

21 世纪 高 职 高 专 规 划 教 材

案例式
教材

软 件 专 业 系 列



软件测试

曹 薇 主 编

清华大学出版社



21世纪高职高专规划教材

软件专业系列

软件测试

曹薇 主编

本书以《软件测试》课程为蓝本，根据教育部颁布的《计算机专业人才培养方案》和《计算机专业教学大纲》的要求，结合编者多年的教学经验和科研成果编写而成。全书共分8章，主要内容包括：软件测试概述、软件测试策略、软件测试方法、软件测试工具、软件测试案例、软件测试管理、软件测试文档、软件测试案例。本书可作为高等院校计算机专业及相关专业的教材，也可供从事软件测试工作的工程技术人员参考。

6. 殷人昆. 软件工程复习与考试指导. 北京: 高等教育出版社, 2005.
7. 朱少民等. 软件测试方法. 北京: 清华大学出版社, 2005.
8. 贺平. 软件测试教程. 北京: 电子工业出版社, 2005.
9. 王健等. 软件测试员培训教材. 北京: 电子工业出版社, 2003.
10. Ron Patton. 软件测试. 周子华译. 北京: 清华大学出版社, 2005.
11. 张雪峰. 面向对象软件测试研究. 计算机工程, 2004, 30(12): 1-5.
12. 洪红信等. 面向对象的软件测试技术探讨. 长沙理工大学学报(自然科学版), 2004, 1(12): 68-71.
13. 杨善红, 李静雯. 测试驱动开发研究. 计算机工程, 2005, 31(10): 1-5.
14. 中华人民共和国国家标准. 计算机软件测试规范 GB/T 9386-1988. 北京: 中国标准出版社, 1989.
15. Rational Software Corporation. Rational Software Corporation. 北京: 清华大学出版社, 2002.
16. <http://blog.csdn.net/cesdbu/archive/2005/02/22/2727272.aspx>
17. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>
18. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>
19. <http://blog.csdn.net/chenshaoying/archive/2005/02/22/2727272.aspx>
20. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>
21. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>
22. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>
23. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>
24. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>
25. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>
26. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>
27. <http://www.cnblogs.com/yangshang/archive/2005/02/22/2727272.aspx>

清华大学出版社
北京
ISBN 7-302-16966-9
定价: 24.00元

本书较系统地介绍了软件测试的概念、原理、方法和技术。主要内容包括:软件测试的目标和原则,软件测试的认识误区,软件测试的定义和分类,测试过程管理,测试驱动开发的概念及策略,软件测试文档的分类和编制规范,黑盒和白盒测试用例的设计策略,单元测试的概念及策略,使用 CppUnit 进行单元测试,集成测试的概念及策略,系统测试的概念、分类及策略,面向对象软件测试的概念及策略,自动化测试的概念及主流测试工具,使用 WinRunner 进行功能测试,软件质量和质量保证;并在书末给出了软件测试案例。本书含有较多的例题、案例和习题,便于教学和自学。

本书强调理论与实践相结合,内容简明易懂、逻辑性强。可作为高职高专院校计算机专业的教材或参考书,也可供其他各类人员参考使用。

编 者 曹 薇

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件测试/曹薇主编. —北京:清华大学出版社,2008.4

21世纪高职高专规划教材. 软件专业系列

ISBN 978-7-302-16966-6

I. 软… II. 曹… III. 软件—测试—高等学校:技术学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2008)第 013181 号

责任编辑:束传政

责任校对:李梅

责任印制:何芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京国马印刷厂

装 订 者:三河市兴旺装订有限公司

经 销:全国新华书店

开 本:185×260

印 张:17.25

字 数:393千字

版 次:2008年4月第1版

印 次:2008年4月第1次印刷

印 数:1~4000

定 价:24.00元

清华大学出版社

前 言

软件测试

查 录

只 SI 年 2005

随着信息技术的飞速发展,软件在社会生活的方方面面发挥着日益重要的作用。软件的复杂性不断增大,开发周期有限,这些都导致了软件质量时常得不到保证。如何提高软件开发的质量呢?除了按照软件工程的过程和方法论去优化需求分析、设计、编程外,通过软件测试发现软件中的缺陷并进行修复显然是提升软件质量的重要途径。

软件测试作为软件开发中的一个重要环节,已逐渐形成一门新的学科和产业。2003年10月,国家人事部和信息产业部联合发文,将“软件评测师”资格考试纳入计算机技术与软件专业技术资格(水平)考试,就是一个显著的标志。

我国的软件测试研究和应用起步较晚,在软件测试理论研究、软件测试工具及框架研发、软件测试过程管理、软件质量保证等方面还落后于一些发达国家。尤其是在软件测试工具方面,国外的产品几乎占据了垄断地位。同时,软件测试的新理论与新技术日新月异。这些对我国的软件从业人员和测试从业人员都是严峻的考验。所以,作为软件测试产业从业人员的后备力量——广大的高职高专院校计算机专业学生(尤其是软件专业学生),理应系统地学习软件测试的概念、原理、方法和技术,为将来投身我国的测试产业做好准备,力争为我国测试产业的发展做出应有的贡献。

本书结合高职高专院校的教学特点,较系统地介绍了软件测试,力求逻辑严谨、简明易懂、实践性强。书中的例题、案例和习题较多,便于教学和自学。本书还吸纳了软件测试中若干新理论和新技术,以反映软件测试的最新进展。

本书共分11章。第1章主要介绍了软件测试的目标,软件测试的认识误区,软件测试的发展史、现状及发展趋势;第2章介绍了软件测试的定义、对象及分类,V&V,软件测试过程模型和过程管理理念,测试驱动开发的概念和策略,软件测试的原则,软件测试文档;第3~4章介绍了黑盒和白盒测试用例的设计方法和策略;第5章介绍了单元测试的概念、步骤、环境和内容,单元测试用例的设计思路,用CppUnit进行单元测试;第6~7章介绍了集成测试、确认测试、系统测试和验收测试的概念和策略;第8章介绍了目前研究和应用的热点——面向对象软件测试的概念、模型和策略;第9章介绍了软件测试自动化的必要性和优点,自动化测试的认识误区,自动化测试的实施流程,自动化测试的原理和方法,主流测试工具,IBM Rational和HP Mercury公司的自动化测试解决方案,使用WinRunner进行功能测试;第10章介绍了软件质量和质量模型,软件度量,CMM及CMMI,PSP及TSP,SQA;第11章给出了几个具体的测试案例。书末附有常用测试术语

表和测试站点推荐。

本书由曹薇任主编,张乃洲任副主编。张乃洲编写了第2章和第5章,其他章由曹薇编写。由于水平有限、时间仓促,书中难免存在不足之处,恳请专家和读者批评指正。

曹薇

编者

2007年12月

软件测试的重要性日益受到重视,软件测试已成为软件开发过程中不可或缺的一部分。本书旨在为软件测试人员提供系统的知识,帮助他们掌握软件测试的基本原理和方法。本书共分11章,主要内容包括:软件测试的基本概念、软件测试的分类、软件测试的模型、软件测试的计划和设计、软件测试的组织和实施、软件测试的自动化、软件测试的度量、软件测试的工具、软件测试的改进等。

本书可作为高等院校计算机专业及相关专业的教材,也可作为从事软件测试工作的工程技术人员参考。本书由曹薇任主编,张乃洲任副主编。张乃洲编写了第2章和第5章,其他章由曹薇编写。由于水平有限、时间仓促,书中难免存在不足之处,恳请专家和读者批评指正。

目 录

| | |
|-------------------|-----------|
| 软件测试 | 1 |
| 1.1 软件测试的意义 | 1 |
| 1.1.1 软件缺陷的典型例子 | 1 |
| 1.1.2 软件缺陷的产生原因 | 3 |
| 1.1.3 软件测试的目标 | 4 |
| 1.2 软件测试中的认识误区 | 4 |
| 1.3 软件测试的发展史及现状 | 6 |
| 1.3.1 软件测试的发展史 | 6 |
| 1.3.2 软件测试的国内外现状 | 7 |
| 1.4 软件测试的发展趋势 | 8 |
| 1.5 小结 | 9 |
| 习题 | 9 |
| 第2章 软件测试基础 | 10 |
| 2.1 软件测试概念 | 10 |
| 2.1.1 软件测试的定义和对象 | 10 |
| 2.1.2 验证与确认 | 11 |
| 2.2 软件测试分类 | 12 |
| 2.2.1 按开发阶段分类 | 12 |
| 2.2.2 按测试实施组织分类 | 13 |
| 2.2.3 按测试策略分类 | 13 |
| 2.2.4 按测试执行方式分类 | 14 |
| 2.2.5 其他测试方法和技术 | 14 |
| 2.3 软件测试过程 | 16 |
| 2.3.1 软件测试过程模型 | 17 |
| 2.3.2 测试过程的实施策略 | 20 |
| 2.4 测试驱动开发 | 21 |
| 2.4.1 测试驱动开发的概念 | 21 |

| | | |
|------------|-----------------------------------|-----------|
| 2.4.2 | 测试驱动开发的优点 | 22 |
| 2.4.3 | 测试驱动开发的原则 | 23 |
| 2.4.4 | 关于测试驱动开发的一些讨论 | 24 |
| 2.5 | 软件测试的原则 | 25 |
| 2.6 | 软件测试文档 | 26 |
| 2.6.1 | 软件测试文档概述 | 26 |
| 2.6.2 | 《计算机软件测试文件编制规范》(GB/T 9386—1988)简介 | 26 |
| 2.6.3 | 规范 GB/T 9386—1988 内容要求 | 29 |
| 2.7 | 小结 | 34 |
| | 习题 | 34 |
| 第3章 | 黑盒测试方法 | 36 |
| 3.1 | 测试用例 | 36 |
| 3.2 | 黑盒测试方法概述 | 37 |
| 3.2.1 | 黑盒测试的概念和对象 | 37 |
| 3.2.2 | 黑盒测试的优点和局限性 | 38 |
| 3.3 | 典型的黑盒测试方法 | 39 |
| 3.3.1 | 等价类划分法 | 39 |
| 3.3.2 | 边界值分析法 | 42 |
| 3.3.3 | 错误推测法 | 42 |
| 3.3.4 | 因果图法 | 45 |
| 3.3.5 | 判定表法 | 50 |
| 3.3.6 | 正交试验法 | 50 |
| 3.3.7 | 场景法 | 55 |
| 3.3.8 | 功能图法 | 60 |
| 3.4 | 黑盒测试方法的综合使用策略 | 61 |
| 3.5 | 小结 | 61 |
| | 习题 | 62 |
| 第4章 | 白盒测试方法 | 64 |
| 4.1 | 白盒测试方法概述 | 64 |
| 4.1.1 | 白盒测试的概念 | 64 |
| 4.1.2 | 白盒测试的优点和局限性 | 65 |
| 4.1.3 | 白盒测试方法的分类 | 65 |
| 4.2 | 典型的白盒测试方法 | 66 |
| 4.2.1 | 代码检查法 | 66 |
| 4.2.2 | 静态结构分析法 | 67 |
| 4.2.3 | 代码质量度量法 | 67 |

| | | |
|--------------|-----------------------|------------|
| 4.2.4 | 逻辑覆盖法 | 67 |
| 4.2.5 | 基本路径测试法 | 70 |
| 4.2.6 | 控制结构测试 | 75 |
| 4.2.7 | 程序插桩 | 77 |
| 4.3 | 白盒测试方法的综合使用策略 | 78 |
| 4.4 | 对黑盒、白盒测试方法的总结 | 79 |
| 4.5 | 小结 | 79 |
| 习题 | | 80 |
| 第 5 章 | 单元测试 | 81 |
| 5.1 | 单元测试概述 | 81 |
| 5.2 | 单元测试的步骤 | 83 |
| 5.3 | 单元测试环境 | 83 |
| 5.4 | 单元测试用例的设计 | 86 |
| 5.4.1 | 单元测试的内容 | 86 |
| 5.4.2 | 单元测试用例的设计思路 | 87 |
| 5.5 | 单元测试工具 CppUnit 简介 | 88 |
| 5.5.1 | CppUnit 简介 | 88 |
| 5.5.2 | CppUnit 单元测试实例 1 | 94 |
| 5.5.3 | CppUnit 单元测试实例 2 | 96 |
| 5.6 | 小结 | 102 |
| 习题 | | 103 |
| 第 6 章 | 集成测试 | 104 |
| 6.1 | 集成测试的概念 | 104 |
| 6.1.1 | 集成测试的含义 | 104 |
| 6.1.2 | 接口的分类 | 105 |
| 6.1.3 | 集成测试的测试方法 | 106 |
| 6.2 | 集成测试的实施 | 106 |
| 6.3 | 集成测试的策略 | 107 |
| 6.3.1 | 典型的集成测试策略 | 107 |
| 6.3.2 | 集成测试策略的选取 | 112 |
| 6.3.3 | 模块和接口的确定 | 113 |
| 6.4 | 小结 | 113 |
| 习题 | | 114 |
| 第 7 章 | 确认测试、系统测试和验收测试 | 115 |
| 7.1 | 确认测试的概念和活动 | 115 |

| | | |
|-----|---------------------------|-----|
| 70 | 7.2 系统测试的概念和类型 | 116 |
| 107 | 7.2.1 系统测试的概念 | 116 |
| 117 | 7.2.2 系统测试的主要类型 | 117 |
| 124 | 7.2.3 系统测试与集成测试的区别 | 124 |
| 125 | 7.3 性能测试策略 | 125 |
| 125 | 7.3.1 全面性能测试模型 | 125 |
| 128 | 7.3.2 性能测试策略 | 128 |
| 130 | 7.3.3 全面性能测试模型的使用 | 130 |
| 131 | 7.4 基于 Web 的系统测试 | 131 |
| 131 | 7.4.1 功能测试 | 131 |
| 133 | 7.4.2 用户界面测试 | 133 |
| 134 | 7.4.3 兼容性测试和配置测试 | 134 |
| 135 | 7.4.4 安全测试 | 135 |
| 136 | 7.4.5 接口测试 | 136 |
| 137 | 7.5 回归测试 | 137 |
| 137 | 7.5.1 回归测试的概念 | 137 |
| 138 | 7.5.2 回归测试策略 | 138 |
| 139 | 7.6 系统测试步骤 | 139 |
| 140 | 7.7 验收测试 | 140 |
| 140 | 7.7.1 验收测试的概念 | 140 |
| 140 | 7.7.2 验收测试的策略 | 140 |
| 141 | 7.8 小结 | 141 |
| 141 | 习题 | 141 |
| 143 | 第 8 章 面向对象软件的测试 | 143 |
| 143 | 8.1 面向对象开发方法概述 | 143 |
| 143 | 8.1.1 面向过程开发方法的不足 | 143 |
| 144 | 8.1.2 面向对象的基本概念 | 144 |
| 146 | 8.1.3 面向对象开发方法的优点 | 146 |
| 147 | 8.1.4 统一建模语言 UML 简介 | 147 |
| 148 | 8.2 面向对象软件测试概述 | 148 |
| 148 | 8.2.1 面向对象软件测试的重要性 | 148 |
| 148 | 8.2.2 面向对象软件的特点及其对软件测试的影响 | 148 |
| 150 | 8.2.3 面向对象软件的测试模型 | 150 |
| 159 | 8.3 小结 | 159 |
| 160 | 习题 | 160 |

| | |
|--------------------------------------|-----|
| 第 9 章 软件测试自动化 | 161 |
| 9.1 自动化测试概述 | 161 |
| 9.2 自动化测试的引入和实施 | 162 |
| 9.2.1 对自动化测试的认识误区 | 162 |
| 9.2.2 自动化测试的实施流程 | 164 |
| 9.3 自动化测试的原理和方法 | 165 |
| 9.3.1 脚本技术 | 165 |
| 9.3.2 脚本预处理 | 167 |
| 9.3.3 自动比较技术 | 167 |
| 9.4 对产品可测试性的考虑 | 167 |
| 9.5 测试工具概述 | 168 |
| 9.5.1 主流测试工具 | 168 |
| 9.5.2 IBM Rational 软件自动化测试解决方案 | 170 |
| 9.5.3 HP Mercury 软件自动化测试解决方案 | 175 |
| 9.6 使用 WinRunner 进行功能测试 | 180 |
| 9.6.1 WinRunner 简介 | 180 |
| 9.6.2 WinRunner 测试流程 | 184 |
| 9.6.3 GUI Map | 187 |
| 9.6.4 创建测试 | 192 |
| 9.6.5 GUI 检查点 | 196 |
| 9.6.6 文本检查点 | 205 |
| 9.6.7 图像检查点 | 208 |
| 9.6.8 使用函数生成器 | 212 |
| 9.6.9 数据驱动测试 | 215 |
| 9.7 小结 | 215 |
| 习题 | 215 |
| 第 10 章 软件质量和质量保证 | 216 |
| 10.1 软件质量 | 216 |
| 10.1.1 软件质量的含义 | 216 |
| 10.1.2 软件质量模型 | 217 |
| 10.2 软件度量 | 218 |
| 10.2.1 软件度量概述 | 218 |
| 10.2.2 软件度量的目标 | 220 |
| 10.2.3 软件度量的内容 | 221 |
| 10.3 软件能力成熟度模型 | 223 |
| 10.3.1 软件能力成熟度模型概述 | 223 |

| | | |
|---------------|------------------------|-----|
| 10.3.2 | 软件能力成熟度模型的建立和评估 | 227 |
| 10.3.3 | 个体软件过程 PSP 和群组软件过程 TSP | 228 |
| 10.3.4 | 能力成熟度模型集成 CMMI | 229 |
| 10.3.5 | SPCA 评估体系 | 232 |
| 10.4 | 其他软件管理体系 | 233 |
| 10.5 | 软件质量保证 | 233 |
| 10.5.1 | 软件质量保证概述 | 233 |
| 10.5.2 | 软件质量保证的工作内容 | 234 |
| 10.5.3 | 软件质量保证的实施 | 234 |
| 10.5.4 | 软件质量保证与测试的区别 | 235 |
| 10.6 | 小结 | 236 |
| | 习题 | 236 |
| 第 11 章 | 软件测试案例 | 237 |
| 11.1 | 企业自动化测试方案选型案例 | 237 |
| 11.1.1 | 公司背景介绍 | 237 |
| 11.1.2 | 公司应用系统现状 | 238 |
| 11.1.3 | 公司软件测试现状 | 238 |
| 11.1.4 | 可供选择的方案 | 238 |
| 11.1.5 | 最终采用的测试自动化方案 | 239 |
| 11.2 | SQL Server 2000 压力测试 | 240 |
| 11.2.1 | 测试项目概述 | 240 |
| 11.2.2 | 测试计划 | 240 |
| 11.2.3 | 测试准备 | 240 |
| 11.2.4 | 测试过程 | 241 |
| 11.2.5 | 测试结果 | 243 |
| 11.3 | 某金融业务系统的性能测试 | 243 |
| 11.3.1 | 测试项目概述 | 243 |
| 11.3.2 | 被测系统的性能要求 | 244 |
| 11.3.3 | 性能测试过程 | 245 |
| 11.4 | 小结 | 249 |
| | 附录 A 常见的软件测试术语 | 250 |
| | 附录 B 优秀的测试站点资源 | 260 |
| | 参考文献 | 263 |

第1章

软件测试概述

本章要点：

- 软件缺陷及其产生原因。
- 软件测试的目标。
- 软件测试中的认识误区。
- 软件测试的发展史。
- 软件测试的国内外现状。
- 软件测试的发展趋势。

软件是与计算机系统操作相关的程序、数据和文档，是人类社会高度发展的产物，是人类智慧的结晶。从20世纪50年代初，软件技术不断取得进展，使软件在内涵、规模、开发方法、应用领域等方面都发生着日新月异的变化。软件越来越多地影响和改变人类生活的各方面。然而，软件构成及开发的日益复杂、软件应用领域的日益拓宽也使得人们常常受到有缺陷的软件的影响，软件缺陷给人们带来了许多物质上和精神上的损失。软件质量不断受到人们的重视，为了发现软件中的缺陷，保证软件质量，软件测试应运而生。本章主要介绍软件测试的目标、发展史、现状及发展趋势，以及人们对软件测试的认识误区，为后续章节的学习打下基础。

1.1 软件测试的意义

1.1.1 软件缺陷的典型例子

俗话说，人无完人，金无足赤。人类智慧的结晶——软件也难以尽善尽美，存在缺陷的事情常有发生，可将软件中的所有质量问题都称为软件缺陷。软件缺陷可小可大，然而软件的性质决定了即使是一个很小的缺陷，也有可能给使用者带来极大的损失。下面就是一些造成较大影响的软件缺陷的例子。

1. 千年虫问题

20世纪70年代，当时计算机的存储空间很小，程序员为了缩减程序所占的存储空间，将表示年份的4位数中的前两位去掉，如“1985”被表示为“85”。

这些程序员当然知道到了 2000 年这种表示方法会带来麻烦,然而,他们认为不会等到 2000 年,软件系统就应该更新换代了,所以并没有顾及未来的问题。

不幸的是,这些程序员采用的年份表达方法在 2000 年快要到来时仍被广泛使用,这时千年虫问题才引起世界各国众多行业尤其是银行业、零售业、电信业的高度重视。为解决千年虫问题,避免出现如 2001 年与 1901 年在计算机日期表达中的混淆不清,全世界已付出数千亿美元的代价。

2. 爱国者导弹中的软件缺陷

在第一次海湾战争中,美军最大的一次伤亡是 1991 年 2 月 25 日在沙特阿拉伯的宰赫兰兵营被伊拉克的飞毛腿导弹击中,死 28 人,伤 98 人。其原因是,大名鼎鼎的爱国者导弹的软件中存在缺陷,导弹运行 100 小时共形成 343.3 毫秒的积累误差,导致 687 米的距离偏差,因此未能成功地拦截飞毛腿导弹。

有意思的是,发现这一微小偏差的并不是爱国者导弹的设计者——美国人,而是思维严密的以色列人。

3. 迪斯尼的圣诞节礼物

1994 年圣诞节前夕,迪斯尼公司发布了第一个面向儿童的多媒体光盘游戏——“狮子王童话”。这是迪斯尼公司第一次进军儿童计算机游戏市场,由于该公司的品牌效应以及大力的广告宣传,“狮子王童话”的市场销售情况非常好,该游戏成为大多数父母圣诞节为孩子必买的礼物。

但随后的情况却出人意料。12 月 26 日,很多客户反映该游戏在自己的机器上无法成功安装或无法正常使用。后来才证实,出现这种情况的原因是迪斯尼公司没有针对“狮子王童话”可能使用的各种机型进行系统兼容性测试,只是对少数机型进行了兼容性测试,所以导致该款游戏只能在少数几种机器上成功安装和运行。

4. 阿丽亚娜火箭发射失败

1996 年 6 月,欧洲阿丽亚娜 5 型火箭第一次发射,由于定位软件出错,导致计算机命令固态推进器与主发动机尾喷管发生偏离,结果火箭发射升空仅 40 秒就爆炸了。

5. “冲击波”计算机病毒

2003 年 8 月,“冲击波”计算机病毒首先在美国发作,导致美国政府机关、企业和个人成千上万台计算机受到攻击。随后,“冲击波”蠕虫病毒很快在因特网上广泛传播,中国、日本、欧洲等地的用户也受到了攻击,结果是大量的邮件服务器瘫痪,给整个世界范围内的网络通信带来了惨重的损失。

制造“冲击波”病毒的黑客只用了 3 周时间就完成了该病毒程序。该病毒仅仅利用微软公司 Messenger Service 中的一个缺陷,就攻破了计算机安全屏障,使所有基于 Windows 操作系统的计算机崩溃。更令计算机安全专家担忧的是,如不立即采用有效的防御措施,黑客将很快找到利用该缺陷控制大部分计算机的方法。

随后,微软公司紧急发布了升级补丁,以修复操作系统中存在的缺陷,抵御该病毒的攻击。

6. 微软 64 位服务器软件缺陷

2004 年上半年,微软公司承认,如果客户使用的是 64 位的 Windows Server 2003 企业版,并且硬件配置是英特尔安腾芯片,很可能突然死机。更可怕的情况是,死机后根本不可能重新启动,将给企业带来巨大的损失。

微软公司称,该问题是由硬件管理程序在检测硬件设备时造成的,随后即推出了升级补丁程序。

7. 索尼电视软件缺陷

2006 年 2 月,索尼(中国)公司称,2005 年下半年在中国内地推出的 5 款电视,包括液晶电视和液晶背投电视,由于在软件方面出现了设计缺陷,导致不能正常开关机。

索尼公司的专业人员研究后发现,特定范围内的液晶背投电视和液晶电视的软件中存在一个计时错误,该错误会导致相关型号电视在待机及累计工作约 1200 小时后,出现在使用中不能正常关机或在待机状态下不能开机的现象。而液晶电视的正常工作时间为 5 万小时。随后索尼(中国)有限公司对存在问题的 5 款电视进行了免费软件升级。

1.1.2 软件缺陷的产生原因

软件缺陷给人类带来的损失,是刚才的几个例子远远不能概括的。这种损失有经济上的、精神上的、身体上的,更有许多人因为软件缺陷而付出了生命的代价。

软件缺陷产生的原因到底是什么呢?软件产品是整个软件开发活动的成果结晶,所以我们应该到软件开发活动的各阶段去寻找原因。

按照软件工程的观点,在对软件系统的可行性进行论证之后,软件开发的主要阶段依次为需求分析、软件设计、软件编码、软件测试、软件运行和维护,如图 1-1 所示。据众多软件从业人员的亲身项目实践,得出的结论是,软件需求分析不够全面、准确是导致软件缺陷的最主要原因。

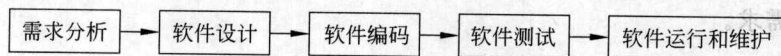


图 1-1 软件开发的主要阶段

需求分析的主要任务是确定待开发软件的功能需求、性能需求及运行环境约束。简单地说,需求分析是确定待开发软件“做什么”。由于软件开发最终是要交付用户使用的,因此所谓“需求”是指用户对软件的需求。系统分析人员和开发人员应在与用户反复、充分沟通的基础上完成需求分析。

由于软件是复杂的逻辑产品,用户在最终拿到软件之前往往很难一次性地精确描述对软件的需求,再加上系统分析人员、开发人员、用户对软件需求的关注角度和描述方式的不同,使得对需求的全面、准确的理解往往不能在需求分析阶段一蹴而就,而是随着软件开发活动的进行不断得到深化。需求分析阶段确定的需求不全面、不准确,即为软件缺陷的产生埋下了祸根。

软件设计和编码过程中的失误也会导致软件缺陷的产生,例如软件设计阶段考虑问题的片面性,软件设计文档不够具体,编码阶段的错误等。但很多情况下,不正确的软件

设计是不正确的需求分析引起的,编码阶段出现的错误则是由需求分析和软件设计不够完善、准确引起的。

1.1.3 软件测试的目标

尽管软件缺陷产生的原因已为很多人所知,但由于软件本身以及软件开发活动的特点,软件缺陷很难根除。所以,在软件交付使用前,为了尽量消除软件中的缺陷,对其进行测试是必不可少的。

软件测试的目标在早期被认为是尽可能多地发现软件中的潜在错误,这一点也可以从 Glenford J. Myers 给出的以下关于软件测试的规则描述中看出。

- 测试是为了发现程序中的错误而执行程序的过程。
- 好的测试方案是尽可能发现迄今为止尚未发现的错误的测试方案。
- 成功的测试是发现了迄今为止尚未发现的错误的测试。

当前仍然有部分人对软件测试存在误解,他们认为软件测试就是要证明软件是正确、可用的,能够满足用户的需要,而不是尽可能多地暴露软件中的潜在错误。

首先,这种想法是行不通的。由于软件是一种复杂的逻辑产品,对软件进行穷举测试是不可能的。因此,即使到目前为止对一个软件的测试中未发现任何错误,也不能说明该软件是绝对正确的,正所谓“软件测试只能证明软件有错,不能证明软件无错”。

从另一个角度说,对软件测试存有这种误解的人在进行测试时,往往在心理上会忽略软件中可能存在的缺陷,而把注意力集中在软件能否完成基本的、已知的功能上,这样的测试显然不是成功的测试。

1983年,在 Glenford J. Myers 观点的基础上, Bill Hetzel 指出,软件测试的目标不仅是尽可能多地发现软件中的错误,还要对软件质量进行度量和评估,以提高软件质量。这一论断将对软件测试的认识提升到更高的层次。

1983年,IEEE 对软件测试的定义则指出,软件测试的目标是为了检验软件系统是否满足用户的需求。



1.2 软件测试中的认识误区

软件测试的概念处于不断的发展之中,再加上缺乏软件测试的理论和实践经验,使得一些人对软件测试的认识存在误区。以下列举对软件测试的一些典型认识误区。

误区 1 测试和调试是一回事。

测试和调试有着根本的不同。测试是一个有计划、可重复的过程,目的是为了发现软件中的潜在错误和缺陷;而调试是一个随机的、不可重复的过程,目的是寻找错误的原因和具体位置,并修复错误。调试一般在测试后进行,当然,调试之后很可能又要进行测试,所以两者常交叉进行。

误区 2 可以对软件进行穷举测试。

对软件进行穷举测试是不可能的。也就是说,不可能对软件进行完全的测试以发现软件中的所有错误和缺陷。

这主要是因为,由于需求规格说明的复杂性和程序逻辑的复杂性,如下的测试是难以

做到的。

- 测试程序中的所有输入条件的取值。
- 测试程序中所有输入条件取值的组合。
- 测试程序中的所有路径。
- 测试出程序中所有潜在的错误和缺陷,例如由于需求分析不完善而导致的错误。

误区3 若交付使用的软件有缺陷,是测试人员的失职。

软件测试的主要目标是发现软件中更多的错误和缺陷,但不能通过已有的测试证明某一个软件是绝对正确的。

交付使用的软件有缺陷,与多方面的人员都有关系,如系统分析人员、设计人员、编码人员、测试人员。软件开发中的任何一个环节出现问题,都有可能使软件出现缺陷。当然,这并不是说软件测试人员可以推卸责任,开发团队中多方面的人员加强沟通、合作才是最重要的。

误区4 关注测试的执行而忽略测试用例的设计。

测试用例(Test Case)是为特定目标开发的一组测试输入、执行条件和预期结果,其目的是测试程序中的某路径,或核实程序或软件是否满足某个特定的需求。

从一定意义上说,设计测试用例是软件测试活动中具有核心地位的一个环节。若不注重测试用例的设计,很可能会遗漏有价值的测试用例,或导致设计出的测试用例不够准确,从而不可能对软件进行充分、有效的测试。

误区5 测试比编程容易得多。

从某种意义上讲,对软件测试人员的要求比编程人员的要求高。这是因为,测试是一件十分复杂的工作。测试人员应具有细致沉稳的性格,很强的专业素质,对被测试软件的功能及架构十分清楚;测试人员还要能在无法实现穷举测试的前提下编写若干有价值的测试用例,以尽可能地揭露软件中的错误和缺陷;在进行自动化测试时,测试人员还应有编写测试脚本的能力。

误区6 测试是编码之后进行的工作。

从软件开发的瀑布模型,人们容易得出这一错误的结论。事实上,软件生命周期中的测试阶段只是表明该阶段的主要工作是测试,并不意味着测试工作不能在需求分析阶段、设计阶段和编码阶段进行。

软件测试不仅是对程序的测试,需求分析和设计也应成为测试的对象。大量测试实践表明,软件中的大部分错误是在编码之前造成的,需求分析和设计造成的软件错误约占所有错误的63%,而编码造成的错误仅占37%。软件中的错误被发现得越晚,为修复它所付出的代价就越大。

所以在软件开发过程中,应尽早地、全面地开展测试。软件测试应成为一个独立的流程,可贯穿到软件开发的各流程,如需求分析、设计、编码,并与之并发地执行。在某流程中,达到恰当的测试就绪点即可进行独立的测试工作。测试还应是可迭代的。只有做到这些,测试才可能是成功的。

误区7 测试自动化是万能的。

软件测试自动化可以提高测试的效率,但成本较高,需要自动化测试工具,还需要测

试人员编写测试脚本等。因此,只有需要经常执行的测试用例才适合于自动化测试。

当对某软件的测试自动化达到一定程度时,再想提高其自动化程度将会变得十分困难,需要付出很大的成本。2/8原则同样适用于测试自动化,即付出20%的成本可以实现80%的测试自动化,若要实现剩余的20%测试自动化,还需付出80%的成本。所以在开发一个软件的过程中,不应盲目地提高其测试自动化的程度。

误区8 软件测试是一种破坏性工作。

一些人认为,以尽可能发现软件中的错误为主要目标的软件测试是一种破坏性工作,是对软件开发人员工作的否定。这种想法在很大程度上制约了软件测试的发展,且十分不利于软件质量的保证和提高。

1.3 软件测试的发展史及现状

1.3.1 软件测试的发展史

软件测试是伴随着软件开发活动的产生而产生的。早在20世纪50年代,英国著名的计算机科学家图灵就给出了软件测试的原始定义。图灵认为,软件测试是程序正确性证明的一种极端的实验形式。这导致人们对软件测试的理解较为狭隘。

当时的软件测试主要针对用机器语言和汇编语言编写的程序,通过设计并运行测试用例,将运行结果与测试人员的预期结果进行比较,从而判断程序的正确性。人们在设计测试用例时常凭经验或直觉,这使得测试用例具有不完备性,测试也不够充分和有效,通过测试之后的软件往往还隐藏着大大小小的缺陷。

当时还有很多人认为,测试就是“调试”,是查找软件中已发现错误的原因并纠正错误的一种活动。人们对软件测试不够重视,在测试上投入的人力、物力也很少,而且测试介入开发过程的时间点很晚,往往是在软件的编码结束后才进行测试。

直到1957年,人们才认识到测试应该是一种发现软件中潜在错误和缺陷的活动。1972年,在北卡罗来纳大学举行了首届软件测试正式会议。1975年,John Good Enough和Susan Gerhart发表了名为《测试数据选择的原理》的文章,使软件测试得到了许多研究者的重视。1979年,Glenford J. Myers的著作《软件测试的艺术》(*The Art of Software Testing*)可以说是软件测试领域一本最重要的专著。Glenford J. Myers在书中对软件测试的定义、目标等进行了描述,他认为测试是为发现错误而执行程序的过程,测试的目的在于尽可能多地发现软件中的错误。

20世纪80年代早期,软件质量日益得到人们的重视。1983年,Bill Hetzel在《软件测试完全指南》(*Complete Guide of Software Testing*)一书中指出:“测试是以评价一个程序或系统属性为目标的任何一种活动,测试是对软件质量的度量。”

1983年,IEEE提出的软件工程专业术语指出:“测试是使用人工或自动的手段来运行或测定某个软件系统的过程,目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别。”这个定义指出了软件测试的目的是检验软件系统是否满足需求。

进入20世纪90年代,软件测试得到了快速发展。随着面向对象分析和设计技术的普遍应用,面向对象的软件测试日益受到人们的重视,面向对象的软件测试的理论和技