

软件测试技术概述

周 煜¹, 周国庆², 奚文骏³

(1. 海军驻上海地区航空军事代表室, 上海 200233; 2. 海军航空工程学院院务部, 山东 烟台 264001;

3. 海军航空工程学院军械工程系, 山东 烟台 264001)

摘 要: 介绍了软件测试的必要性和目的, 阐述了软件测试的静态测试、动态测试和黑、白盒测试法, 以及软件测试的一般过程和步骤, 并提出了软件测试的几个原则, 最后论述了几种新技术对软件测试的影响。

关键词: 软件测试; 质量; 错误

中图分类号: TP311.53

文献标识码: A

文章编号: 1672-4984(2005)03-0056-03

Summarize of software test technique

ZHOU Yu¹, ZHOU Guo-qing², XI Wen-jun³

(1. Navy Aviatric Military Representative Office in Shanghai Area, Shanghai 200233, China;

2. Department of Scientific Research, NAEL, Yantai 264001, China;

3. Department of Amament Engineering, NAEL, Yantai 264001, China)

Abstract: Introduce the necessity and purpose of software test. Expound several kinds of software test techniques such as static test, dynamic test, white box test and black box test. And then indicate the procedure and steps of test and several principles which should be followed. Finally, discuss the effects to software test brought by new technologies.

Key words: Software test; Quality; Error

随着软件产品在各个应用领域的普及, 软件的质量问题逐渐成为人们关注的焦点。质量不过关的软件产品在使用中可能出现各种意想不到的问题, 不但影响正常的操作、增加使用成本, 甚至会产生严重的或灾难性的后果。另一方面, 软件的质量问题也会增加软件开发商的维护费用, 影响其信誉和形象, 甚至导致经济或法律纠纷。因此不论软件用户还是开发商都希望尽可能提高软件的可靠性, 得到优质的软件。

然而, 完美无缺的软件是不存在的, 有错是软件的必然属性。导致软件错误的原因是多方面的, 开发商对用户要求认知的不深刻、软件本身的复杂性、软件开发工具存在的局限、软件开发中的人为错误、时间压力等等都会影响产品的顺利开发, 从而导致软件开发成本、进度和质量上的失控。

经过软件专家和相关人员的研究和分析, 发现软件的质量问题仅仅通过开发人员的认真和仔细是不能消除的, 程序语言的更新换代对提高软件的可靠性也收效甚微。随着软件工程学的兴起和软件开发的工程化, 人们认识到新的语言、先进的技术和完

善的管理仍然不能杜绝软件中的错误, 而软件测试则在这一工程化的过程中逐渐担负起评价和发现错误的责任, 成为软件开发的重要部分。统计表明, 在典型的软件开发项目中, 软件测试的工作量往往占软件开发工作总量的 40%, 并因此而开销 30% ~ 50% 的总成本, 其重要性不言而喻。

1 软件测试的目的

软件测试是一个破坏性的过程, 其目的是为了尽可能多地发现软件中的错误, 而不是为了演示软件的正确功能。软件测试不能保证软件中不再存在错误, 而且发现错误也不是软件测试的唯一目的, 查找不出错误的测试不等于没有价值。

首先, 测试并不仅仅是为了要找出错误。通过分析错误产生的原因和错误的分布特征, 可以帮助项目管理人员发现当前所采用的软件开发过程的缺陷, 以便改进。同时, 也有利于设计出有针对性的检测方法, 改善测试的有效性。

其次, 没有发现错误的测试也是有价值的, 完整的测试是评定软件性能和质量的一个重要手段, 可以用来检查软件是否满足设计和项目合同书所规定的技术要求, 检验软件对误操作的处理能力, 并为软件可靠性与安全性的评估提供依据。

因此软件测试的目的决定了如何去组织测试: 如果测试的目的是为了尽可能多地找出错误, 那么测试就应该直接针对软件比较复杂的部分或是以前出错比较多的位置; 如果测试的目的是为了给最终用户提供具有一定可信度的质量评价, 那么测试就应该直接针对在实际应用中会经常用到的商业假设。

2 软件测试的实施

2.1 测试方法

软件测试的方法原则上可以分为两大类, 即静态测试和动态测试。静态测试是对被测软件进行特性分析的方法的总称, 主要特点是不利用计算机运行被测试的软件, 而针对需求说明、设计文件等文档和源程序进行人工检查和静态分析, 以保证软件质量。静态测试能够有效地发现软件中 30% 到 70%^[1] 的逻辑设计错误和编码错误。动态测试是在计算机上实际运行被测试的软件, 通过选择适当的测试用例, 判定执行结果是否符合要求, 从而测试软件的正确性、可靠性和有效性。动态测试的两种主要方法是白盒测试和黑盒测试。

白盒测试是对软件内部工作过程的细致检查,

它允许测试人员利用程序内部的逻辑结构及有关信息, 设计或选择测试用例, 对程序所有逻辑路径进行测试。通过在不同测试点检查程序的状态, 确定实际的状态是否与预期的状态一样, 因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试一般选用最少量的可以有效揭露隐藏错误的路径进行测试, 所以如何设计软件测试用例是这种方法的关键。

黑盒测试则着眼于软件的外部结构, 不考虑程序的逻辑结构和内部特性, 仅依据软件的需求规格说明书, 在软件界面上检查程序的功能是否符合要求, 因此黑盒测试又叫做功能测试或数据驱动测试。用黑盒测试发现程序中的错误, 必须在所有可能的输入条件和输出条件中确定测试数据, 来检查程序是否都能产生正确的输出。

白盒、黑盒测试不能相互替代, 而应互为补充, 在测试的不同阶段为发现不同类型的错误而灵活选用。

2.2 测试过程

软件测试过程一般按四个步骤进行, 即单元测试、组装测试、确认测试和系统测试, 如图 1 所示^[1]。

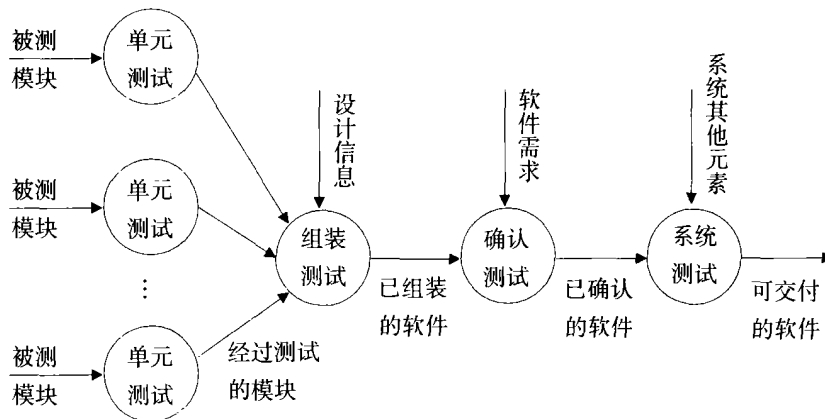


图 1 软件测试过程

单元测试是指依据详细的设计描述, 对每一个功能相对独立的程序模块进行测试, 检查各个单元是否正确地实现规定的功能。单元测试一般在完成某一程序模块的编程后由程序员立即进行, 主要对程序内部结构进行检验, 着重发现和解决代码编写过程中的差错, 多采用白盒测试法。

组装测试是指在将单元测试无误的程序模块组装成软件系统的过程中, 对程序模块间的接口和通讯方面的正确性的检查。通常可采用增式和非增式两种方法, 其中前者把一个待测试的模块组合到已经测试好的模块组上进行测试, 而后者则把通过单

元测试的模块组合成整个系统进行统一测试。组装测试一般在完成了软件的所有或大部分编码工作后, 由不同开发人员共同完成。

确认测试又称有效性测试, 即验证软件的功能和性能是否与用户的需求相一致, 以及软件配置是否完全正确。一般以软件的需求规格说明书为依据, 采用黑盒测试法。

系统测试是将通过确认测试的软件作为一个元素, 在实际运行环境中, 与计算机硬件、外设、某些支持软件、数据和人员等元素结合在一起, 对整个系统进行的测试。与前三种测试不同, 实施系统测试的

人员应是最终用户代表。

3 软件测试的几个原则

3.1 软件测试应贯穿软件开发的全过程

由于原始问题的复杂性、软件的复杂性和抽象性、软件开发各个阶段工作的多样性以及各种层次人员之间工作的配合关系等因素,使得开发的每一

个环节都可能产生错误。所以软件测试不仅仅是软件开发的一个独立阶段,而应当把它贯穿到软件开发的各个阶段中,坚持各个阶段的技术评审,才能尽早发现和预防错误。图 2 所示为软件测试的 W 模型^[3],形象地说明了软件测试与开发的同步性。

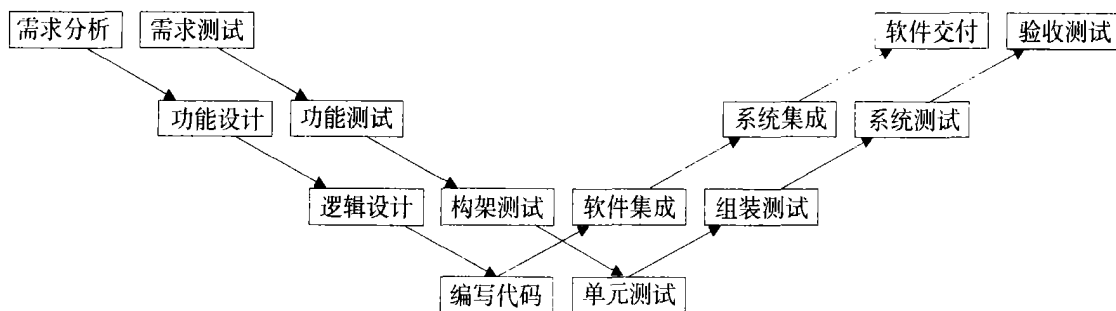


图 2 软件测试的 W 模型

3.2 合理选择测试用例

完整的测试用例不但需要测试的输入数据,而且需要对应这些输入数据的预期输出结果。如果对测试输入数据没有给出预期的输出结果,那么就缺少了检验实测结果的基准,就有可能主观地把一个似是而非的错误结论当成正确的结果。

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。合理的输入条件是指能验证软件的输入条件;不合理的输入条件则是指异常的、临界的、可能引起问题异变的条件。用不合理的输入条件测试软件能核实软件的容错能力和完全性,往往比合理的输入条件能发现更多的错误。

3.3 程序员和程序设计机构应避免测试自己的程序

测试工作需要严谨的作风,客观的态度和冷静的情绪,由开发组成员以外的人来完成软件测试会更客观,更有效,并更容易发现问题。这是因为:一方面,程序中可能包含一些由于对问题的叙述和说明误解而导致的错误,自己一般很难发现;另一方面,就心理上来讲,程序员总不希望自己的程序出错,因此不能有效测试自己的软件。

4 新技术对软件测试的影响

4.1 自动化技术对软件测试的影响

由于软件的人工测试越来越困难,而且容易出现人为错误,所以测试过程的自动化已经成为测试的发展方向。目前,在软件测试领域已经出现了许多有效的测试工具,能够自动生成测试用例、自动进行测试跟踪、自动分析测试结果,并且能够进行专业的测试管理。测试的自动化提高了软件测试的质量

和效率,方便了错误的检测与定位。

当然,目前自动测试技术还不是十分完善,在测试的针对性、灵活性和智能化方面与人工测试相比仍然存在不小的差距,不能完全替代人的工作。因此,如何将人工测试与自动测试相结合,设计更实用的自动化工具,是一个迫切需要解决的问题。

4.2 面向对象技术对软件测试的影响

面向对象的软件设计思想与技术的出现,对软件测试技术、特别是动态测试产生了较大影响。由于面向对象技术在面向过程的程序设计基础上引入了类的概念,使得数据与处理之间相对独立。与传统程序中“从上到下”、“分支”、“循环”的控制流不同,类与类之间处理的是一种“由外到内”的渗透顺序,同时在数据的定义与引用上也出现了“由内到外”的层次流向,从而产生了新的控制流和数据流^[3]。

相应地,软件测试也要增加新的内容。面向对象的测试是从类和对象的测试开始的,测试的方法不再是传统的输入输出模型,更加关心的是有效的动作序列,并要求引入“类测试”的概念。

4.3 CASE 技术与可视化技术对软件测试的影响^[3]

近几年 CASE (Computer Aided Software Engineering) 技术与可视化 (Visual) 技术在软件应用领域迅猛发展。由于 CASE 技术确保了定义的一致性,并帮助设计人员进行了某些设计工作,使设计人员可以把更多精力放在模型设计、界面划分、流程定义等关键之处,从而大大简化了静态测试的强度。由于支持可视化技术的软件经过了十分严格的测试和反复使用,与手工录入相比,正确性较高,因此 (下转第 75 页)

维护企业应用的基础数据, 在其他模块使用前应该首先将系统中涉及到的基础数据代码进行维护, 在其他模块选取使用。

(2) 系统维护子系统 维护整个项目的权限和日志信息, 通过系统维护子系统分配用户的登陆权限, 记录查看各种日志信息, 系统维护子系统是系统数据安全的基础。

(3) 工艺生产信息维护子系统 维护企业加工零件的加工工艺信息, 这些信息将作为生产部进行计划的数据依据, 在生产部进行生产计划前, 应该先维护零件的工艺生产信息。

(4) 生产部计划管理子系统 这个模块作为整个系统的关键, 生成各个工段的生产计划和仓库的出入库计划, 将作为分厂管理和库存管理的数据参考依据。

(5) 生产部库存管理子系统 负责对企业的出入库材料进行管理, 设计到企业客户, 企业供货商, 企业内部的各种出入库操作。

(6) 工序进展信息管理子系统 根据生产部下发的投交指标, 分厂根据下达的计划进行生产, 工序跟踪对分厂的实际加工情况进行跟踪记录。

(7) 质量信息跟踪子系统 分厂出现不同的质量损失, 通过质量信息跟踪对发生的质量信息进行记录跟踪, 降低公司的质量问题。

在整个系统中生产部计划管理是整个系统中最复杂的流程, 在这个流程中涉及到滚动作业计划, 零件投交指标, 加工路线单等单据和不同范围的计划信息。具体的处理流程和生成原则如下:

(1) 在生成工艺过程是首先得到当前生产产品的零件列表, 属于滚动作业计划中毛坯交库指标的

计划数据, 从产品零件列表中中查询得到存在毛坯加工工段信息的零件(利用零件分配表信息), 生成不同毛坯加工工段的交库指标, 如果零件没有毛坯工艺则不形成交库指标。(注: 零件类型为自制件(通用件, 借用件, 基本件))。

(2) 属于滚动作业计划中毛坯库投入零件加工车间的计划, 和产品零件列表中存在机械加工工段的零件进行运算(利用零件分配表信息), 生成不同的机械加工工段投料指标。如果零件没有零件加工工艺则不生成投料指标。

(3) 对属于滚动作业计划中零件交库的计划, 和零件投入计划一致, 生成交库计划, 只是两个的工艺过程标识不同。

(4) 对于投入装配车间的东西没有进行计划。实际上应该进行计划, 将产品零件列表中所有的东西都将作为投入计划。

(5) 生成过程中在生成毛坯交库指标时, 毛坯的编号要发生变化, 毛坯的编号要查询材料表中定义的这种零件的毛坯编号, 作为本次计划的交库指标毛坯的编号。

参考文献

- [1] John W. Satzinger Robert B. Jackson Stephen D. Burd 系统分析与设计[M]. 北京: 机械工业出版社, 2002
- [2] 潘尔顺. 生产计划与控制[M]. 上海: 上海交通大学出版社, 2003
- [3] 张毅. 企业资源计划(ERP)[M]. 北京: 电子工业出版社, 2001
- [4] 张文. ERP、CRM 企业实施案例[M]. 北京: 清华大学出版社, 2003
- [5] 林峰. 现代工业企业生产管理[M]. 北京: 经济科学出版社, 1997

(上接第 58 页)

对其产生的程序的测试只要集中于人工控制段上就可以了, 尤其是对界面的调试, 只要进行若干次运行即可, 这些都使得测试强度大大减弱。

5 结束语

实践证明, 作为保证软件质量的一种重要手段, 软件测试是必不可少的。随着计算机技术的发展和人们对软件质量要求的不断提高, 软件测试必将受到越来越多的关注和推广。当然, 软件测试是一项复杂的任务, 是一门综合的学科, 虽然目前已经有了

一些成熟的理论和成功的方法, 随着其应用领域的拓展和新兴技术的显现, 仍然需要不断的发展和完善。

参考文献

- [1] 薛赛男, 赵伟. 软件测试技术——计量测试技术的新领域[J]. 计量技术, 2003, 5
- [2] 吴东升. 软件测试的三种模型[J]. 廊坊师范学院学报, 2002, 12
- [3] 李锦伟, 马兆丰. 浅谈软件新技术的应用对软件测试的影响[J]. 交通与计算机, 2000, 4