



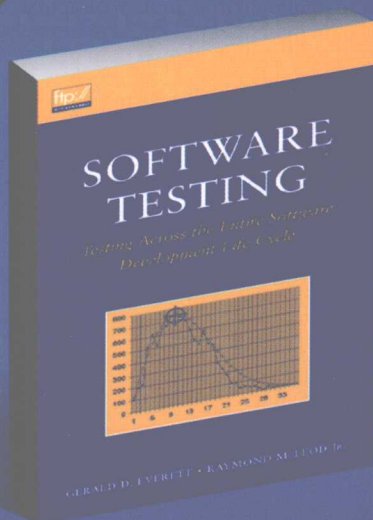
国外经典教材·计算机科学与技术



软件测试

跨越整个软件开发生命周期

Software Testing
Testing Across the Entire Software
Development Life Cycle



Gerald D. Everett
Raymond McLeod, Jr.
郭耀 等译



清华大学出版社

国外经典教材·计算机科学与技术

软件测试

跨越整个软件开发生命周期

Gerald D. Everett

Raymond McLeod, Jr.

著

郭耀等译

清华大学出版社
北京

Gerald D. Everett, Raymond McLeod, Jr.

Software Testing: Testing Across the Entire Software Development Life Cycle

EISBN: 978-0-471-79371-7

Copyright © 2008 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

Simplified Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by McGraw-Hill Education (Asia) Co., within the territory of the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书中文简体字翻译版由美国 John Wiley & Sons, Inc. 公司授权清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾)独家出版发行。未经许可之出口, 视为违反著作权法, 将受法律之制裁。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 John Wiley & Sons 公司防伪标签, 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: 跨越整个软件开发生命周期/(美)爱弗里特(Everett, G. D.)等著; 郭耀等译.
—北京: 清华大学出版社, 2008.9

书名原文: Software Testing: Testing Across the Entire Software Development Life Cycle
(国外经典教材·计算机科学技术)

ISBN 978-7-302-17862-0

I. 软… II. ①爱… ②郭… III. 软件—测试—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2008)第 087909 号

责任编辑: 龙啟铭

责任校对: 徐俊伟

责任印制: 杨 艳

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: 北京密云胶印厂

装 订 者: 三河市李旗庄少明装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 14.5

字 数: 358 千字

版 次: 2008 年 9 月第 1 版

印 次: 2008 年 9 月第 1 次印刷

印 数: 1~3000

定 价: 29.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: 010-62770177 转 3103 产品编号: 028573-01

译者序

软件测试是软件开发过程中不可或缺的阶段，自 20 世纪 70 年代开始，业界就公认，在一个典型的编程项目中，测试会占到一半乃至更多的时间和资源。30 多年来编程语言和软件开发技术的发展日新月异，今天我们拥有了更成熟更丰富的手段来便利地构造精妙复杂的软件，但是测试依然扮演着同样重要的角色。并且，随着计算机软件技术在各行各业的普及运用，人们对软件质量的要求也越来越专业化和多样化。软件开发团队必须尽可能地在交付产品之前控制未来的质量风险，这就必然需要依赖于卓有成效的软件测试。

软件测试远比人们所直观想象的要复杂，哪怕是测试一个简单的系统，也会是一个潜在的旷日持久的任务。在经费和进度紧张的情况下，测试者需要切合实用的技术，亲自实践的经验 and 正确的策略来有效而高效地进行软件测试。

在现实生活中，有效而高效的软件开发技能常常成为业界人员的热门话题和追逐目标，有效而高效的软件测试技能却相对为人们所冷落。正如软件测试泰斗 Glenford Myers 在其经典著作《软件测试艺术》中所言，“我们的学生每年毕业进入业界，几乎全无软件测试方面的知识，甚至，我们几乎从未指导过学生应该如何来测试和调试他们写下的课程程序。”本书作者在对美国 21 所大学进行的一个调查中发现，其中 18 所大学均没有开设任何软件测试方面的课程。这说明软件测试的培训与教育无法满足当今 IT 产业的需求。

这个现象在中国的软件产业中也有所反映，尽管客户对高质量高可靠度软件产品的需求日益增加，尽管软件测试愈来愈为国内业界重视，高效的软件测试工程师依然是各开发团队争相网罗的人才。另一方面，随着敏捷开发方法等新一代软件工程概念的风靡，软件编码和测试过程多次迭代，测试人员更积极地参与到软件生命周期的各个阶段中，使得整个项目团队收取事半功倍的成效。编码和测试人员越来越紧密地协同工作，优秀的程序员必然需要了解测试的方法和概念，优秀的测试人员所具备的技能也不再是仅仅是对现成的程序进行直观的功能测试，软件测试的涵义和策略日益复杂，软件质量风险控制涉及到愈来愈多的方面。故而软件企业对员工进行测试概念、过程、技巧、工具等方面的培训，能显著地提高整个团队的工作效率。

本书就是在这样的产业与教育背景下的产物。本书的原作者旨在为有意走上软件测试工作领域的计算机专业的大学生以及其它专业人士提供一本基础性的、综合性的软件测试教程。我们相信本书所提供的软件测试方面的系统知识一定会为软件测试培训与教育的推广提供很好的促进作用。

本书深入浅出地介绍了软件测试的基本概念与方法，提供成为一个成功的测试专家所必须具备的技术与知识体系。全书中选择丰富有趣的例子贯穿全书，由简单到复杂，让读者可以轻松地掌握测试的关键技术。本书的主要内容如下。

- 引言（第 1 章） 简单介绍了测试的基本目标，从生活中非计算机相关的例子，比如购车过程中出现的与测试类似的活动入手，形象地介绍了测试的基本目标。

- **准备知识（第2~5章）** 简要介绍了本书测试活动所使用的软件开发生命周期模型，所采用的结构化测试技术的基本流程，以及流程中所涉及到的主要问题。
- **常用软件测试技术（第6~9章）** 详细介绍了常用的几类测试技术，包括静态测试、功能测试、结构（非功能）测试、性能测试等。针对每类测试技术中的每一个具体的技术都做了详细的介绍与分析。
- **测试环境与自动化测试工具（第10~11章）** 软件测试离不开好的测试环境与自动化测试工具的支持。
- **测试结果的分析与解释（第12章）** 讲解如何对测试结果进行分析，如何阅读和解释测试结果，特别是采用统计学的方法分析测试结果的方法。
- **一个跨越整个软件开发生命周期的测试实例（第13章）** 通过针对一个应用进行整个软件生命周期的测试的过程的介绍，让读者看到如何把前面介绍的各种测试技术应用到实际的软件测试中。
- **软件测试高级知识（第14~15章）** 包括复杂应用程序的测试，以及测试的未来发展方向的简要探讨。

由上述章节的安排可以看出，本书不仅涉及了软件测试的基础知识与基本技术，同时还介绍了一些普通的测试教程中很少涉及的内容，其中主要包括跨越整个软件开发生命周期的测试方法，以及用统计学的方法来进行测试结果的分析等。这些测试相关的理论有比较高的学术价值，可以为测试实践活动提供更好的指导。

本书适合从事软件测试与软件开发的IT从业人员或者高等院校计算机专业的高年级本科生以及研究生阅读。

本书中文版由北京大学郭耀博士负责翻译工作的总体组织与校稿。参与部分章节翻译工作的包括北京大学的博士生王啸吟、王立杰、刘殿兴、张磊、邵津、邵凌霜、王波等人。

由于译者的水平有限，书中错误与疏漏在所难免，敬请读者批评指正。

译者

谨以此书献给我的妻子 Nell，感谢她在我用周末和假日写书的时候给我持之以恒的鼓励。

Jerry

谨以此书献给我的好朋友 Carolyn，她的提醒、建议、和激励让我成为了一个更好的人、更好的父亲、并更加感激这个世界给予我们的美丽。

Ray

序

作者进行过的一个非官方调查表明：在接受调查的 21 所美国大学中，有 18 所没有开设任何软件测试方面的课程。在走访仅有的三所开设软件测试课程的负责教授时，我们认识到开设软件测试课程的困难很大程度上在于缺乏一本好的教材。我们得知他们目前在选择教材时，需要三四本书组合起来才能覆盖相关的许多话题，然而还有很多重要的话题还是没有办法涉及到。这种情况使得讲课教师需要自己补充很多内容。但如果讲课教师本身缺乏软件测试的背景，那么教材本身的缺陷是很难弥补的。

虽然这样的现状看起来比较令人沮丧，然而相比美国而言，欧洲和亚洲的大学及产业界似乎对软件测试的专业技能更为看重。相比在美国 21 所大学中仅有 3 所开设相关课程，欧洲的大学中大致有七成学校均开设了测试相关课程。这种差别的原因在于他们的文化中对软件测试价值的看重已经不需要借助于一本全面的基础教科书来进行推动了。

软件测试的重要性

本书提供了软件测试的基础性概念和方法。这个话题的重要性在于两方面。第一，按照美国政府的调查，自 2000 年起，低质量软件造成了大约 595 亿美元的损失。第二，虽然现在有价值 222 亿美元的测试人员工作机会，作者却难以找到经验丰富的软件测试人员，这说明现在对经验丰富的软件测试人员的需求仍然很旺盛。

这个话题是值得专门出书论述的。因为就作者的意见来看，目前并没有单独一本全面的软件测试教科书能给刚入门的软件测试人员提供全局性的指导。目前有很多偏重某方面的、非常深入的软件测试书籍，它们可能比较适合于有经验的测试人员，然而这种较为深入的书籍容易让测试新手感到迷茫和沮丧。我们的任务就是为这些测试新手提供一个完整的软件测试概览：包括当前的实践技术、未来的实践技术以及作为选择职业生涯的指导。

本书所采用的方法

本书的写作采用了四个角度相结合的方法。第一，它使用诸如买车这样的非技术性的例子，来阐述一个测试人员的思维方式；第二，它详细介绍了着重于测试规划的结构化测试方法；第三，它考查了如何选择软件测试方法以及它们通常在软件开发生命周期中的何时被使用；最后，它引领读者们从头至尾地跟踪一个软件项目，阐述了如何在这个过程中适当地使用软件测试方法——这些方法在前述章节都是单独进行讨论的。

本书的特色

本书的最大特色在于：

- 本书包括了对于一个专业技术人员想要成为一个软件测试人员所需的知识的全面介绍。本书的表述包括了由浅入深的实例。书中所采用的描述和例子更多是针对实践

者，而非学术研究人员。

- 本书包含专门一章用来讲解分别使用简单的和复杂的数学模型来对于测试结果进行有效地分析。我们还没有看到有其他的软件测试教材使用统计学方法对测试结果进行分析。正相反，其他软件测试教材的作者们认为统计并不属于测试教材的范围。
- 本书采用了两个案例。

案例 A 第一个案例使用了 Sun Microsystems 公司开发的一个很流行的 Internet 应用 PetStore2，该案例用于展示使用 Java 来开发应用系统的最佳实践。通过给读者提供的练习，本书阐述了如何在这个众所周知的应用程序上，规划和执行本书第 7 章到第 12 章中所提出的方法。对读者来说，这样做的好处有两方面。第一，提供给读者一个非常适合动手练习的应用系统，以便于读者获得直观的感受，以加强对本书所描述的方法的认识。第二，当读者成功地完成了本案例的练习之后，他们可以在简历上声称他们拥有对企业所认可的应用系统进行测试的经验。

案例 B 第二个案例（第 13 章）是对一个成功的测试项目的一步一步的描述和练习。这是一个真实的项目，因此在整个过程都会遇到很多困难。这个案例给读者提供了一个将前述章节中独立阐述的概念与方法，以及在案例 A 中的动手实践的内容，进行融会贯通的机会。

本书的组织

第 1 章提供了一个对测试的概述，涉及了以下一些话题：测试的目标和局限性、测试的价值与开销的对比。第 2 章描述了测试所在的系统开发生命周期（SDLC）。讲述了 SDLC 的一些主要模型，如瀑布模型、原型开发、快速应用开发（rapid application development）以及逐步开发方法。本教材使用逐步开发方法作为它的基本软件开发框架。

第 3 章提供了结构化测试的概览，介绍了一种通用的结构化测试方法——SPRAE，这种方法包含下列部分：SPECIFICATION（规约），PREMEDITATION（预先策划），REPEATABILITY（可重复性），ACCOUNTABILITY（责任说明），以及ECONOMY（经济性）。接下来，第 4 章介绍了四个基本测试策略——静态测试、黑盒测试、白盒测试和性能（负载）测试。我们使用二维和三维的“棋盘”来阐述这些基本的策略。当这些基本策略确定之后，测试规划就可以开始进行了——这是我们在第 5 章所阐述的内容。本章我们提供了一些在真实世界中撰写测试计划和测试用例的指南。

第 6~9 章详细解释了第 4 章中所提到的测试的四种基本类型。第 6 章解释静态测试，第 7 章解释功能测试，第 8 章解释结构（非功能）测试，第 9 章解释性能测试。上述章节的内容都是细致而完善的，比如说，对结构化测试的讨论包括了接口测试、安全测试、安装测试，还有一个被称作是冒烟测试（Smoke Test）。

在理解了测试的机理之后，第 10 章将关注点转移到了测试环境上，讲解如何分辨好的和坏的测试环境。接着，第 11 章描述了自动测试工具的重要性。第 12 章讲解了如何分析和解释测试结果。

在打好了上面的基础之后，第 13 章基于本书的第一作者在科罗拉多州的一个项目，介绍了一个完整的软件开发生命周期的案例。

在本书的最后，第 14 章介绍了对复杂系统的测试，第 15 章对测试的未来趋势作了展

望，以期对希望从事软件测试工作的人员有所帮助。

学习辅助

在测试概述一章之后，第 2 章作为本书的基础，介绍本书中方法和技术的概念基础。这一章非常依赖于框架形式的插图，以帮助读者成功的理解这些概念。描述测试过程的章节中也使用了大量的表格和示例，因此读者可以对实际过程有非常直观的了解。

配套网站

本书的配套网站是 ftp://ftp.wiley.com/public/sci_tech_med/software_testing/，由 John Wiley & Sons 有限公司提供，它包括如下内容：

- 每章附带的学习指南和习题，案例 A 和案例 B。
- 一套教师指南，包括课程大纲，用于投影的教材图片，教学目标，教学技巧，讨论题和每一章的思考题。如果你需要访问教师指南，请联系 Wiley Interscience 高级编辑 Paul Petrali，他的电子邮件是 ppetrali@wiley.com。

致谢

从头至尾，作者都使用“我们”作为代词。尽管我们对本书的内容和表现形式负全部责任，我们对我们在成书过程中受到的大量帮助表示感谢。首先，我们感谢我们的数千名学生——包括学校里的和产业界的，他们不仅提供给我们整理和组织材料的机会，更提供给我们大量的反馈意见，使我们不至于走弯路。其次，我们要感谢我们的业务客户，他们为我们提供了真实的实验室，用于应用我们的知识和经验。最后，我们要感谢在 John Wiley & Sons 的相关工作人员，他们为本书的付梓提供了大量的专业经验。我们尤其要感谢 Valerie Moliere、Paul Petrolia、Whitney Lesch 和 Danielle Lacourciere。

致 谢

我们要感谢 Kenneth Everett 花费了大量的时间同我们争论本书中所讲的各种测试方法。有时候是他赢了，有时候是我们赢了。然而通过这样激烈的讨论，不管最后是谁赢了，结果是有好几章的内容得到了很大程度上的巩固。Ken 同时也贡献了书中所包含的案例，为读者理解和掌握测试技术提供更直接的帮助。

我们也要感谢 Stephen Kan 博士，他对于著述的讨论与专业的清晰写作风格给了我们写这本书的灵感。

我们还要感谢我们的出版编辑 Paul Petralia 和他最值得信赖的编辑助手 Whitney Lesch，他们熟练地引领我们征服了出版一本书所涉及的迷宫一样的琐碎事物，完成了这样一本好看的教材，让您有兴趣拿起来看个究竟。

目 录

第1章 测试概述.....1	
1.1 概述.....1	
1.2 测试的目标与局限性.....2	
1.2.1 测试人员的思维.....2	
1.2.2 用户级别的非软件测试 ——买车.....3	
1.2.3 开发人员级别的非软件 测试——造车.....4	
1.2.4 测试的4个主要目标.....7	
1.2.5 开发公理——质量必须 在开发时考虑,而不是 测试时.....9	
1.3 测试的价值与开销.....9	
1.3.1 市场级别的非软件测试—— 车辆安全与销售额.....10	
1.3.2 估算失败的开销.....10	
1.3.3 Basili 和 Boehm 的指数 定律.....11	
1.3.4 Internet 上的“淘金潮”.....13	
1.3.5 电子商务的致命弱点.....13	
1.4 测试与软件开发生命周期的关系.....13	
1.4.1 软件测试作为一项技术 职业的发展史.....13	
1.4.2 好的软件测试的10个 准则.....14	
1.4.3 “GOSSIP”游戏.....18	
1.5 软件测试中的测试人员和开发 人员的角色.....19	
1.5.1 软件质量期望的简史—— “天堂的迷失”.....19	

1.5.2 测试人员在软件开发中的 角色.....20	
1.5.3 测试工具专家在软件 开发中的角色.....20	
1.5.4 测试组中都有谁.....21	
1.6 关于软件测试的观点讨论.....21	
1.7 本章小结.....21	
1.7.1 4个主要的测试目标.....21	
1.7.2 开发公理——质量必须在 开发时而不是测试时考虑..22	
1.7.3 软件测试作为一项技术 职业的发展史.....22	
1.7.4 好的软件测试的10个 准则.....22	
第2章 软件开发生命周期.....24	
2.1 概述.....24	
2.2 方法和工具.....24	
2.3 软件开发生命周期的发展.....25	
2.3.1 经典的SDLC.....25	
2.3.2 原型化方法.....25	
2.3.3 快速应用程序开发.....26	
2.4 逐步开发方法.....27	
2.4.1 生命周期阶段.....27	
2.4.2 系统开发步骤.....28	
2.4.3 在各阶段中的软件测试.....28	
2.5 初始调研阶段.....30	
2.5.1 进行企业分析.....30	
2.5.2 定义系统目的、目标和 性能标准.....31	
2.5.3 评估系统和项目风险.....32	

2.5.4 评估系统和项目的可行性...33	第3章 结构化测试概述..... 47
2.5.5 通过 JAD 会议确认初始 结论.....33	3.1 概述..... 47
2.5.6 获得批准.....34	3.2 软件测试人员的清单..... 48
2.6 分析阶段.....34	3.3 SPRAE——一个通用的结构化 测试方法..... 49
2.6.1 分析功能需求.....34	3.3.1 规约..... 50
2.6.2 分析已有系统文档.....34	3.3.2 预先策划..... 50
2.6.3 访谈.....34	3.3.3 可重复性..... 50
2.6.4 调查研究.....35	3.3.4 责任说明..... 50
2.6.5 举行 JAD 会议.....35	3.3.5 经济性..... 51
2.6.6 观察现有系统.....35	3.4 关于结构化测试的观点讨论..... 52
2.6.7 记录功能需求.....35	
2.7 设计阶段.....36	第4章 测试策略..... 53
2.7.1 寻找可行的设计方案.....37	4.1 概述..... 53
2.7.2 评估可行的设计方案.....38	4.2 测试策略中的“棋子”..... 53
2.7.3 举行 JAD 会议确定并 评估设计方案.....39	4.2.1 静态测试..... 53
2.7.4 选取最优的设计方案.....39	4.2.2 白盒测试..... 54
2.8 初始构建阶段.....39	4.2.3 黑盒测试..... 55
2.8.1 构建每个模块的软件.....40	4.2.4 性能测试..... 55
2.8.2 向用户和项目投资方演示 新的系统模块.....42	4.3 二维测试策略棋盘..... 56
2.9 最终构建阶段.....42	4.4 三维测试策略棋盘..... 60
2.9.1 构建并测试可投产的软件.....42	4.5 关于测试策略的观点讨论..... 62
2.9.2 构建并测试可投产的 数据库或数据仓库.....43	
2.9.3 安装必要的新硬件和 网络连接.....43	第5章 测试计划..... 63
2.9.4 准备好所需的物理设施.....43	5.1 概述..... 63
2.9.5 测试硬件配置.....43	5.2 测试计划..... 63
2.9.6 完成所有文档.....43	5.3 测试用例..... 66
2.9.7 争取用户同意开始运行.....43	5.3.1 测试用例细节..... 67
2.9.8 培训参与人员和用户.....44	5.3.2 动作..... 70
2.10 安装阶段.....44	5.4 在现实世界中编写测试计划与 测试用例..... 71
2.10.1 安装系统组件.....44	5.5 测试文档标准..... 72
2.10.2 移交新系统.....44	5.6 关于测试计划的观点讨论..... 73
2.10.3 进行实施后评估.....44	
2.11 关于 PDM 的观点讨论.....45	第6章 静态测试..... 74
2.12 本章小结.....45	6.1 概述..... 74
	6.2 静态测试的目标..... 74
	6.3 静态测试的备选文档..... 74
	6.3.1 软件开发管理人员文档..... 75

6.3.2 软件开发人员文档.....	76	8.4 安装测试.....	101
6.3.3 测试人员文档.....	76	8.5 冒烟测试.....	101
6.3.4 管理员文档.....	76	8.6 管理测试.....	102
6.3.5 最终用户文档.....	77	8.7 备份和恢复测试.....	102
6.4 静态测试技术.....	77	8.8 关于结构测试的观点讨论.....	103
6.5 跟踪静态测试所发现的错误.....	78	8.9 本章小结.....	103
6.6 关于静态测试的观点讨论.....	78		
第7章 功能测试.....	80	第9章 性能测试.....	105
7.1 概述.....	80	9.1 概述.....	105
7.2 从用况到功能测试用例.....	80	9.2 负载规划技术.....	105
7.3 功能测试方法.....	83	9.2.1 性能负载需求文档.....	106
7.3.1 用户导航测试.....	83	9.2.2 性能负载峰值文档.....	107
7.3.2 事务界面测试.....	83	9.3 负载执行技术.....	108
7.3.3 事务流程测试.....	84	9.3.1 负载上行到峰值.....	109
7.3.4 报告界面测试.....	84	9.3.2 负载从峰值下行.....	109
7.3.5 报告流程测试.....	85	9.3.3 峰值性能的度量.....	110
7.3.6 数据库测试.....	85	9.4 构件性能测试.....	110
7.4 回归测试的方法.....	86	9.5 往返性能.....	110
7.5 白盒测试技术.....	86	9.5.1 一个典型的性能响应 曲线.....	112
7.5.1 语句覆盖技术.....	87	9.5.2 空测试系统中的周六 峰值负载.....	114
7.5.2 分支(简单条件)覆盖 技术.....	87	9.5.3 周六的峰值负载组合.....	115
7.5.3 复合条件覆盖技术.....	87	9.5.4 周六负载的致命缺陷.....	116
7.5.4 路径覆盖技术.....	88	9.5.5 周六负载的致命缺陷的 修正.....	116
7.5.5 循环覆盖技术.....	89	9.5.6 工作日负载性能计划的 执行.....	119
7.5.6 直觉和经验.....	89	9.6 关于性能测试的观点讨论.....	119
7.6 黑盒测试技术.....	91	9.7 本章小结.....	120
7.6.1 等价类技术.....	91		
7.6.2 边界值分析技术.....	92	第10章 测试环境.....	122
7.6.3 预期结果覆盖技术.....	93	10.1 概述.....	122
7.6.4 直觉和经验.....	95	10.2 模拟.....	122
7.7 本章小结.....	96	10.3 基准.....	123
7.8 关于功能测试的观点讨论.....	98	10.4 测试环境.....	123
		10.5 测试环境的目标.....	124
第8章 结构(非功能)测试.....	99	10.6 好的测试环境及使用它们的 原因.....	126
8.1 概述.....	99		
8.2 接口测试.....	99		
8.3 安全测试.....	100		

10.7 不好的测试环境及避免它们的原因.....127	第 12 章 分析并解读测试结果 144
10.8 关于测试环境的观点讨论.....128	12.1 概述 144
10.9 本章小结128	12.2 尝试过的测试用例与成功的测试用例..... 144
第 11 章 自动化测试工具130	12.3 关注单个缺陷的发现..... 147
11.1 概述.....130	12.4 关注缺陷积压日志的发现..... 148
11.2 软件自动化测试工具简史.....130	12.5 关注缺陷集群的发现..... 149
11.3 测试工具记录/回放范型.....132	12.6 以往错误发现模式的作用..... 153
11.4 测试工具触点范型.....134	12.6.1 使用以往项目缺陷历史作为出发点..... 154
11.4.1 功能测试工具的触点范型134	12.6.2 在启动新项目之前利用以往项目的缺陷历史..... 156
11.4.2 性能测试工具的触点范型135	12.6.3 在开始新项目时利用以往项目的缺陷历史..... 157
11.5 测试工具执行范型.....137	12.6.4 在新项目过程中利用以往项目的缺陷历史..... 159
11.6 测试工具带来的好处.....138	12.7 Rayleigh 曲线——缺陷发现模式的瞄准线..... 161
11.6.1 可重复性——投资回报问题138	12.8 更多的缺陷跟踪度量指标..... 164
11.6.2 可重复性的精度——技术问题138	12.9 关于测试结果的观点讨论..... 165
11.6.3 隐藏动作和结果的验证——技术问题.....139	12.10 本章小结..... 165
11.6.4 时间点精度和分布——技术问题139	第 13 章 一个全软件开发生命周期的测试项目 167
11.6.5 完成手动测试不可能完成的任务——投资回报问题140	13.1 概述 167
11.6.6 可复用性（回归测试）——投资回报问题.....140	13.2 初始调研阶段..... 168
11.7 测试工具带来的问题.....140	13.3 分析阶段..... 169
11.7.1 测试工具成本——财务、资源和技巧问题.....140	13.3.1 初始会议和结论 169
11.7.2 一次性的测试安装成本.....141	13.3.2 需求撰写和复审——用况 171
11.7.3 测试工具的能力范围.....141	13.3.3 需求静态测试 172
11.8 关于自动化测试工具的观点讨论.....142	13.3.4 使用静态测试的结果修正和改进需求 172
11.9 本章小结.....142	13.3.5 完成测试策略 174
	13.4 设计阶段..... 175
	13.4.1 支持系统的设计 175
	13.4.2 应用程序的逻辑设计 ... 176
	13.4.3 逻辑设计的静态测试 ... 177
	13.4.4 设计测试计划 177

13.5 初始构建阶段.....180	第 14 章 复杂应用测试..... 193
13.5.1 静态测试.....180	14.1 概述 193
13.5.2 测试环境搭建和测试 数据准备181	14.2 单级应用..... 193
13.5.3 功能测试.....182	14.3 两级应用..... 195
13.5.4 结构测试.....183	14.4 三级应用..... 198
13.5.5 性能测试.....184	14.5 多级应用..... 201
13.5.6 缺陷追踪和分析.....185	14.6 关于复杂应用测试的观点 讨论..... 206
13.5.7 初始构建阶段的结束187	14.7 本章小结..... 206
13.6 最终构建阶段.....188	第 15 章 测试的未来发展方向 208
13.6.1 静态测试.....188	15.1 概述 208
13.6.2 功能测试.....189	15.2 会增加对专业软件测试人员 需求的软件开发的未来发展 方向..... 208
13.6.3 结构测试.....189	15.3 软件测试目前面临的挑战..... 209
13.6.4 性能测试.....189	15.4 软件测试近期面临的挑战..... 209
13.6.5 缺陷跟踪和分析.....190	15.5 软件测试将要面临的挑战..... 210
13.7 实施阶段191	15.6 关于未来测试方向的观点讨论...211
13.8 实施后阶段.....191	15.7 本章小结.....211
13.9 案例结束191	
13.9.1 小结.....192	
13.9.2 案例声明192	
	参考文献..... 213

第 1 章

测试概述

本章学习目标

- 认清一个测试人员的思维方式，不论测试对象是什么
- 明确在商业实践中进行测试的正确动机
- 解释测试在商业实践中被低估的一些原因
- 解释软件测试人员与软件开发人员的区别

1.1 概述

在过去的 10 年中，已经有无数知名的计算机软件著作权成为杂志的精彩封面故事。尽管有这些看得见的教训，劣质软件导致的软件故障依然不断地在后台发生。这些故障每年据估计会对美国经济造成高达 595 亿美元的损失[1]。有估计认为通过在软件开发各个阶段适当地进行测试将可以减少其中每年 222 亿美元的损失[2]。

本书首次全面阐述了针对 21 世纪的开发中或修改中的软件从测试规划到测试实施的各个阶段的测试活动。此书的作者相信，那些封面故事中的软件灾难可以通过实施全面的软件测试来避免。作者还相信，常规而持续的测试可以提高软件开发人员所能提供的软件质量以及客户对质量的预期。通过采用一种全面的测试方法，软件测试人员可以将主要商业损失的负面风险转化为正面的竞争力。

很多市场上优秀的教材深入地探讨了软件开发中某些很窄的特定阶段的测试方法[3-5]。例如本书作者推荐一本可以作为本书后续的中级读本，就是由 James A. Whittaker 博士所著的《测试实用指南（Practical Guide to Testing）》[6]。但是这些教材都没有从整个软件开发生命周期的角度来探讨软件测试，这应该包括在软件开发的每个阶段中的测试规划、测试完成和理解测试结果等方面的问题。

将会从这本书中获益最大的读者是那些软件专业人员，商业系统分析师，以及高年级的计算机专业和信息管理系统专业的学生。这些貌似不同背景的读者群体的共同点就是他们都很关注软件开发过程中的技术挑战。正是在软件开发中的这一共同经验使得这些读者能够很快的对软件测试带来的好处有一个现实的预期，并且可以认清好的软件测试的界限。

尽管本书仅关注于软件测试，但是在第 1 章中介绍的一些测试的基本概念适用于所有产品的测试，无论是汽车还是葡萄酒。这是因为测试是一种任何人可以应用于任何专业或业余任务的思维方式。

计算机硬件测试人员会发现，本书中 85% 的内容也可以直接应用到他们的任务中去。

当然他们需要查找一些额外的参考资料来补充他们所需要的余下的 15% 的特定技术。

注意：区分软件测试和硬件测试最容易的方式就是检查当测试结果是“这一系统运行太慢”时所给出的建议。如果得到的建议是对软件进行调整，或者升级到更多更快的硬件，那么你就正在做的就是软件测试；如果得到的建议是去拿焊枪过来，那么你就是在做硬件测试。

通常来说，一个对软件测试感兴趣的专家一开始往往总是专注于特定类型的测试，例如功能测试。在本章开始所提及的 Whittaker 教材可以作为更深入理解功能测试的下一步读物。然而本书中讨论内容的广度决定了本书更适合于提醒这些专家还有很多其他方面的测试，如果做不好的话常常会影响他们自己所擅长的测试的成功。

1.2 测试的目标与局限性

在一个人的职业与个人生活中，有很多机会要进行测试。我们将首先举出一些与计算机无关的测试的例子来说明进行测试活动的思维和行为模式。然后我们举出一些测试活动会受到的限制，例如费用考虑，时间约束和其他商业限制。

1.2.1 测试人员的思维

Kaner、Bach 和 Pettichord 阐述了一个好的测试人员应表现出来的四种不同的思维能力[7]：

1. 技术思维能力：对技术的建模能力和理解原因与后果的能力
2. 创造思维能力：提出新想法和预见可能性的能力
3. 批判思维能力：评价想法并进行推理的能力
4. 实践思维能力：将想法变成现实的能力

一个有关这些思维能力的例子可以在一则“国王的挑战”的寓言中找到：

国王的挑战（一则寓言）

从前有一天，一个强大的国王想知道他的三个宫廷巫师中，哪个法力最强大。

所以他将这三个巫师囚禁在城堡的地牢中，并宣布那个能够最先逃出地牢的巫师就将获得王国中最强大的巫师的称号。

（在继续阅读之前，你想想如果是你的话会怎样去做？）

第一个巫师马上开始唱魔法术语试图打开他的牢门。

第二个巫师马上开始在地上摆弄小石头和骨头碎片的阵法来寻找如何才能打开牢门的方法。

第三个巫师在牢门对面坐下，用一分钟思考了一下他现在的情形，然后他站起来，走到牢门跟前，过去拉门把手。牢门竟然就这样被打开了，因为门只是被关上了并没有上锁。

所以结果是，第三个巫师最先逃出了地牢，并从此获得了王国中最强大的巫师的称号。