



高等职业教育精品示范教材

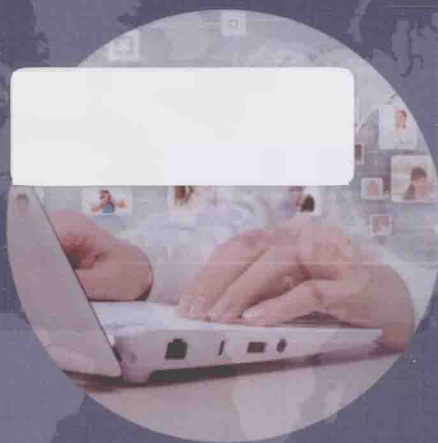
电子信息课程群

GAODENG ZHIYE JIAOYU
JINGPIN SHIFANJIAOCAI

软件测试技术

(第二版)

主 编 库 波 杨国勋
副主编 罗 炜 董 宁
主 审 王路群



中国水利水电出版社
www.waterpub.com.cn

高等职业教育精品示范教材（电子信息课程群）

软件测试技术

（第二版）

主 编 库 波 杨国勋

副主编 罗 炜 董 宁

主 审 王路群



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书介绍了实用的软件测试技术。全书共分为8章,主要包括:软件测试基础知识、软件测试阶段、软件测试过程与管理、黑盒测试、白盒测试、性能测试、Web应用测试和易用性测试等。

本书在软件测试技术内容的选取、概念的引入、文字的叙述以及案例和习题的选择等方面,都力求遵循面向应用、逻辑结构简明合理、由浅入深、深入浅出、循序渐进、便于自学原则,突出其实用性与应用性。

本书可作为高职高专学校的计算机专业教材,也适合作为各校非计算机专业辅修计算机专业课程的教材,还可供从事计算机软件开发的科技人员自学参考。

本书配有电子教案,读者可以从中国水利水电出版社网站和万水书苑免费下载,网址为:
<http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目(CIP)数据

软件测试技术 / 库波, 杨国勋主编. -- 2版. -- 北
京: 中国水利水电出版社, 2014. 8
高等职业教育精品示范教材: 电子信息课程群
ISBN 978-7-5170-2134-6

I. ①软… II. ①库… ②杨… III. ①软件—测试技
术—高等教育—教材 IV. ①TP311.5

中国版本图书馆CIP数据核字(2014)第123255号

策划编辑: 祝智敏 责任编辑: 张玉玲 加工编辑: 宋 杨 封面设计: 李 佳

书 名	高等职业教育精品示范教材(电子信息课程群) 软件测试技术(第二版)
作 者	主 编 库 波 杨国勋 副主编 罗 炜 董 宁
出版发行	主 审 王路群 中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658(发行部)、82562819(万水) 北京科水图书销售中心(零售) 电话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×240mm 16开本 15.25印张 337千字
版 次	2010年10月第1版 2010年10月第1次印刷 2014年8月第2版 2014年8月第1次印刷
印 数	0001—3000册
定 价	32.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

高等职业教育精品示范教材（电子信息课程群）

丛书编委会

主 任 王路群

副主任 雷顺加 曹 静 江 骏 库 波

委 员 （按姓氏笔画排序）

于继武 卫振林 朱小祥 刘 芊

刘丽军 刘媛媛 杜文洁 李云平

李安邦 李桂香 沈 强 张 扬

罗 炜 罗保山 周福平 徐凤梅

梁 平 景秀眉 鲁 立 谢日星

鄢军霞 綦志勇

秘 书 祝智敏

I

序

为贯彻落实国务院印发的《关于加快发展现代职业教育的决定》，加快发展现代职业教育，形成适应发展需求、产教深度融合、中职高职衔接、职业教育与普通教育相互沟通的现代职业教育体系，我们在围绕中国职业技术教育学会研究课题的基础上、联合大批的一线教师和技术人员，共同组织出版“高等职业教育精品示范教材（电子信息课程群）”职业教育系列教材。

职业教育在国家人才培养体系中有重要位置，以服务发展为宗旨，以促进就业为导向，适应技术进步和生产方式变革以及社会公共服务的需要，从而培养数以亿计的高素质劳动者和技术技能人才。紧紧围绕国家发展职业教育的指导思想和基本原则，编委会在调研、分析、实践等环节的基础上，结合社会经济发展的需求，设计并打造电子信息课程群的系列教材。本系列教材配合各职业院校专业群建设的开展，涵盖软件技术、移动互联、网络系统管理、软件与信息管理等专业方向，有利于建设开放共享的实践环境，有利于培养“双师型”教师团队，有利于学校创建共享型教学资源库。

本次精品示范系列教材的编写工作，遵循以下几个基本原则：

（1）体现就业为导向、产教结合的发展道路。学科和专业同步加强，按企业需要、按岗位要求来对接培养内容。既反映学科的发展趋势，又能结合专业教育的改革，且及时反映教学内容和教学体系的调整更新。

（2）采用项目驱动、案例引导的编写模式。打破传统的以学科体系设置课程体系、以知识点为核心的框架，更多地考虑学生所学知识与行业需求及相关岗位、岗位群的需求相一致，坚持“工作流程化”、“任务驱动式”，突出“走向职业化”的特点，努力培养学生的职业素养、职业能力，实现教学内容与实际工作的高仿真对接，真正以培养技术技能型人才为核心。

（3）专家教师共建团队，优化编写队伍。由来自于职业教育领域的专家、行业企业专家、院校教师、企业技术人员协同组合编写队伍，跨区域、跨学校来交叉研究、协调推进，把握行业发展和创新教材发展方向，融入专业教学的课程设置与教材内容。

（4）开发课程教学资源，推进专业信息化建设。从充分关注人才培养目标、专业结构布局等入手，开发补充性、更新性和延伸性教辅资料，开发网络课程、虚拟仿真实训平台、工作

过程模拟软件、通用主题素材库以及名师讲义等多种形式的数字化教学资源，建立动态、共享的课程教材信息化资源库，服务于系统培养技术技能人才。

电子信息类教材建设是提高电子信息领域技术技能型人才培养质量的关键环节，是深化职业教育教学改革的有效途径。为了促进现代职业教育体系建设，使教材建设全面对接教学改革、行业需求，更好地服务区域经济和社会发展，我们殷切希望各位职教专家和老师提出建议，并加入到我们的编写队伍中来，共同打造电子信息领域的系列精品教材！

丛书编委会

2014年6月

II

再版前言

“软件测试”领域在当今社会发展得欣欣向荣，软件测试的培训火热，从业人数众多，测试方面的图书也是琳琅满目。四年前，《软件测试技术》第一版问世，深受读者喜爱，多次印刷。出版后，陆续收到读者的反馈，其中针对该书提出不少宝贵的意见。

为了不辜负读者的厚望，我们认真吸取读者的反馈意见，参考更多的资料，历时一年多，对第一版内容做了大量修改，虽然保持了第一版的整体结构，但对一些章节做出调整，完成《软件测试技术》（第二版）的编写。例如，将“黑盒测试”和“白盒测试”的基本内容从原来（第一版）第2、3章调整到现在的第4、5章；将软件测试阶段、软件测试过程与管理的内容添加到第2、3章，从测试项目管理角度来全面介绍测试各阶段和测试过程。第二版还删除了一些和测试内容关系不够紧密的内容，使本书更加专业，留出更大空间来介绍更多的软件测试知识和技术，使之跟上软件技术的发展，更贴近软件测试领域的实际应用，同时，本书在内容上更加完整，涵盖了实际测试工作中所需的各项技能。

本书在第1~4章中做出很大改动，加上前面所述的修改，使本书在内容组织上更加自然、合理，从基本概念到方法，再从方法到技术，逐步推进，使“软件测试”这门课程的学习达到最好的效果。另外，本书在测试工具应用上增加了分量，不仅提高了测试技术水平，而且涉及面更广，从单元测试、GUI功能测试到服务器的性能测试等各个方面，进行了更深入地讨论；在性能测试上也比第一版有更详细的介绍。

本书由库波、杨国勋担任主编，由罗炜、董宁担任副主编，由王路群担任主审，赵丙秀、袁晓曦、李文蕙、胡双参加编写。其中第1章由赵丙秀修订，第4章由李文蕙修订，第5章由袁晓曦修订，第6章由库波修订，第7章由杨国勋修订，第8章由胡双修订。新添加第2章由罗炜编写，第3章由董宁编写。

本书最大的特点是注重实践应用。各种典型的测试技术及方法的介绍均从实际出发，避免抽象的理论论述，在介绍中深入浅出、简洁明了。每章都设有对应测试方法工具使用的介绍，这些实例许多都是根据实际工程案例进行设计的。根据实例，再通过上机不但能够使学生印证许多基本概念，而且能加深理解，从而更好地掌握相应的软件测试方法并能达到熟练应用，通

过把应用与理论知识紧密结合，激发学生学习软件测试的兴趣。

本书既适合作为计算机应用、计算机软件、软件工程、软件测试等学科的教材，也适合从事软件开发和维护的工程技术人员阅读，包括软件测试人员、开发人员、项目经理和产品经理。

由于作者水平有限，本书经过修订仍会存在一些问题，欢迎读者继续提出宝贵意见，不断提高本教材的质量。

编 者

2014 年 6 月

III

目 录

序

再版前言

第1章 软件测试基础知识	1	本章小结	38
1.1 软件的概念	1	实训习题	38
1.2 软件测试的基本概念	2	第3章 软件测试过程与管理	39
1.3 软件测试的必要性	5	3.1 软件测试过程	39
1.4 软件测试的分类	9	3.1.1 测试需求分析	39
本章小结	13	3.1.2 测试计划	40
实训习题	13	3.1.3 测试设计	42
第2章 软件测试阶段	14	3.1.4 测试执行	43
2.1 软件生命周期	14	3.1.5 测试总结	44
2.1.1 软件生命周期的阶段划分	14	3.2 软件测试团队管理	45
2.1.2 常见的生命周期模型	17	3.3 软件测试文档管理	47
2.2 软件测试阶段	22	本章小结	52
2.2.1 单元测试	24	实训习题	52
2.2.2 集成测试	28	第4章 黑盒测试	53
2.2.3 确认测试	30	4.1 黑盒测试方法	53
2.2.4 系统测试	31	4.1.1 等价类划分法	55
2.2.5 回归测试	32	4.1.2 边界值分析法	63
2.3 软件测试模型	34	4.1.3 因果图法	73
2.3.1 V模型	35	4.2 黑盒测试工具	87
2.3.2 W模型	36	4.2.1 黑盒测试工具介绍	87
2.3.3 X模型	37	4.2.2 使用 QTP 进行黑盒测试	90
2.3.4 H模型	37	本章小结	114

实训练题	114	实训练题	184
第 5 章 白盒测试	121	第 7 章 Web 应用测试	185
5.1 白盒测试方法	121	7.1 Web 应用概述	185
5.1.1 代码检查	122	7.1.1 Web 应用	185
5.1.2 覆盖测试	123	7.1.2 Web 应用技术	186
5.1.3 路径测试	127	7.1.3 Web 应用服务器	187
5.2 白盒测试工具 (JUnit)	133	7.2 Web 应用设计测试	191
5.2.1 白盒测试工具介绍	133	7.2.1 Web 应用界面测试	191
5.2.2 使用 JUnit 进行白盒测试	139	7.2.2 Web 应用功能测试	192
本章小结	144	7.2.3 兼容性测试	194
实训练题	144	7.3 Web 应用安全测试	195
第 6 章 性能测试	145	7.3.1 Web 应用安全	195
6.1 软件性能	145	7.3.2 Web 应用安全测试方法	195
6.1.1 软件性能概述	145	7.4 Web 应用压力测试	199
6.1.2 软件性能指标	148	7.4.1 压力测试	199
6.2 软件性能测试方法论	157	7.4.2 压力测试工具 (JMeter) 介绍	201
6.2.1 SEI 负载测试计划过程	157	本章小结	211
6.2.2 RBI 方法	158	实训练题	211
6.2.3 性能下降曲线分析法	159	第 8 章 易用性测试	212
6.2.4 LoadRunner 的性能测试过程	159	8.1 易用性测试概述	212
6.2.5 Segue 提供的性能测试过程	160	8.2 安装测试	215
6.3 性能测试的方法	161	8.3 功能易用性测试	219
6.3.1 性能测试	161	8.4 用户界面测试	221
6.3.2 负载测试	162	8.4.1 界面整体测试	221
6.3.3 压力测试	162	8.4.2 图形用户界面测试用例	226
6.3.4 并发测试	163	8.5 用户文档测试	229
6.4 性能测试工具	163	本章小结	230
6.4.1 性能测试工具介绍	163	实训练题	230
6.4.2 使用 LoadRunner 进行性能测试	165	附录 测试报告模板	231
本章小结	184	参考文献	234

软件测试基础知识



1. 掌握：软件测试的基本概念。
2. 理解：软件测试的分类。
3. 了解：软件测试的必要性。

1.1 软件的概念

软件就是人类在生产实践和改造自然过程中一切无形的知识积累，比如各种发明、专利等。自从有了人类，就开始有了发明和积累，例如我国的造纸术、排版术、火药制造等都可以重复利用的技术，只要能重复在生产实践中使用的无形技术就是软件。

在 20 世纪计算机产生后，软件技术得到重大发展，以至于计算机软件成为了软件科学的代名词，因此本书中所称软件也就是计算机软件。

计算机软件和计算机硬件是相对应的一组概念，计算机硬件特指计算机有形的部分，包含中央处理器、存储器和磁盘，而计算机软件则是一组旨在直接或间接用于计算机以取得一定结果的语句或指令的集合，软件包括了所有与指令代码和数据集合相联系的表示方式。也就是说软件并不只是包括可以在计算机上运行的程序源代码和数据文件，还包括了所有在需求、分析设计等软件开发各阶段中产生相关的文档。简单的说，软件就是程序加文档的集合体。

软件是计算机的灵魂，相当于人类的大脑活动，如果没有软件，计算机就和失去知觉的

植物人一样,虽然能活动,但是没有意识,不能完成最基本的任务。随着社会的发展,计算机软件已经完全进入了我们的生活,比如我们天天在计算机上聊天使用的 QQ、MSN,用手机上网看的股票、新闻,用影碟机看的电影,用 PSP 玩的游戏,用机顶盒看的电视等,可以毫不夸张的说,计算机软件已经完全融入了我们生活的方方面面。

计算机软件主要分为系统软件和应用软件两大类。系统软件为计算机使用提供最基本的功能,负责管理计算机系统中各种独立的硬件,使得它们可以协调工作。系统软件使得计算机使用者和其他软件将计算机当作一个整体而不需要顾及到底层每个硬件是如何工作的。

系统软件又分为操作系统和支撑软件,其中操作系统是最基本的软件,也是一切软件的基础,其余所有软件都必须在这个基础上开发,就像我们要学习各种科学知识前,必须先学好语文一样。操作系统是管理计算机硬件与软件资源的程序,同时也是计算机系统的内核与基石。操作系统身负诸如管理与配置内存、决定系统资源供需的优先次序、控制输入与输出设备、操作网络与管理文件系统等基本事务。操作系统也提供一个让使用者与系统交互的操作接口。操作系统作为最基础的软件,无处不在。智能卡中有 TimeCOS、手机中有 Windows CE、Symbian、Android、IOS 等,路由器中有 μ CLinux、vxWorks 等,计算机中有 BSD、DOS、Linux、Mac OS、OS/2、QNX、UNIX、Windows 等。支撑软件是支撑各种软件开发与维护的软件,又称为软件开发环境 (IDE)。它主要包括环境数据库、各种接口软件和工具组。著名的软件开发环境有 Borland 公司的 BDS, Sun 公司的 NetBean 和 Java, IBM 公司的 WebSphere, 微软公司的 Studio.NET 等。它们都包括一系列基本的工具 (如编辑器、编译器、数据库管理、存储器格式化、文件系统管理、用户身份验证、驱动管理、网络连接等方面的工具)。

但是系统软件并不针对某一特定应用领域。而应用软件则相反,不同的应用软件根据用户和所服务的领域提供不同的功能。

应用软件是为了某种特定的用途而被开发的软件。它可以是一个特定的程序,比如一个图像浏览器,也可以是一组功能联系紧密,可以互相协作的程序的集合,比如微软的 Office 软件。应用软件是计算机软件产品的主体,我们谈论的软件 90% 以上是指应用软件,像先前提到的日常生活中常见的 QQ、MSN、手机股票、游戏软件等都属于应用软件。

应用软件主要分为办公软件、行业软件、翻译软件、安全工具、财务软件、读书软件、刻录软件、图形图像软件、音频视频播放软件、媒体制作软件、网络服务软件、网络应用软件、输入法和压缩解压缩软件等。

1.2 软件测试的基本概念

有人觉得测试是可有可无的一个过程,也有人认为测试是一件很遥远的事情,是那么虚无缥缈飘忽不定。其实,当我们静下心来仔细想一想,就会发现测试就在我们身边,测试无处不在,可以说没有测试,就不会有社会进步和人类的文明。在生活方面,做饭需要先品尝味道,买衣服需要试穿,打针需要做皮试,喝茶需要尝试,输血要检查血型,牛奶出厂要检查有没有

三聚氰氨,买鸡蛋要看是否有损坏等;在工作方面,公司要对员工技术能力进行考核,采购前对办公设备性能进行评估,员工对领导管理要进行评测,产品出厂要抽查等;在科研方面,流感疫苗需要临床试验,航天飞机需要模拟试验,克隆技术需要验证,武器装备需要测试等。总之,每出现一个新事物,每做一件新事情,都需要经过一个重要的环节,那就是测试。如果没有测试,或者测试不全面,我们的行为就会成为冒险,也是一种赌博,并且为此付出沉痛的代价。1986年,美国的“挑战者”号航天飞机在升空不到一分钟就化为灰烬,七名宇航员全部遇难,其原因是一个绝缘垫出现一个小小的裂痕,就是这个小小的裂痕,导致整个航天飞机发生爆炸,让上万人的努力付之东流,损失高达几百亿美元,比黄金还珍贵的宇航员葬身太空;无独有偶,2003年美国的“哥伦比亚”号航天飞机在返回地球途中,在大气层发生了爆炸,原因是飞机机身被高速绝热泡沫碎片擦出一个印记,在高温下导致航天飞机发生断裂。可见,即使高科技发展到今天,一个小小的错误都会引起灾难性的后果。

对于产品的测试就需要更加严格,因为这是关系到千家万户利益的大事。例如2008年因为不法奶农在牛奶中添加三聚氰氨,三鹿公司又没有进行认真的质量测试,造成全国12892个婴儿服用三鹿奶粉后患肾结石不得不住院治疗,最后直接导致了三鹿奶粉公司的破产。从上述事实可以看出一个产品在正式使用前必须经过严格的测试,只有通过严格测试的产品才能够把错误发生率降到最低,保证客户利益不受损害,保证使用者的使用安全,同时也保障的公司的利益。

计算机软件是一种无形的产品,每个产品上市之前必须要经过测试。我们已经明确了软件的定义,又知道了测试的重要,接下来我们看看什么是软件测试。软件测试是测试的一种,顾名思义就是对软件进行测试。软件测试是由于软件缺陷的存在而产生的。我们将所有软件问题统称作软件缺陷,不管它们的规模 and 危害有多大,由于它们都会产生使用障碍,故都称为软件缺陷。

软件测试就是在软件投入运行前,对软件需求分析、设计规格说明和编码实现的最终审查,它是软件质量保证的关键步骤。

1. 软件测试的定义

(1) 软件测试:通常对软件测试的定义有两种描述:

- 软件测试是为了发现错误而执行程序的过程。
- 软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计的一批测试用例,并利用这些测试用例运行程序以及发现错误的过程,即执行测试步骤。

其实这两种定义是一致的,前面一种定义,强调了软件测试的目的,后面一种定义强调了软件测试的方法和过程。通过这两个定义,我们可以知道,软件测试就是通过设计测试方法和用例,对软件进行流程抽样得到软件正确性、可靠性的过程。

软件测试在软件生命周期中横跨两个阶段:

- 第一个阶段:单元测试阶段,即在每个模块编写出后所做的必要测试。
- 第二个阶段:综合测试阶段,即在完成单元测试后进行的测试,如集成测试、系统测试、验收测试。

(2) 测试用例: 所谓测试用例是为特定的目的而设计的一组测试输入、执行条件和预期的结果; 测试用例是执行测试的最小实体。一个好的测试用例在于可以发现还未曾发现的错误; 一次成功的测试则是发现了错误的测试。

(3) 测试步骤: 详细规定了如何设置、执行、评估特定的测试用例。

2. 软件测试的对象

软件测试不等于程序测试。软件测试贯穿于软件定义和开发的整个过程。因此, 软件开发过程中所产生的需求规格说明、概要设计规格说明、详细设计规格说明以及源程序都是软件测试的对象。

3. 软件测试的目的

很多人认为开发一个软件是复杂的, 需要花费大量的人力和物力, 而测试一个软件相对比较容易, 通过测试能够找出所有软件错误。有很多测试新手是满怀下面这些信念进入测试领域的: 他们能够充分地测试每一个程序, 通过他们的完全测试能够确保程序正常运行, 测试人员的工作任务就是通过进行完全测试, 来保证程序的正确性。然而, 事实上, 无论程序多么简单, 我们都不可能对程序进行完全测试, 原因有三点:

- (1) 可能的输入范围太大, 根本无法穷举测试;
- (2) 程序中可能的运行路径太多, 根本无法穷尽测试;
- (3) 用户界面问题太复杂, 不可能进行完全测试。

既然不可能进行完全测试, 那么我们就无法验证一个程序的正确性。那测试的目的到底是什么呢?

测试的目的是发现错误。测试是程序的执行过程, 目的在于发现错误, 并且尽可能地发现更多的错误, 发现的错误越严重越好。软件测试只能证明程序中有错误存在, 而不能证明程序的正确性。软件测试是从软件中包含有缺陷和故障这个假设出发去测试程序, 以期从中发现尽可能多的软件故障。测试以发现错误为目的, 是为了发现错误而执行程序的过程。好的测试用例比差的测试用例更有可能找出错误, 或是更有可能找出严重的错误。

注意: 测试无法说明错误不存在, 只能说明软件错误已出现。

4. 软件测试的原则

根据以上软件测试的目的, 软件测试的原则是:

- (1) 尽早地和及时地测试, 错误发现得越晚, 修复的代价越高。
- (2) 测试用例应当由测试数据和与之对应的预期结果两部分组成。
- (3) 在程序提交测试后, 应当由专门的测试人员进行测试, 避免由程序设计者自行检查程序。
- (4) 测试用例应包括合理的输入条件和不合理的输入条件。
- (5) 严格执行测试计划, 排除测试的随意性。
- (6) 充分注意测试当中的群体现象。
- (7) 应对每一个测试结果做全面的检查。

(8) 保存测试计划、测试用例、出错统计和最终分析报告,为维护工作提供充分的资料。

5. 测试停止的依据(标准)

受到经济、工期或其他方面的条件制约,测试最终是要停止的。下面是常用的 5 类停止测试的标准:

(1) 第一类标准:测试超过了预定时间,则停止测试。这类标准不能用来衡量测试质量。

(2) 第二类标准:执行了所有的测试用例,但并没有发现故障,则停止测试。这类标准对测试也没有好的指导作用,相反却鼓励测试人员不去编写更好的、能暴露出更多故障的测试用例。

(3) 第三类标准:使用特定的测试用例设计方案作为判断测试停止的基础。这类标准仍然是一个主观的衡量尺度,无法保证测试人员准确、严格地使用某种测试方法。这类测试标准只是给出了测试用例设计的方法,并非确定的目标,这类标准只对某些测试阶段适用。

(4) 第四类标准:正面指出停止测试的具体要求,即停止测试的标准可定义为查出某一预订数目的故障。比如规定发现并修改了 60 个故障就可以停止测试,对系统测试的标准是,发现并修改若干个故障或者至少系统要运行一定时间,如 3 个月等。

(5) 第五类标准:根据单位时间内查出故障的数量决定是否停止测试。这类标准看似容易,但是在实际操作中要用到很多直觉和判断。通常使用某个图表表示某个测试阶段中单位时间检查出的故障数量,通过分析表,确定应继续测试还是停止测试。

6. 软件测试和修复

软件测试和修复是不同意义的行为过程,最能体现修复行为的是调试和修正。经过测试发现错误后,往往不能直觉地从测试结果中找到错误的根源,这就需要充分利用测试结果和测试过程中提供的信息进行全面分析,通过调试发现错误,并修正这些发现的错误。

1.3 软件测试的必要性

现代社会已经离不开软件,软件产品的质量要求也就与日俱增,一个小的疏忽,轻者让软件公司信誉扫地,重则导致公司破产,甚至会损坏国家利益,造成不可挽回的损失,下面就列举几个软件产品由于存在缺陷而造成失败或者重大损失的案例,让我们通过这些生动的案例了解软件测试的必要性。

1. Ashton Tate 公司的 dBase IV

Ashton Tate 公司在 1980 年以 dBase 数据库管理系统起家。dBase 不久就占据了市场的主导。Ashton Tate 成为软件业三大企业之一。1987 年,Ashton Tate 销售额为 2.15 亿美元,仅仅略落后 Lotus 公司的 2.83 亿美元,微软的 2.6 亿美元。dBase 产品占据 Ashton Tate 公司的 65%。

当竞争者开始提供更高效易用的数据库产品时,Ashton Tate 开发了一个增强版称为 dBase IV。1988 年 2 月,Ashton Tate 公司宣称 dBase IV 将在 5 月发布。到了 5 月,又宣称延迟 2 个月,到 8 月,又延迟 2 个月。到了 9 月,当它最终被用户放弃时,Ashton Tate 公司宣称新的 dBase

产品将在 10 月底发布。

不幸的是, dBase IV 有太多的缺陷, 在投入使用 2 个月后, Ashton Tate 不得不收回它。在 1989 年 9 月, CMM 创始人汉弗莱碰到 Ashton Tate 的 CEO 时, 工程师们还在测试装配 dBase IV。但是 Ashton Tate 的 CEO 并没有按照汉弗莱的建议制作产品质量问题的数据库, 并利用数据库来确定缺陷的位置进行重新编写, 采取合适的流程来管理缺陷, 而是采用了边测试、边修复的应对措施, 结果他们没有在 2 个月内发布, 在以后的一年里, 都在不断的测试和修复问题。到了 1991 年 2 月, dBase IV 还处在第 2 个测试版。Ashton Tate 公司的 CEO 也被更换。据报道, 因为质量问题, Ashton Tate 公司每季度亏损 560 万美元, 不久就被 Borland 公司收购。Ashton Tate, 这个曾经软件业第三大的公司, 再也不存在了。

2. Borland 公司的 Borland C++ 4.0

在 1994 年前 Borland 依靠 Borland C++ 3.0 统治了 C++ 市场 3 年之久, 但是当微软公司推出 Visual C++ 1.0 在 C/C++ 开发工具市场获得空前的成功之后, Borland 才开始研发 Borland C/C++ 4.0。事实上, 当时外围公用程序, Shareware 等都是使用 Borland C/C++ 3.1 开发的。因此, 就算 Borland 不着急, 好好地开发下一代的 C/C++ 开发工具, 即使 Microsoft Visual C++ 能够掠夺一些市场占有率, 但是如果下一代的 Borland C/C++ 能够像 Borland C/C++ 3.0 一样立刻拉开和 Visual C/C++ 的距离, 那么 Borland 在 C/C++ 市场仍将拥有王者的地位。

可惜的是, Borland 公司当时的总裁 Philippe Kahn 却急于推出 Borland C/C++ 4.0 来挽回面子, 在短短的一年的时间内就要全新升级集成开发环境, 把 OWL 完全重写, 大幅修改最佳化编译器, 整合当时棘手的 VBX。这些都是技术难点, 每个难点都需要不少的时间, 但是在 Philippe Kahn 的催促下, Borland 公司的工程师们硬着头皮将新版本拿出来了。

可惜由于 Borland 太急于推出 4.0, 因此并没有在最后阶段修正许多的 bug, 又没有经过最后系统微调的工作, 同时又过于大胆地加入太多先进的技术, 造成了整个产品大量存在着以下软件缺陷:

- (1) 集成开发环境方面: bug 太多, 容易当掉而且反应速度缓慢。
- (2) 编译器方面: 最佳化玩得过火, 产生错误的编译程序代码。
- (3) OWL 方面: 采用全新的多重继承架构, 虽然是正确的做法, 却和 Borland C/C++ 3.1 中的 OWL 不兼容, 造成许多程序员无法升级 C/C++ 项目。
- (4) VBX 方面: 大胆采用在 16/32 位都能使用的 VBX 技术, 造成一些 VBX 无法顺利地 Borland C/C++ 4.0 中使用。

由于 Borland 推出了根本不堪使用的产品 Borland C/C++ 4.0, 立刻造成了严重的后果。首先是 Borland C/C++ 的市场份额大量而且快速地流失, 使得 Visual C/C++ 快速地成长。其次是当初 Borland C/C++ 3.1 在公用程序市场打下的江山也拱手让人, 原本许多使用 Borland C/C++ 3.0/3.1 撰写驱动程序的硬件厂商也开始转换到 Visual C/C++。而更严重的是, 由于 4.0 的品质以及 OLE 的关系, 应用程序市场也开始大量地转为使用 Visual C/C++ 来编写应用程序。从此 Borland 在开发工具市场退出主流。

3. 迪斯尼公司的狮子王游戏

1994 年, 美国迪斯尼公司狮子王游戏光盘被称为成百上千万孩子的“圣诞杀手”, 销售异常火爆, 但是产品售出不久, 该公司的客户支持和售后电话就一直不断, 愤怒的家长 and 玩不成游戏的孩子们大量投诉该软件的缺陷, 一段时间各种报纸和电视媒体都大量报道了这个游戏的各种问题, 最主要的问题是没有进行兼容性测试, 原来该游戏依赖于微软当时最新的 WinG 图形引擎, 用户必须修改计算机的显卡驱动设置方能正常使用。但 1994 年时惠普推出的 Presario 计算机所使用的光驱与 WinG 并不兼容, 圣诞节一大早当家长们为孩子们安装狮子王游戏光盘时, 所有的计算机都无一例外出现死机。

这次软件故障使得迪斯尼公司的声誉大受影响, 并为改正这个软件的缺陷和故障付出了沉重的代价。

4. 千年虫问题

千年虫问题产生的原因是由于在计算机软、硬件以及数字化程序控制芯片的各种设备和业务处理系统中, 只使用了两位十进制数来表示年份, 因此, 当日期从 1999 年 12 月 31 日进入 2000 年 1 月 1 日后, 系统将无法正常识别由“00”表示的 2000 年(计算机可能将这个年份识别为 1900 年)这一具体年份, 从而带来进行跨世纪的年份、日期处理时的计算错误, 引发各种各样的计算机业务处理系统和控制系统的功能紊乱。“千年虫问题”又称为“2000 年问题”或“千年病毒”, 或简称为“Y2K”。

2000 年零点钟声在一片欢呼雀跃中敲响, 人类欢庆久盼新千年的到来。

欣喜之余, 人们也终于可以长舒一口气, 原来被形容得如同洪水猛兽般的 Y2K 千年虫并没有大范围侵入我们的生活, 电灯仍然发出光芒, 银行自动提款机像平日那样“吐出”现款, 电话照常运作, 水也继续从水龙头流出来。在花费了 6000 亿美元对计算机系统进行调整, 在系统管理员和程序设计员的严密监视下, 人类跨入了新千年。

钟声逐渐平息, 激情日益淡去, 人类又回到理性的思维之中。6000 亿美元的代价是何等的惨痛, 其损失已不亚于一场大规模的战争, 而原因仅仅是因为当年不经意之中的两位数字。在计算机诞生初期, 程序设计员为了节省计算机内部珍贵的存储空间, 就把年份的时间表示设定为两位数字, 而不是四位数字。当时间转到跨世纪的这一天, 计算机中就会出现时间上 2000 年小于 1999 年的荒谬现象, 网络系统的运算逻辑发生混乱, 最终导致人类生活的瘫痪。他们没有预计到计算机的发展速度会有如此之快, 普及程度会有如此之广。

Y2K 给我们的教训是深刻的, 新千年的人类应该更理性、更慎重地对待科学技术的发展, 避免再犯这样看似容易的“小”问题。

5. 暴风软件召回

2009 年中国互联网出现了一次大地震, 5 月 19 日, 江苏、安徽、广西、海南、甘肃、浙江等六省用户申告访问网站速度变慢或无法访问, 原因是 2 个游戏私服恶性竞争, 攻击导致 DNSPod 公司所属的 6 台 DNS 域名解析服务器瘫痪, 直接造成包括暴风影音在内的多家网络服务商的域名解析系统瘫痪, 由此引发网络拥塞, 造成大量用户不能正常上网。