件开发规律进行广泛和深入的研究,提出了以“工程”的方法来开发、维护和管理软件,以提高软件的生产效率和质量,产生了软件工程。

在软件工程的理论中,保证和提高软件质量的最重要的手段之一就是测试。软件测试在整个软件开发进程中占有重要的地位,测试所投入的人力和成本大约占整个项目开发费用的40%以上。

六十年代,国外的软件测试工作起步,进行了不同测试方法的探索;七十年代由于程序设计方法学的发展,为软件测试提供了新思想和新方法,推进了对软件测试的研究,逐步建立了软件测试的理论基础;自七十年代末期,软件测试走向了成熟阶段,不仅在软件测试的理论上和技术上,而且在应用的广度和深度上都取得了巨大的成功,软件测试进入了工程化、标准化和规范化阶段,建立了各种测试工具和环境,丰富和完善了软件测试的理论和方法,在软件开发中得到广泛的应用,使软件测试成为改进和提高软件质量有效和不可缺少的方法和工具。

我国对于软件测试理论和技术的深入研究始于八十年代初,目前已开发出一些软件测试工具。我国软件测试技术的应用还不广泛,虽然一些软件开发人员和项目管理人员知道软件测试的重要性的方法,但目前在实际的软件开发工作中,只有为数不多的软件系统是进行了较严格的测试。

软件测试作为软件开发过程中一个非常重要的阶段,是在精心控制的环境下,通过系统的方法检验程序,以发现程序中的错误。测试的目的不是证明程序中没有错误,也不是为了说明程序已正确执行了它的功能,而是要揭示程序中存在的错误。

测试是开发软件过程中一个必不可少的、有计划的步骤。其目的可以归纳如下:

- * 测试是为了发现错误而执行程序的过程;
- * 一个好的测试用例很可能是发现至今尚未察觉的错误;
- * 一个成功的测试则是发现至今尚未察觉的错误的测试。

后面的两句话象是文字游戏,但它的意义是深刻的,它意味着:好的测试就是要通过所设计的测试用例,暴露出程序中存在的各种潜在的错误。而“找不到程序中的错

误的测试,是成功的测试”的观点是根本错误的。

软件测试追求的目标是以最短的时间和最小的代价来系统地发现软件中的各种错误。首先是要预防错误的发生和扩大,在软件开发的初期就要对各种文档进行测试,尽可能早地发现错误,以减小错误对开发工作造成的损失。其次,是通过系统的方法发现程序中的错误。最后,是提供良好的错误诊断信息,以便于完善系统,改正程序中的错误。

为了进行正确有效的测试,测试过程和测试人员应遵循如下原则:

1.测试应贯穿开发过程的各个阶段,要尽早地不断地进行软件测试,以尽可能早地发现错误,特别是在系统需求分析和设计阶段。统计表明,编程中的错误只占全部错误的36%,软件开发的前期发生的错误必然会影响到后期工作,而在开发的后期再去发现并纠正前期的错误,要付出更大的代价。

2.程序开发人员不应测试自己的程序。从心理上说,程序员对自己的程序有偏好,总认为是不会有错误的,让他找出自己的错误是很困难的。若程序员对程序功能的理解本身就是错误的,那么自己进行测试是根本不可能发现的。

3.在设计测试用例时,不仅要有确定的测试输入数据,而且要给出预期的、确定的输出结果,以验证程序的输出是否与预期的正确结果一致。

4.设计测试用例不仅要有合理的输入数据,而且要有不合理的输入数据。因为程序开发人员往往忽视了对于不合理输入的处理,实际系统运行中经常因为输入非法数据而使程序发生故障。使用预期的不合理输入数据进行测试,会比合理的数据收获更大。

5.不仅要检查程序是否完成了应做的工作,还要检查它是否做了不该做的工作。因为多余的工作不仅无益,而且有害。

6.程序中发现错误越多的地方,也就是存在错误可能性越大的地方。应重点检查。

7.保留全部测试用例,做为软件文档的组成部分之一,当程序改错或改进之后,需要进行重新测试时,可再次使用。

对软件测试中发现的错误,可以按照错误的性质进行分类,以便于对错误出现的规律进行分析研究,防止开发过程中再发生类似的错误。软件中的错误可分为两类:

第一类为性能错误。这是由于受时间和空间的限制,程序不能正常运行所导致的错误。

软件测试并不研究和分析这类错误。

第二类为逻辑错误。这类错误与时间和空间无关。它们又可细分为:

1.问题定义错误。又称为需求分析错误。这是由于在软件开发的最初阶段,分析人员对于问题的定义不能满足用户的要求,是问题的范围、目标的说明和各种分析报告中的错误。

2.说明书错误。指需求说明书同问题定义不一致所导致的错误。这种错误发生在需求分析阶段,包括:

- * 不一致性错误:说明书中的功能说明有矛盾;
- * 冗余性错误:说明书中有些功能是多余的;
- * 不完整性错误:缺少必要的功能说明;
- * 不可行错误:有些功能要求是不可行的;
- * 不可测试错误:有些功能的测试要求是不现实的。

3.设计错误。发生在设计阶段,表现为系统的设计同说明书的功能说明不符。包括:

- * 设计不完全:有些功能没有设计,或设计不完全。
- * 算法错误:算法选择不合适。算法的基本功能不满足功能要求,算法不可行或算法效率不符合要求。
- * 模块界面(接口)错误:模块结构不合理;模块与外部数据库的界面不一致;模块之间的界面不一致。
- * 模块界面(接口)错误:模块结构不合理;模块与外部数据库的界面不一致;模块之间的界面不一致。
- * 控制逻辑错误:控制流程与说明书不一致;控制结构不合理。
- * 数据结构错误:数据设计不合理,与算法不匹配;数据结构不满足说明书的要求。

4.编码错误。编程过程中的错误是多种多样的,可分为以下几种:数据说明错误,数据使用错误,计算错,比较错,控制流错,界面错,输入输出错和其它错误。

软件在开发过程的不同阶段所采用的测试方法是不同的。

在需求分析和设计阶段,评审和审查是保证开发初期质量的唯一有效的方法。评审或审查是以需求分析文档、设计文档和测试计划等为对象,由分析设计人员、质量保证人员、专家和用户等有关人员参加,对文档进行详细评审。评审和审查是一种静态的测试方法。

在编程阶段除了可以进行评审、审查之外,还可以采

用代码走查和动态测试的方法。代码走查就是用人工的方法模拟程序的运行,发现程序中的错误,也是静态的测试方法。动态测试是通过在机器上运行实际的程序来寻找错误。

动态测试的基础是程序的功能说明或程序的逻辑结构。方法有两种:黑盒测试法和白盒测试法。

如果已知程序的功能,测试它的每一个功能是否都达到了预期的要求,这种方法叫黑盒测试。黑盒测试又称为功能测试。它着眼于软件的外部特性,将程序看成一个黑盒子,完全不考虑软件的内部逻辑结构。仅检验程序结果和设计说明书的一致性,测试用例的设计完全从设计说明书出发。黑盒测试只在软件的接口上进行测试,检验其能否满足功能要求,能否正确地接收输入,并正确地输出结果,以及能否保持外部信息(如数据文件)的完整性等。

如果要用黑盒法来发现程序中的全部错误,必须对全部可能的输入进行检查,看程序是否都产生正确的输出。采用穷举的方法进行测试是不现实的,为了进行黑盒测试,常用方法有三种:

1.等价划分。将所有可能的输入划分为若干等价类,采用等价类中的代表值对程序进行测试。若程序对于某个等价类中的一个测试用例运行发现错误,则这个等价类中的其它测试用例也会查出同样的错误;若程序对于某个等价类中的一个测试用例没有发现错误,则这个等价类中的其它测试用例也不会查出错误。

2.边界值分析。在设计测试用例时使用正好等于、小于或大于边界值的数据进行测试。它着重于检查等价类边界上和边界附近的情况。

3.图形技术。借助图形表明条件与对应动作之间的逻辑关系,或者表明程序过程路径元素,以便进行测试用例的设计。因果图就是设计测试用例的一种工具,它着重检查各种输入条件的组合。程序图是分析软件过程流向的通用工具,在设计测试用例时,有利于选择测试,分离程序的路径。

如果已知程序的内部活动方式,测试它的内部活动是否都符合设计要求,这种方法叫白盒测试。白盒测试又称为结构测试或逻辑覆盖。与黑盒测试不同,它与程序的内部结构密切相关,着力于检查软件的内部逻辑结构,以检查程序的内部细节为基础,通过提供一组指定条件和循环的测试用例,对穿过软件的逻辑路径进行测试,可以在不同点检查程序的状态,以确定实际状态与预期状态是否一

致。

穷举所有测试路径,进行彻底的白盒测试是根本不可能的。白盒测试的基本测试方法有:

1.语句覆盖:要求每个语句都至少执行一次;

2.分支覆盖:要求程序中的每条分支路径至少执行一次;

3.条件覆盖:要求使分支的各种可能的条件至少执行一次;

4.条件分支覆盖:要求不仅各个分支中的所有可能取值至少出现一次,而且它们的各种可能的组合也至少出现一次。

5.循环覆盖:检查循环结构中控制条件中的错误。

在实际的测试工作中,通常将白盒测试法(逻辑覆盖)、等价划分和边界值分析测试结合使用。将白盒测试和黑盒测试技术结合起来,既可以检查设计的内部要求,又可以检查设计的接口要求。

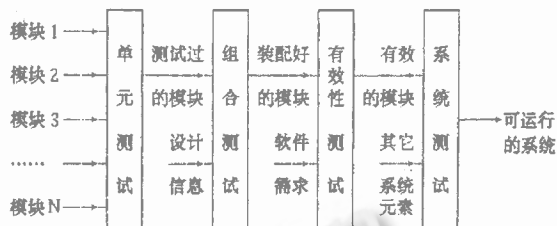


图 1

对于软件系统的测试,在软件工程的范围内实际上可分为四步:单元测试、组合测试、有效性(功能)测试和系统测试。它们的关系为:

单元测试是集中力量检验软件设计的最小单位——模块,目的是找出与模块内部结构有关的全部错误。它以详细设计为指南,对重要的控制路径进行测试,以发现编程错误。单元测试应当使用白盒测试法。

在单个模块通过单元测试之后,要将通过测试的模块进行集成,在集成过程中要进行组合测试。组合测试在软件装配的同时进行测试,以发现与接口相联系的问题,目的是将通过单元测试的模块构成一个符合设计要求的软件结构。组合测试以概要设计为指南。模块集成的方法可以采用自顶向下结合和自底向上结合两种方法,组合测试时也同时要用与模块集成方法相同的测试方法。

对于集成的软件要进行有效性测试。有效性是指软件功能与用户要求的一致性。软件的有效性测试是通过黑盒测试,证实软件功能与用户要求一致。根据测试计划验证软件的各种功能性能要求,检查文档资料的正确性和完整性,以及系统的可移植性、兼容性、错误恢复功能和可维护性等。

在最终将软件与其它系统元素(如新的硬件和信息、其它的外部系统等)结合在一起时,要对整个完整系统的组合和功能进行系统测试。系统测试的部分内容已超出了软件工程的范围,它包括:

- * 功能验证:对系统功能进行验证。
- * 繁忙测试:使系统的全部资源处于繁忙工作状态,进行企图“摧毁”系统的测试。
- * 装入和性能测试:为了保证系统满足性能目标的测试。

背景测试:采用实际负载取代无负载的重复测试。

配置测试:保证全部逻辑或物理设备在指定组合下实现全部功能的测试。

恢复测试:测试系统能否在硬件或软件故障时进行恢复,而没有丢失或损失数据。

安全测试:为保证系统的安全机构不会被系统非法用户破坏的测试。

验收测试:由用户完成正式验收计划,执行正式验收测试。

单元测试中发现的错误主要是编码阶段产生的错误,错误的范围较小,一般只涉及一个独立的模块,修改比较容易。组合测试发现的错误主要发生在设计阶段,会涉及到一组相关的模块,可能会引起这一组相关模块的修改或功能调整,修改的工作量会成倍增长。有效性测试主要发现分析阶段的错误,分析阶段的错误会涉及到软件的全局性问题。发现错误之后,常常需要回到编码、开发和分析等阶段,要进行再编码、再开发、再分析和再测试。在大多数情况下,要纠正系统测试中发现的错误,是非常困难的,因为系统测试发现的错误会涉及到用户的最终需求。最初的一个错误经过开发的各个阶段后,已经被放大了许多倍。如果这时发现是对用户需求的理解性错误,将是致命性错误,要纠正就需要付出极大的代价。

由于软件测试工作通常占软件开发项目总工作量40%左右,能够减少测试时间的工具(但不会提高测试的彻底性)是很有价值的。研究人员和测试人员已经开发出

大量的自动测试工具。常见的测试工具类型包括:

- * 静态分析器。用于分析程序的结构、模块调用关系、模块的控制结构及变量和参数的定义与引用情况。
- * 代码检查器。用来检查软件的质量,以保证程序满足基本的编码标准。
- * 断言处理器。检查程序员提供的有关程序行为的各种断言在程序实际运行时是否相符。

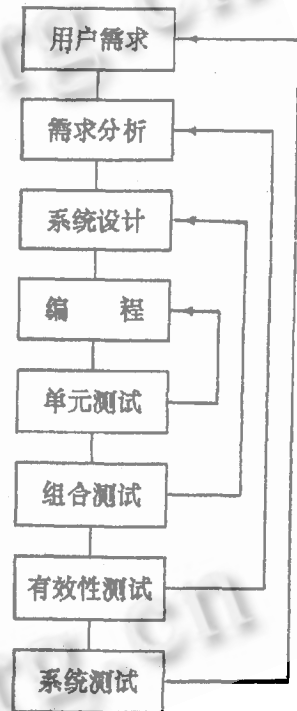


图2 测试与开发的关系

* 测试文件生成器。用预定的数据生成并填入程序输入文件,以便进行测试。

测试数据生成器。自动分析被测系统,帮助用户选择并产生测试数据。

* 动态测试器。在被测程序中安置“探针”,记录程序运行时的动态情况,获得程序内部结构的各种覆盖报告。

* 测试支持环境,支持测试处理,如安装被测程序,提供驱动模块和桩模块,输入测试数据,对测试用例进行管理,生成测试文档等。

输出比较工具。对程序的一组输出和先前得到的另

一组输出作比较,确定二者之间的差别。

* 数据流分析器。专门用于切片和追踪通过系统的数据流,找出未定义的数据引用、不正确的下标和其它有关数据异常的错误。

* 符号执行器。用代数符号而不是具体的数值作为输入来完成程序的测试。这种测试不需特定的测试用例,其输出结果也是特定的代数表达式,可与相应形式的期望结果进行比较。

* 环境模拟器。用于动态模拟实际运行环境,以适应特定的实时软件测试要求。

软件测试工作是软件系统开发过程中一个必不可少的重要的工作。在我国自行开发的第十一届亚洲运会计算机系统中,曾专门为此投放了大量的人力、物力和财力。在系统的需求分析和设计阶段,强调文档在整个工程中的重要地位,保证各项分析和设计文档的完整性和准确性,为后续的开发和测试工作打下良好的基础。为了保证工程的质量,建立评审制度,多次召开评审会,认真听取用户意见,成

立了专家组,听取专家的意见和建议。在编程阶段,为了提高模块的测试质量,在部分子系统的单元测试时,试用了我国自行开发的 C 语言测试工具。在组合测试中采用黑盒测试法,着重解决模块之间的接口,保证模块之间数据接口的正确性。在有效性测试时,专门聘请用户和第三方的非开发人员,对系统的功能和各项性能进行了详细全面的测试。在系统测试时,着重进行了系统级的功能验证、极限负载能力测试,意外故障恢复能力测试、安全性测试和配置测试,并在尽可能真实的情况下,进行了全面模拟实际运行情况的全系统彩排。正是这种严格全面的测试工作,发现了系统开发中存在的错误,找出了系统存在的隐患,保证了整个系统的高水平和高质量。

随着我国软件产业的发展,软件开发技术和水平的不断提高,软件测试的重要性和意义将会为更多的开发人员所认识,软件测试方法和测试工具的发展和广泛应用,会为提高软件产品的质量,缩短开发周期,降低开发费用,做出积极的贡献。