

Software
Testing

软件测试丛书

51CTO
鼎力推荐

软件测试技术大全：

测试基础 流行工具 项目实战（第3版）

Software Testing Guide:
Fundamentals, Tools and Practice

陈能技 黄志国 编著

- 介绍国内外先进的测试技术和测试理念，微软公司的测试方法，RUP中的测试过程，AEP的理念、敏捷测试思想等
- 详细讲述主流的测试工具的使用技术和技巧，如性能测试工具LoadRunner，功能自动化测试工具QTP和TestComplete，测试过程管理工具QC，Visual Studio的单元测试工具，安全测试工具AppScan，测试用例设计工具PICT、CTE XL等
- 介绍典型的开源测试工具的具体应用，如JMeter、White、Watir等
- 介绍测试工具的开发，如使用VBScript进行Web自动化测试、利用UIAutomation实现GUI自动化测试、正交表测试用例自动生成工具等
- 介绍敏捷测试的最佳实践，如持续集成、代码审查、探索性测试等实战技术
- 来自测试实践中的典型案例，如C/S结构测试、B/S结构测试、手机测试、游戏测试、界面测试、数据库测试、Web Service测试、内存泄漏测试等，突出实用性和实战性



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

Software
Testing

软件测试丛书

软件测试技术大全

测试基础 流行工具 项目实战 (第3版)

陈能技 黄志国 编著

人民邮电出版社

北京

图书在版编目 (CIP) 数据

软件测试技术大全：测试基础 流行工具 项目实战 / 陈能技, 黄志国编著. — 3版. — 北京：人民邮电出版社, 2015.8

ISBN 978-7-115-39787-4

I. ①软… II. ①陈… ②黄… III. ①软件工具—测试技术 IV. ①TP311.56

中国版本图书馆CIP数据核字(2015)第163372号

内 容 提 要

本书介绍了国内外先进的测试技术和测试理念, 包括微软的测试方法、RUP 中的测试过程、敏捷测试的理念等; 详细讲述了几个主要的测试工具的使用, 包括 LoadRunner、HP UFT、AppScan 等; 还介绍了各种常用的开源测试工具, 为期待引入开源测试工具的团队提供参考。另外, 还结合项目实践, 介绍了各种测试辅助工具的开发, 包括每日构建框架的开发、UFT 系统的搭建、性能测试框架的搭建、正交表测试用例自动生成工具的设计、数据库比较工具的制作以及分布式配置管理工具 Git 的使用。

本书一些章的最后, 针对测试新手可能碰到的各种疑惑和困难, 给出了精准的分析 and 解答; 而且还特意为测试新手们准备了模拟面试题目, 并为每个问题提供了参考答案, 方便希望进入测试行业的新手们做好应聘准备工作。

-
- ◆ 编 著 陈能技 黄志国
责任编辑 张 涛
责任印制 张佳莹 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
—北京昌平百善印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 36
字数: 870 千字 2015 年 8 月第 3 版
印数: 11 001—14 500 册 2015 年 8 月北京第 1 次印刷
-

定价: 69.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

前 言

目前软件的质量问题几乎都可以归咎为测试阶段没有发现问题，然而事实上我们在测试阶段是不可能发现所有问题的。这当然与软件的复杂度有关系，但是，不规范的测试过程和缺乏测试管理也是造成很多测试不充分、测试遗漏，甚至软件未经测试就匆忙发布的原因。

测试人员本身的素质、技能、测试方法也存在一定的问题。一方面测试人员在抱怨测试环境不佳，另一方面又不断地有新人期待进入这个行业。目前的测试行业有一些浮躁的情况，测试人员的素质参差不齐，测试技能和水平亟待提高。

我们精心编写了本书，目的是指出很多人对测试的各种误解，以及测试过程中的各种误区，并为测试新手进入测试行业提供一个测试知识的阶梯。作者结合多年的测试经验和测试团队管理经验，为广大测试人员介绍了各种先进的测试技术和测试理念，为测试人员提高自己的测试水平、完善自己的知识结构、扩展自己的测试知识面提供了有益的参考。

本书的内容安排

本书分为4篇，共25章，从软件测试的基本概念讲起，再进一步介绍一个完整的测试过程所经历各个阶段，然后结合目前测试流行的各种实用技术和常用工具，讲解如何进行各种类型的测试，最后结合我们的经验讲解如何营造一个良好的学习环境，让测试人员的水平得以不断地提高。

第1篇（第1~3章）软件测试的基础。

本篇讲述了软件测试的基础知识，包括测试起源和发展、测试行业的现状、测试人员的现状以及真正优秀的测试工程师应该具备的素质，为希望进入测试领域的人提供一些基础知识。测试新手能通过这3章的基础知识来判断自己是否适合在测试领域发展，自己目前的不足是什么，需要努力提高的方向是什么。

第2篇（第4~10章）软件测试必备知识。

本篇讲述了与软件测试相关的各种知识，包括软件工程、配置管理、软件测试的目的与原则、软件测试的各种方法、软件测试的具体过程、软件测试的质量度量方法。软件测试是一门需要具备广泛知识的职业，测试人员应该掌握与测试相关的方方面面的知识。这7章的内容是由一名初级测试人员通往测试工程师必备的知识。

第3篇（第11~23章）实用软件测试技术与工具应用。

本篇具体讲述了各种实用的软件测试技术的使用和目前常用的各种测试工具的使用，包括测试管理工具QC的应用、自动化测试工具UFT的应用、性能测试工具LoadRunner的应用、安全测试工具AppScan的应用、单元测试工具MSTest的应用。这几章的内容是作者多年软件测试经验和测试管理经验的总结，其中还重点介绍了一些测试辅助工具的开发，是测试人员综合利用测试技术和测试方法进行各种类型测试的重要参考，也是普通测试工程师通往高级测试工程师需要掌握的核心知识。

第4篇（第24~25章）软件测试的学习和研究。

本篇主要介绍测试人员的发展和提高途径，以及测试团队管理的技巧。测试人员需要找到自己的发展方向，清楚自己的缺点，与其他测试人员一起营造一个共享的交流和学习的氛围。

本书由浅入深，由理论到实践，尤其适合初级读者逐步学习和完善自己的知识结构。

目 录

第 1 篇 软件测试的基础

| | |
|--------------------------|-------------------------|
| 第 1 章 软件测试行业..... 1 | 向技术”的测试..... 17 |
| 1.1 软件测试的起源..... 2 | 2.2.4 测试的划分对敏捷项目开 |
| 1.1.1 第一个 Bug 的故事..... 2 | 发的重要性..... 18 |
| 1.1.2 几个导致严重错误的 Bug... 3 | 2.2.5 如何融入一个项目团队... 18 |
| 1.1.3 软件测试的起因..... 3 | 2.2.6 快速融入项目团队的 |
| 1.2 软件测试的发展..... 3 | 技巧..... 19 |
| 1.2.1 软件调试..... 4 | 2.2.7 尽快投入测试工作的 |
| 1.2.2 独立的软件测试..... 4 | 技巧..... 19 |
| 1.2.3 软件测试的第一次定义... 4 | 2.3 软件测试的团队建设..... 19 |
| 1.2.4 软件测试成为专门的学科... 5 | 2.3.1 学习型团队的组建..... 19 |
| 1.2.5 开发与测试的融合趋势... 5 | 2.3.2 让每一位测试人员找到 |
| 1.2.6 为什么软件测试发展比较 | 适合自己的位置..... 20 |
| 缓慢..... 5 | 2.3.3 “无规矩则不成方圆”... 21 |
| 1.3 软件测试行业的现状和前景..... 6 | 2.3.4 测试规范..... 21 |
| 1.3.1 国内测试行业现状..... 6 | 2.3.5 部门制度..... 22 |
| 1.3.2 测试人员的现状..... 7 | 2.4 小结..... 23 |
| 1.3.3 软件测试的前景..... 8 | 2.5 新手入门须知..... 23 |
| 1.4 小结..... 8 | 2.6 模拟面试问答..... 23 |
| 1.5 新手入门须知..... 9 | 第 3 章 软件测试的人员要求..... 25 |
| 1.6 模拟面试问答..... 9 | 3.1 测试人员的素质要求..... 26 |
| 第 2 章 软件测试的组织..... 11 | 3.1.1 你对测试感兴趣吗..... 26 |
| 2.1 测试的组织形式..... 12 | 3.1.2 你有适合做软件测试的 |
| 2.1.1 微软的经验教训..... 12 | 性格特征吗..... 27 |
| 2.1.2 最简单的软件测试组织... 12 | 3.1.3 好奇心..... 27 |
| 2.1.3 组织形式的分类方式..... 13 | 3.1.4 成就感..... 28 |
| 2.1.4 综合型的测试组织..... 14 | 3.1.5 消极思维..... 29 |
| 2.2 融入测试组织..... 15 | 3.1.6 全面的思维能力..... 29 |
| 2.2.1 根据开发的模式判断自己 | 3.1.7 测试的正确态度..... 29 |
| 的测试角色定位..... 15 | 3.1.8 责任感..... 30 |
| 2.2.2 “支持编码”的测试与“批 | 3.1.9 压力..... 30 |
| 判产品”的测试..... 17 | 3.2 测试人员的技能要求..... 31 |
| 2.2.3 “面向业务”的测试与“面 | 3.2.1 业务知识..... 31 |

| | | | | | |
|-------|-------------------------|----|--------|------------------|----|
| 3.2.2 | 产品设计知识 | 32 | 3.2.9 | 界面设计中的3种模型 | 34 |
| 3.2.3 | 测试人员需要了解软件 架构知识 | 32 | 3.2.10 | 人机交互认知心理学 | 35 |
| 3.2.4 | 测试人员需要了解统一建 模语言(UML) | 32 | 3.2.11 | 测试人员是否需要编程 技能 | 35 |
| 3.2.5 | 测试人员的“武器” | 33 | 3.2.12 | 掌握编程技能的好处 | 35 |
| 3.2.6 | 测试人员需要掌握的测试 工具 | 33 | 3.2.13 | 脚本语言 | 36 |
| 3.2.7 | 测试人员需要掌握开发 工具吗 | 34 | 3.2.14 | 文档能力 | 39 |
| 3.2.8 | 用户心理学 | 34 | 3.3 | 小结 | 40 |
| | | | 3.4 | 新手入门须知 | 40 |
| | | | 3.5 | 模拟面试问答 | 41 |

第2篇 软件测试必备知识

| | | | | | |
|--------|---------------------|----|--------|--------------------|----|
| 第4章 | 软件工程与软件测试 | 42 | 4.2.11 | 极限编程(XP) | 52 |
| 4.1 | 软件工程简介 | 43 | 4.2.12 | XP中的软件测试 | 54 |
| 4.1.1 | 什么是软件工程 | 43 | 4.2.13 | XP工具 | 54 |
| 4.1.2 | 软件的生命周期 | 43 | 4.3 | 不同软件开发模式下的软件 测试 | 54 |
| 4.1.3 | 软件工程的研究领域 | 43 | 4.3.1 | CMM和ISO中的软件 测试 | 54 |
| 4.1.4 | 软件工程的发展历史 | 44 | 4.3.2 | CMM与软件测试 | 54 |
| 4.1.5 | 软件工程化概念的提出 | 44 | 4.3.3 | ISO与软件测试 | 55 |
| 4.1.6 | “软件工厂” | 44 | 4.3.4 | 敏捷开发中的软件 测试 | 56 |
| 4.1.7 | 软件过程管理 | 45 | 4.4 | 小结 | 57 |
| 4.1.8 | 软件过程相关方法和 工具 | 45 | 4.5 | 新手入门须知 | 57 |
| 4.1.9 | 软件工程发展的新趋势 | 46 | 4.6 | 模拟面试问答 | 58 |
| 4.1.10 | 软件工程的目的是 | 46 | 第5章 | 软件配置管理与软件测试 | 61 |
| 4.2 | 软件开发模式 | 47 | 5.1 | 软件配置管理的应用 | 62 |
| 4.2.1 | 常见的软件开发模式 | 47 | 5.1.1 | 什么是配置管理 | 62 |
| 4.2.2 | 线性模型 | 47 | 5.1.2 | 实施软件配置管理的 好处 | 63 |
| 4.2.3 | 渐进式模型 | 48 | 5.1.3 | 配置管理计划 | 63 |
| 4.2.4 | 变换模型 | 49 | 5.1.4 | 配置标识 | 64 |
| 4.2.5 | 软件开发模式的发展 | 49 | 5.1.5 | 变更控制 | 65 |
| 4.2.6 | RUP的历史 | 49 | 5.1.6 | 配置状态记录和报告 | 66 |
| 4.2.7 | RUP过程模型下的软件 测试 | 50 | 5.1.7 | 配置审计 | 66 |
| 4.2.8 | RUP工具 | 51 | 5.1.8 | 配置管理的自动化 | 66 |
| 4.2.9 | “重型”过程VS.“轻量” 过程 | 51 | 5.1.9 | 进度控制与软件测试 | 67 |
| 4.2.10 | 敏捷运动 | 52 | | | |

| | | | | | |
|--------------|-------------------|-----------|--------------|--------------------------|-----------|
| 5.1.10 | 变更控制与软件测试 | 67 | 6.2.3 | SQA 组织 | 84 |
| 5.1.11 | 配置管理与软件测试 | 68 | 6.2.4 | SQA 的工作内容 | 85 |
| 5.2 | VSS 的安装和使用 | 68 | 6.2.5 | QA 与 QC 的区别 | 85 |
| 5.2.1 | VSS 简介 | 68 | 6.3 | 质量保证体系建设 | 86 |
| 5.2.2 | VSS 的安装 | 69 | 6.3.1 | ISO 9000 质量管理体系与八项质量管理原则 | 86 |
| 5.2.3 | 创建 VSS 数据库 | 69 | 6.3.2 | ISO 9000 质量管理体系的建立过程 | 87 |
| 5.2.4 | 创建 VSS 项目 Project | 69 | 6.3.3 | CMM 质量管理体系与过程改进 | 87 |
| 5.2.5 | VSS 备份 | 69 | 6.3.4 | 结合 PSP、TSP 建立 CMM 过程改进体系 | 87 |
| 5.3 | SVN 的安装和使用 | 70 | 6.3.5 | 应用 PDCA 质量控制法持续改进软件质量 | 88 |
| 5.3.1 | SVN 的基本原理 | 70 | 6.4 | 小结 | 88 |
| 5.3.2 | SVN 的下载与安装 | 71 | 6.5 | 新手入门须知 | 89 |
| 5.3.3 | 创建资源库 | 71 | 6.6 | 模拟面试问答 | 89 |
| 5.3.4 | 运行 SVN 服务 | 72 | 第 7 章 | 软件测试的目的与原则 | 91 |
| 5.3.5 | 用户授权 | 72 | 7.1 | 软件测试的目的 | 92 |
| 5.3.6 | 导入项目 | 73 | 7.1.1 | 测试是为了建立软件的信心 | 92 |
| 5.3.7 | 检出项目 | 73 | 7.1.2 | 软件测试与软件信心的关系 | 92 |
| 5.3.8 | 用 add 命令添加文件 | 74 | 7.1.3 | 软件测试的两面性 | 92 |
| 5.3.9 | 用 commit 命令提交文件 | 74 | 7.1.4 | 软件测试的验证与确认 | 93 |
| 5.3.10 | 用 update 命令更新文件 | 74 | 7.1.5 | 测试是一种服务 | 94 |
| 5.3.11 | 将 SVN 服务注册为系统服务 | 74 | 7.2 | 软件测试应该遵循的原则 | 94 |
| 5.3.12 | 远程客户端访问 | 75 | 7.2.1 | Good enough 原则 | 94 |
| 5.3.13 | 目录访问权限控制 | 75 | 7.2.2 | Pareto 原则 | 95 |
| 5.4 | Git 的安装和使用 | 76 | 7.2.3 | 尽可能早开展测试 | 95 |
| 5.4.1 | Git 简介 | 76 | 7.2.4 | 在发现较多错误的地方投入更多的测试 | 96 |
| 5.4.2 | 安装 Git | 77 | 7.2.5 | 同化效应 | 96 |
| 5.4.3 | 远程仓库 | 77 | 7.3 | 小结 | 96 |
| 5.4.4 | 分支管理 | 78 | 7.4 | 新手入门须知 | 97 |
| 5.4.5 | 标签管理 | 78 | 7.5 | 模拟面试问答 | 97 |
| 5.5 | 小结 | 79 | 第 8 章 | 软件测试的方法论 | 99 |
| 5.6 | 模拟面试问答 | 79 | 8.1 | 软件测试的五大流派 | 100 |
| 第 6 章 | 软件质量与软件测试 | 81 | 8.1.1 | 分析学派 | 100 |
| 6.1 | 软件质量属性 | 82 | | | |
| 6.1.1 | 质量的 3 个层次 | 82 | | | |
| 6.1.2 | 软件质量模型 | 83 | | | |
| 6.2 | 软件质量保证与软件测试 | 83 | | | |
| 6.2.1 | SQA 与软件测试 | 84 | | | |
| 6.2.2 | SQA 与项目组各成员之间的关系 | 84 | | | |

| | | | | | |
|------------|------------------|------------|--------|----------------|-----|
| 8.1.2 | 标准学派 | 100 | 9.4 | 测试的设计及测试用例 | 122 |
| 8.1.3 | 质量学派 | 101 | 9.4.1 | 基于需求的测试方法 | 122 |
| 8.1.4 | 上下文驱动学派 | 101 | 9.4.2 | 等价类划分法 | 123 |
| 8.1.5 | 敏捷学派 | 102 | 9.4.3 | 边界值分析法 | 124 |
| 8.1.6 | 不同流派的测试定义 | 102 | 9.4.4 | 等价类+边界值 | 126 |
| 8.2 | 软件测试的方法应用 | 102 | 9.4.5 | 基本路径分析法 | 126 |
| 8.2.1 | 微软公司的第一类测试 | 103 | 9.4.6 | 因果图法 | 127 |
| 8.2.2 | 微软公司的第二类测试 | 103 | 9.4.7 | 场景设计法 | 128 |
| 8.2.3 | 微软的缺陷管理 | 104 | 9.4.8 | 错误猜测法 | 129 |
| 8.3 | IBM公司的软件测试方法 | 104 | 9.4.9 | 正交表与TCG的使用 | 129 |
| 8.3.1 | 回归测试 | 104 | 9.4.10 | 利用均匀试验法设计测试用例 | 131 |
| 8.3.2 | 测试的度量 | 105 | 9.4.11 | 组合覆盖与PICT的使用 | 132 |
| 8.3.3 | 用例驱动 | 105 | 9.4.12 | 分类树与TESTONA的使用 | 134 |
| 8.3.4 | RUP对软件测试的分类 | 105 | 9.4.13 | 测试用例设计的自动化 | 136 |
| 8.3.5 | RUP对测试阶段的划分 | 107 | 9.4.14 | 敏捷测试用例设计 | 137 |
| 8.4 | 自动错误预防(AEP)方法 | 107 | 9.4.15 | 测试用例的粒度 | 137 |
| 8.4.1 | AEP的基本概念 | 107 | 9.4.16 | 基于需求的测试用例设计 | 138 |
| 8.4.2 | 实现软件自动错误预防的五大法则 | 108 | 9.4.17 | 测试用例的评价 | 138 |
| 8.5 | 小结 | 110 | 9.4.18 | 测试用例数据生成的自动化 | 139 |
| 8.6 | 新手入门须知 | 110 | 9.5 | 测试的执行 | 139 |
| 8.7 | 模拟面试问答 | 112 | 9.5.1 | 测试用例的合理选择 | 139 |
| 第9章 | 软件测试的过程管理 | 113 | 9.5.2 | 测试的分工与资源利用 | 140 |
| 9.1 | 软件测试的各个阶段 | 114 | 9.5.3 | 测试环境的搭建 | 140 |
| 9.2 | 测试需求 | 114 | 9.5.4 | BVT测试与冒烟测试 | 141 |
| 9.2.1 | 需求规格说明书的检查要点 | 115 | 9.5.5 | 每日构建的基本流程 | 142 |
| 9.2.2 | 需求文档的检查步骤 | 115 | 9.5.6 | 通过每日构建来规范源代码管理 | 142 |
| 9.2.3 | 通过编写测试用例来检查需求 | 118 | 9.5.7 | 通过每日构建来控制版本风险 | 143 |
| 9.3 | 测试计划 | 119 | 9.6 | 测试的记录和跟踪 | 143 |
| 9.3.1 | 为什么要制定测试计划 | 119 | 9.6.1 | Bug的质量衡量 | 143 |
| 9.3.2 | 测试计划是对测试过程的整体设计 | 119 | 9.6.2 | 如何录入一个合格的Bug | 144 |
| 9.3.3 | 确定测试范围 | 120 | | | |
| 9.3.4 | 制定测试策略 | 120 | | | |
| 9.3.5 | 安排好测试资源 | 121 | | | |
| 9.3.6 | 安排好进度 | 121 | | | |
| 9.3.7 | 计划风险 | 122 | | | |

| | | | | | |
|--------|----------------------|-----|---------------|-----------------------|------------|
| 9.6.3 | 报告发现问题的版本 | 144 | 9.10 | 新手入门须知 | 157 |
| 9.6.4 | 报告问题出现的环境 | 145 | 9.11 | 模拟面试问答 | 157 |
| 9.6.5 | 报告问题重现的操作 步骤 | 145 | 第 10 章 | 软件测试的度量 | 160 |
| 9.6.6 | 描述预期的行为 | 145 | 10.1 | 软件测试度量的目的 | 161 |
| 9.6.7 | 描述观察到的错误行为 | 145 | 10.1.1 | 测试度量的难度 | 161 |
| 9.6.8 | Bug 报告应该注意的几个 问题 | 146 | 10.1.2 | 测试人员工作质量的 鉴定 | 162 |
| 9.6.9 | 如何跟踪一个 Bug 的生命 周期 | 146 | 10.1.3 | 度量的目的 | 163 |
| 9.6.10 | 如何与开发人员沟通一个 Bug | 147 | 10.2 | 软件测试的度量方法及其 应用 | 164 |
| 9.6.11 | Bug 评审要注意的 问题 | 147 | 10.2.1 | Bug 的数量能说明什么 | 164 |
| 9.6.12 | 基于 QC 的缺陷管理 | 148 | 10.2.2 | 度量 Bug 的数量 | 164 |
| 9.7 | 回归测试 | 149 | 10.2.3 | 加权法度量缺陷 | 164 |
| 9.7.1 | 为什么会回归 | 149 | 10.2.4 | Bug 的定性评估 | 166 |
| 9.7.2 | 回归测试的难度 | 149 | 10.2.5 | Bug 综合评价模型 | 166 |
| 9.7.3 | 基于风险的回归测试 | 150 | 10.2.6 | 测试覆盖率统计 | 167 |
| 9.8 | 测试总结和报告 | 151 | 10.2.7 | 代码覆盖率 | 167 |
| 9.8.1 | 缺陷分类报告 | 152 | 10.2.8 | 功能模块覆盖率 | 168 |
| 9.8.2 | 缺陷类型分布报告 | 152 | 10.2.9 | 需求覆盖率 | 169 |
| 9.8.3 | 缺陷区域分布报告 | 152 | 10.2.10 | 测试用例文档产出率与 测试用例产出率 | 170 |
| 9.8.4 | 缺陷状态分布报告 | 152 | 10.2.11 | 考核测试人员的硬指标 和软指标 | 170 |
| 9.8.5 | 缺陷趋势报告 | 153 | 10.2.12 | 硬指标 | 170 |
| 9.8.6 | 典型缺陷与 Bug 模式 | 154 | 10.2.13 | 软指标 | 171 |
| 9.8.7 | 测试中的 PDCA 循环 | 155 | 10.2.14 | 考核表 | 172 |
| 9.8.8 | 客观全面的测试报告 | 156 | 10.3 | 小结 | 173 |
| 9.8.9 | 实用测试经验的总结 | 156 | 10.4 | 新手入门须知 | 173 |
| 9.9 | 小结 | 157 | 10.5 | 模拟面试问答 | 174 |

第 3 篇 实用软件测试技术与工具应用

| | | | | | |
|---------------|----------------------|------------|--------|-----------------------|-----|
| 第 11 章 | 实用软件测试技术 | 175 | 11.2.4 | 手工测试、自动化测试, 一个都不能少 | 179 |
| 11.1 | 软件测试技术的发展 | 176 | 11.2.5 | 自动化测试的目的 | 179 |
| 11.2 | 软件测试技术 | 177 | 11.2.6 | 手工测试的不可替 代性 | 179 |
| 11.2.1 | 不管黑盒、白盒,找到 Bug 就行 | 177 | 11.2.7 | 探索性测试的“技术” | 180 |
| 11.2.2 | 黑盒测试 | 177 | 11.2.8 | 探索性测试的基本过程 | 181 |
| 11.2.3 | 白盒测试 | 178 | | | |

- 11.2.9 探索性测试的管理 181
- 11.2.10 单元测试的定义 182
- 11.2.11 单元测试由谁做 182
- 11.2.12 结对单元测试 183
- 11.2.13 单元级别的性能测试 184
- 11.2.14 性能测试“从小做起” 184
- 11.2.15 数据库性能检查 186
- 11.2.16 软件的“极限考验”——压力测试 187
- 11.2.17 软件的容量如何 187
- 11.2.18 安全性测试 191
- 11.2.19 网页安全漏洞检测 192
- 11.2.20 SQL 注入 192
- 11.2.21 缓冲区溢出 193
- 11.2.22 安装测试 193
- 11.2.23 环境测试 196
- 11.3 实用软件测试技术的综合应用 197
 - 11.3.1 跟踪法测试 197
 - 11.3.2 跟踪法的典型应用 197
 - 11.3.3 跟踪法测试的好处 199
 - 11.3.4 跟踪法测试的必要性 200
 - 11.3.5 C/S 结构软件系统的测试 200
 - 11.3.6 B/S 结构软件系统的测试 201
 - 11.3.7 链接测试 201
 - 11.3.8 Cookies 测试 202
 - 11.3.9 兼容性测试 202
 - 11.3.10 并发访问测试 202
 - 11.3.11 手机应用测试的要点 202
 - 11.3.12 手机应用软件的特点 203
 - 11.3.13 手机应用软件的测试要点 203
 - 11.3.14 游戏软件系统的测试重点是“玩” 204
 - 11.3.15 游戏可玩性的测试 204
 - 11.3.16 游戏的环境测试 205
 - 11.3.17 网络游戏的安全性测试 205
 - 11.3.18 游戏的性能测试 205
 - 11.3.19 界面交互及用户体验测试 205
 - 11.3.20 使用用户模型对界面交互进行测试 205
 - 11.3.21 界面和用户体验测试的要点 206
 - 11.3.22 数据库测试 207
 - 11.3.23 数据库设计的测试 207
 - 11.3.24 SQL 代码规范性测试 207
 - 11.3.25 SQL 语句效率测试 208
 - 11.3.26 SQL 数据库兼容性测试 209
 - 11.3.27 Web Services 的测试 211
 - 11.3.28 内存泄露测试 212
 - 11.3.29 造成软件内存泄露的原因 212
 - 11.3.30 如何检测内存泄露 213
 - 11.3.31 对内存问题测试的分工与合作 213
 - 11.3.32 检查程序员的编码规范 214
 - 11.3.33 报表测试 215
 - 11.3.34 报表测试的业务基础 216
 - 11.3.35 报表测试中的细节问题检查 216
 - 11.3.36 报表测试中的性能测试、安全性测试 216
 - 11.3.37 报表的保存和打印测试 216
 - 11.3.38 报表的格式测试 216
 - 11.3.39 联机帮助和用户手册的测试 217
 - 11.3.40 联机帮助的测试要点 217
 - 11.3.41 用户手册的测试要点 218
 - 11.3.42 缺乏工具支持的性能测试 218

| | | | | | |
|---------------|-------------------------------------|------------|---------------|-------------------------------------|------------|
| 11.3.43 | 借助其他小工具和自己开发的小程序来解决 问题 | 219 | 12.4.3 | 执行测试 | 242 |
| 11.3.44 | 手工的性能测试 | 221 | 12.5 | 缺陷登记与跟踪 | 243 |
| 11.3.45 | 本地化测试与国际化 测试 | 222 | 12.5.1 | 添加新缺陷 | 243 |
| 11.3.46 | 本地化软件测试和国际化 测试的要点 | 222 | 12.5.2 | 如何避免录入冗余的 缺陷 | 244 |
| 11.3.47 | 本地化软件测试和国际化 测试对测试人员的 要求 | 223 | 12.5.3 | Bug 的生命周期 | 245 |
| 11.3.48 | 本地化软件测试和国际化 测试工具的使用 | 223 | 12.5.4 | 把缺陷链接到测试 | 245 |
| 11.3.49 | 可访问性测试 | 224 | 12.6 | 在 QC 中生成测试报告的 图表 | 247 |
| 11.3.50 | Section 508 Web 指南 | 224 | 12.7 | 基于 QC 的测试项目管理 | 247 |
| 11.3.51 | 可访问性测试工具 | 225 | 12.7.1 | QC 的库结构 | 247 |
| 11.4 | 小结 | 225 | 12.7.2 | 创建 QC 项目库 | 248 |
| 11.5 | 新手入门须知 | 225 | 12.7.3 | 定制项目 | 248 |
| 11.6 | 模拟面试问答 | 226 | 12.7.4 | 添加项目组成员 | 248 |
| 第 12 章 | 测试管理工具 QC 的应用 | 229 | 12.7.5 | 自定义 QC 的数据 字段 | 249 |
| 12.1 | 测试管理平台 | 230 | 12.7.6 | 配置跟踪提醒规则 | 252 |
| 12.1.1 | 测试过程管理规范化 | 230 | 12.7.7 | 设置警报规则 | 254 |
| 12.1.2 | 测试管理平台——QC 简介 | 231 | 12.7.8 | 设置 workflow | 255 |
| 12.1.3 | QC 安装 | 231 | 12.7.9 | “缺陷模块”列表 自定义 | 255 |
| 12.2 | 测试需求管理 | 237 | 12.7.10 | 脚本编辑器 | 256 |
| 12.2.1 | 定义测试需求 | 237 | 12.7.11 | QC 项目的导入 导出 | 257 |
| 12.2.2 | 把需求项转换成测试 计划 | 237 | 12.8 | 其他资源 | 259 |
| 12.3 | 测试计划管理 | 238 | 12.9 | 练习和实践 | 259 |
| 12.3.1 | 测试用例的管理 | 238 | 第 13 章 | 功能自动化测试工具 UFT 的 应用 | 261 |
| 12.3.2 | 设计测试步骤 | 239 | 13.1 | 如何开展功能自动化测试 | 262 |
| 12.3.3 | 测试用例的重用 | 240 | 13.1.1 | 选取合适的测试项目来 开展自动化测试 | 262 |
| 12.3.4 | 测试用例对需求项的 覆盖 | 241 | 13.1.2 | 自动化测试工程师的知识 体系 | 262 |
| 12.4 | 测试执行 | 242 | 13.1.3 | 自动化测试工具选型 | 263 |
| 12.4.1 | 定义测试集 | 242 | 13.1.4 | 自动化测试项目计划 | 264 |
| 12.4.2 | 为测试集添加测试 用例 | 242 | 13.2 | 使用 UFT 开展功能自动化 测试 | 265 |
| | | | 13.2.1 | UFT 的安装 | 265 |
| | | | 13.2.2 | 使用 UFT 录制脚本 | 266 |

| | | | | | |
|---------------|-------------------------------|-----|---------------|--------------------------|-----|
| 13.2.3 | 使用关键字视图和专家视图编辑脚本 | 269 | 14.2.12 | 选择需要监控的性能参数 | 302 |
| 13.2.4 | 回放脚本 | 270 | 14.2.13 | 性能测试报告与性能瓶颈分析 | 302 |
| 13.2.5 | 插入检查点 | 271 | 14.3 | 其他资源 | 303 |
| 13.3 | 构建功能自动化测试框架 | 273 | 14.4 | 练习和实践 | 304 |
| 13.3.1 | 模块化框架 | 273 | 第 15 章 | 安全测试 | 305 |
| 13.3.2 | 函数库结构框架 | 275 | 15.1 | 常见安全漏洞分析 | 306 |
| 13.3.3 | 数据驱动框架 | 276 | 15.1.1 | 缓冲区溢出 | 306 |
| 13.4 | 其他资源 | 279 | 15.1.2 | 整数溢出 | 308 |
| 13.5 | 练习和实践 | 279 | 15.1.3 | 命令注入 | 309 |
| 第 14 章 | 性能测试工具 Load Runner 的应用 | 281 | 15.1.4 | SQL 注入 | 309 |
| 14.1 | 如何开展性能测试 | 282 | 15.1.5 | XSS——跨站脚本攻击 | 311 |
| 14.1.1 | 性能测试工程师的素质要求 | 282 | 15.2 | 使用 AppScan 进行安全测试 | 313 |
| 14.1.2 | 认识性能测试 | 283 | 15.2.1 | AppScan 简介 | 313 |
| 14.1.3 | 性能测试的类型 | 284 | 15.2.2 | 利用 AppScan 进行 Web 安全测试 | 313 |
| 14.1.4 | 性能测试成熟度模型 | 285 | 15.2.3 | 使用 AppScan 测试 AltoroJ 项目 | 315 |
| 14.1.5 | 分析和定义性能需求 | 286 | 15.3 | 其他资源 | 318 |
| 14.1.6 | “不成文的”性能需求定义 | 287 | 15.4 | 练习和实践 | 318 |
| 14.1.7 | 计划性能测试 | 287 | 第 16 章 | 单元测试工具 MSTest 的应用 | 320 |
| 14.2 | 使用 LoadRunner 开展性能测试 | 288 | 16.1 | 单元测试范围管理 | 321 |
| 14.2.1 | LoadRunner 简介 | 288 | 16.1.1 | 单元测试的分类 | 321 |
| 14.2.2 | LoadRunner 基本使用方法和步骤 | 290 | 16.1.2 | 静态单元测试 | 322 |
| 14.2.3 | 选择协议 | 292 | 16.1.3 | 动态单元测试 | 322 |
| 14.2.4 | 录制脚本 | 293 | 16.1.4 | “广专结合”、“动静相宜” | 323 |
| 14.2.5 | 解决常见的脚本回放问题 | 296 | 16.1.5 | 单元测试的效果 | 323 |
| 14.2.6 | 修改和完善脚本 | 297 | 16.1.6 | 单元测试的范围 | 323 |
| 14.2.7 | 脚本参数化 | 297 | 16.2 | 单元测试的过程管理 | 324 |
| 14.2.8 | 添加事务 | 298 | 16.2.1 | 单元测试的过程策划 | 324 |
| 14.2.9 | 添加内容检查点 | 299 | 16.2.2 | 管理层对单元测试的重视 | 324 |
| 14.2.10 | 性能参数的选择和监视 | 300 | 16.2.3 | 单元测试意识的改变 | 324 |
| 14.2.11 | 运行场景 | 301 | 16.2.4 | 单元测试的组织 | 325 |
| | | | 16.2.5 | 单元测试模式的选择 | 325 |

- 16.2.6 单元测试的管理规范... 325
- 16.2.7 单元测试的人员分工... 326
- 16.2.8 单元测试的策略... 326
- 16.2.9 单元测试用例的设计... 326
- 16.2.10 代码标准和规范... 326
- 16.2.11 代码审查制度... 327
- 16.2.12 单元测试的流程... 327
- 16.2.13 单元测试与每日构建的结合... 328
- 16.2.14 单元测试的自动化方面... 328
- 16.2.15 自动化单元测试与每日构建的结合... 329
- 16.3 单元测试的质量度量... 330
 - 16.3.1 单元测试覆盖率... 330
 - 16.3.2 单元测试评审... 331
- 16.4 单元测试工具 MStest 的应用... 331
 - 16.4.1 建立单元测试项目... 332
 - 16.4.2 巧用 NMock 对象... 333
 - 16.4.3 对缺乏接口实现的类的方法进行测试... 334
 - 16.4.4 使用 NMock 对象... 334
 - 16.4.5 使用 NMock 的场合... 335
 - 16.4.6 单元测试的执行... 335
 - 16.4.7 测试管理... 335
 - 16.4.8 运行测试代码... 336
 - 16.4.9 查看测试结果... 336
- 16.5 数据驱动的单元测试... 336
 - 16.5.1 为什么要使用数据驱动的方式... 336
 - 16.5.2 创建数据驱动单元测试... 336
 - 16.5.3 使用数据源... 338
 - 16.5.4 使用配置文件定义数据源... 339
 - 16.5.5 编写单元测试代码使用配置文件定义的数据源... 339
- 16.6 小结... 340
- 16.7 新手入门须知... 341
- 16.8 模拟面试问答... 341
- 第 17 章 开源测试工具... 343**
 - 17.1 开源测试工具简介... 344
 - 17.1.1 开源的背景... 344
 - 17.1.2 开源测试工具的发展现状... 344
 - 17.1.3 开源测试工具的分布... 344
 - 17.1.4 开源测试工具的来源... 345
 - 17.1.5 开源测试工具的优势... 345
 - 17.1.6 开源测试工具的不足... 345
 - 17.2 常用开源测试工具介绍——测试管理类... 346
 - 17.2.1 Bugzilla... 346
 - 17.2.2 Mantis... 347
 - 17.2.3 BugFree... 348
 - 17.2.4 综合比较... 349
 - 17.3 常用开源测试工具介绍——单元测试类... 349
 - 17.3.1 NUnit... 349
 - 17.3.2 NMock... 350
 - 17.3.3 NUnitForms... 351
 - 17.4 常用开源测试工具介绍——性能测试类... 352
 - 17.4.1 JMeter... 353
 - 17.4.2 TestMaker... 354
 - 17.4.3 DBMonster... 355
 - 17.5 常用开源测试工具介绍——自动化功能测试类... 357
 - 17.5.1 Abbot Java GUI Test Framework... 357
 - 17.5.2 White... 358
 - 17.5.3 Watir... 359
 - 17.6 如何在测试组中引入开源测试工具... 360
 - 17.6.1 开源测试工具的成本考虑... 361
 - 17.6.2 引入开源测试工具的步骤... 361

| | | | | |
|---------|---------------------------------------|-----|--------|-------------------------------|
| 17.6.3 | 引入开源测试工具可能碰到的问题 | 362 | 动化测试 | 380 |
| 17.7 | 小结 | 363 | 18.4.2 | 利用 UI Automation 实现 GUI 自动化测试 |
| 17.8 | 新手入门须知 | 363 | 18.5 | 设计一个性能测试框架 |
| 17.9 | 模拟面试问答 | 364 | 18.5.1 | 性能测试的基本原理 |
| 第 18 章 | 测试工具的原理及制作 | 366 | 18.5.2 | Controller 的简单设计 |
| 18.1 | 自制测试工具的优势 | 367 | 18.5.3 | Agent 的简单设计 |
| 18.2 | 辅助工具的制作 | 368 | 18.5.4 | 虚拟用户的产生 |
| 18.2.1 | 测试工具的开发策划 | 368 | 18.6 | 正交表测试用例自动生成工具的设计 |
| 18.2.2 | 测试语言的选择 | 368 | 18.6.1 | 正交表类的设计 |
| 18.2.3 | 测试工具开发的各种实现技术 | 369 | 18.6.2 | 加载正交表文件 |
| 18.2.4 | 接口驱动 | 369 | 18.6.3 | 解释输入 |
| 18.2.5 | 测试执行器及远程代理 | 370 | 18.6.4 | 查找正交表 |
| 18.2.6 | 测试解释器和测试生成器 | 371 | 18.6.5 | 改进方向 |
| 18.3 | 利用 Windows 脚本辅助测试 | 371 | 18.7 | 数据库比较工具的制作 |
| 18.3.1 | 利用 JScript 进行简单的 GUI 自动化测试 | 372 | 18.7.1 | “三库”的问题 |
| 18.3.2 | 利用 JScript 检查注册表 | 372 | 18.7.2 | SQL Server 表结构原理 |
| 18.3.3 | 利用 JScript 的 FileSystem Object 对象处理文件 | 373 | 18.7.3 | 数据库比较工具的设计 |
| 18.3.4 | 读取文件 | 373 | 18.8 | Oracle 的 SQL 语句跟踪工具的制作 |
| 18.3.5 | 创建文件 | 374 | 18.8.1 | 设置 Oracle 的 SQL 跟踪参数 |
| 18.3.6 | 利用 JScript 操作 Excel | 374 | 18.8.2 | 打开 SQL 跟踪 |
| 18.3.7 | 在 JScript 中运行应用程序 | 375 | 18.8.3 | 关闭 SQL 跟踪 |
| 18.3.8 | 在 JScript 中使用 WMI | 376 | 18.8.4 | 改进方向 |
| 18.3.9 | 在 JScript 中访问网络 | 377 | 18.9 | 一个简单的猴子测试工具的制作 |
| 18.3.10 | 在 JScript 中使用正则表达式 | 378 | 18.9.1 | 猴子测试工具应该具备的功能 |
| 18.3.11 | 使用 JScript 发送邮件 | 379 | 18.9.2 | Windows API 的调用 |
| 18.3.12 | JScript 脚本的调试方法 | 379 | 18.9.3 | 截屏功能的实现 |
| 18.4 | 简易自动化测试 | 380 | 18.9.4 | 让猴子动起来 |
| 18.4.1 | 使用 VBScript 进行 Web 自 | | 18.9.5 | 记录猴子的足迹 |
| | | | 18.9.6 | 给猴子一些知识 |
| | | | 18.9.7 | 记录被测试应用程序的资源使用情况 |

- 18.9.8 完整的猴子测试工具···421
- 18.9.9 扩展·····427
- 18.10 测试覆盖率辅助管理工具的制作·····428
 - 18.10.1 测试覆盖率管理·····428
 - 18.10.2 需求覆盖率管理·····428
 - 18.10.3 测试用例覆盖率管理···430
 - 18.10.4 功能模块覆盖率管理···431
 - 18.10.5 代码覆盖率管理·····431
 - 18.10.6 数据覆盖率管理·····432
 - 18.10.7 测试覆盖率统计的自动化·····433
 - 18.10.8 测试覆盖率对测试管理的意义·····434
 - 18.10.9 测试覆盖率辅助管理工具的设计·····434
 - 18.10.10 调用 DevPartner 的代码覆盖率统计工具·····435
 - 18.10.11 用 C#来调用 DPAnalysis 执行被测试应用程序··435
 - 18.10.12 测试覆盖率辅助管理工具的使用·····439
- 18.11 小结·····440
- 18.12 新手入门须知·····441
- 18.13 模拟面试问答·····441
- 第 19 章 小工具的使用·····443**
 - 19.1 巧用 Windows 自带的小工具··444
 - 19.1.1 Windows 任务管理器··444
 - 19.1.2 利用 Windows 任务管理器检查进程驻留·····444
 - 19.1.3 利用 Windows 任务管理器检查内存问题·····444
 - 19.1.4 利用 Windows 任务管理器检查网络使用情况·····445
 - 19.1.5 利用 Windows 任务管理器检查 CPU 使用情况·····446
 - 19.1.6 Perfmon 的性能监控···446
 - 19.1.7 NetStat 的网络监视·····449
 - 19.2 免费小工具的妙用·····450
 - 19.2.1 SQL Server 数据库的 SQL 事件探查器·····450
 - 19.2.2 Visual Studio 开发工具的 Spy++·····452
 - 19.2.3 Visual Source Safe 的文件比较器·····453
 - 19.2.4 HTTP 协议包查看器——HTTP Watch·····454
 - 19.2.5 HTML DOM 查看器——IE Developer Toolbar·····455
 - 19.3 小结·····456
 - 19.4 新手入门须知·····456
 - 19.5 模拟面试问答·····457
- 第 20 章 持续集成·····458**
 - 20.1 持续集成简介·····459
 - 20.1.1 持续集成的价值·····459
 - 20.1.2 持续集成包含的过程··459
 - 20.2 利用 Windows 脚本搭建一个每日构建框架·····459
 - 20.2.1 每日构建框架的基本要素·····459
 - 20.2.2 获取源代码·····460
 - 20.2.3 编译源代码·····462
 - 20.2.4 分析编译结果·····462
 - 20.2.5 处理编译结果·····464
 - 20.2.6 发送编译报告·····465
 - 20.2.7 利用 Windows 任务计划来定时启动脚本·····466
 - 20.2.8 每日构建框架的扩展 1——单元测试·····467
 - 20.2.9 每日构建框架的扩展 2——自动化功能测试··472
 - 20.2.10 每日构建框架的扩展 3——每日缺陷简报·····473
 - 20.2.11 缺陷库表结构分析·····473
 - 20.2.12 缺陷统计程序的设计··475
 - 20.2.13 每日构建框架的扩展 4——每日配置管理简报·····479

| | | | | | |
|---------------|---------------------------|-----|---------------|-----------------|-----|
| 20.2.14 | 配置管理的现状 | 480 | 22.1.2 | 探索性测试与即兴测试的区别 | 502 |
| 20.2.15 | 缺陷简报程序的设计 | 480 | 22.1.3 | 探索性测试的意义 | 503 |
| 20.2.16 | 每日构建框架的扩展5 ——每日里程碑预报 | 482 | 22.2 | 如何进行探索性测试 | 503 |
| 20.2.17 | 每日构建框架的其他扩展思路 | 484 | 22.2.1 | 优秀探索性测试人员的基本素质 | 503 |
| 20.2.18 | 每日缺陷率统计 | 484 | 22.2.2 | 测试就是向程序提问 | 504 |
| 20.2.19 | 每日缺陷简报 | 486 | 22.3 | 探索性测试的过程管理和度量 | 505 |
| 20.3 | 利用 Windows 脚本整合一个自动错误预防系统 | 487 | 22.3.1 | 测试组长是“教练” | 505 |
| 20.3.1 | 轻量级的 AEP 框架 | 487 | 22.3.2 | 基于探索任务的测试计划 | 505 |
| 20.3.2 | 把 AEP 系统整合到每日构建框架中 | 487 | 22.3.3 | 探索性测试的“碰头会议” | 506 |
| 20.3.3 | 整合 FxCop | 487 | 22.4 | 小结 | 509 |
| 20.3.4 | 整合 SQL BPA | 489 | 22.5 | 新手入门须知 | 509 |
| 20.3.5 | 测试结果检查和发送 | 489 | 22.6 | 模拟面试问答 | 509 |
| 20.4 | 其他资源 | 490 | 第 23 章 | 用户界面测试管理 | 511 |
| 第 21 章 | 代码审查 | 491 | 23.1 | 用户界面测试的必要性 | 512 |
| 21.1 | 代码审查实践 | 492 | 23.2 | 如何进行用户界面测试 | 512 |
| 21.1.1 | 为什么需要代码审查 | 492 | 23.2.1 | 用户界面测试的时机 | 512 |
| 21.1.2 | 代码静态分析的工作内容 | 493 | 23.2.2 | 后期修改界面的风险 | 513 |
| 21.1.3 | 类型检查 | 493 | 23.2.3 | 界面测试遗漏 | 513 |
| 21.1.4 | 风格检查 | 493 | 23.2.4 | 用户界面测试的要点 | 513 |
| 21.1.5 | 程序理解 | 494 | 23.2.5 | “射箭”原理 | 514 |
| 21.1.6 | Bug 查找 | 495 | 23.2.6 | 减少用户的工作量 | 514 |
| 21.2 | 自动代码审查 | 496 | 23.2.7 | “少就是多” | 514 |
| 21.2.1 | 代码分析工具 PCLint 的应用 | 497 | 23.3 | 用户界面测试原则 | 514 |
| 21.2.2 | PCLint 与 VC6 的整合 | 497 | 23.3.1 | 亲和力 | 515 |
| 21.2.3 | 代码风格审查工具 StyleCop 的应用 | 498 | 23.3.2 | 协助 | 516 |
| 21.2.4 | StyleCop 的设置 | 499 | 23.3.3 | 有效 | 517 |
| 21.3 | 其他资源 | 500 | 23.3.4 | 鼓励 | 518 |
| 第 22 章 | 探索性测试管理 | 501 | 23.3.5 | 熟悉 | 518 |
| 22.1 | 探索性测试的必要性 | 502 | 23.3.6 | 明显 | 519 |
| 22.1.1 | 探索性测试的原理 | 502 | 23.3.7 | 个性化 | 519 |
| | | | 23.3.8 | 安全 | 520 |
| | | | 23.3.9 | 满意 | 520 |
| | | | 23.3.10 | 简单 | 521 |
| | | | 23.3.11 | 支持 | 521 |

| | | | |
|-------------------|-----|-------------------|-----|
| 23.3.12 多样性 | 522 | 23.5 新手入门须知 | 523 |
| 23.4 小结 | 522 | 23.6 模拟面试问答 | 523 |

第4篇 软件测试的学习和研究

| | | | |
|---------------------------------------|-----|---------------------------------------|-----|
| 第24章 软件测试的学习环境 | 525 | 24.3.5 解释自己的工作 | 540 |
| 24.1 学习氛围的建立 | 526 | 24.3.6 尽量减少会产生误会和曲 解的 Bug 报告 | 540 |
| 24.1.1 培训导师制度 | 526 | 24.3.7 与管理层的交流 | 541 |
| 24.1.2 把测试人员的学习内容作 为工作考核的一部分 | 527 | 24.3.8 宣传测试 | 541 |
| 24.1.3 把测试人员的学习计划纳 入项目计划 | 527 | 24.3.9 主动报告测试 | 541 |
| 24.1.4 把测试人员的学习和技术 研究任务化、专门化 | 527 | 24.3.10 外部交流 | 541 |
| 24.1.5 建立一帮一的导师 制度 | 528 | 24.4 小结 | 542 |
| 24.1.6 建立一个持续的培训 体系 | 529 | 24.5 新手入门须知 | 542 |
| 24.1.7 读书会 | 530 | 24.6 模拟面试问答 | 543 |
| 24.1.8 找个师傅学习软件 测试 | 530 | 第25章 软件测试的研究方向与 个人发展 | 545 |
| 24.2 软件测试经验的总结 | 531 | 25.1 软件测试角色与其他项目角色的 可转换性 | 546 |
| 24.2.1 测试知识库的建立 | 531 | 25.1.1 转向售前 | 546 |
| 24.2.2 知识库的“进” | 532 | 25.1.2 转向售后 | 547 |
| 24.2.3 知识库的“出” | 532 | 25.1.3 转向开发 | 549 |
| 24.2.4 办一份内部期刊 | 533 | 25.1.4 转向 QA | 550 |
| 24.2.5 测试管理经验的总结 | 534 | 25.2 测试人员的发展路线 | 551 |
| 24.2.6 过程管理经验总结 | 534 | 25.2.1 管理路线 | 551 |
| 24.2.7 个人管理经验总结 | 536 | 25.2.2 技术路线 | 553 |
| 24.3 软件测试的交流 | 537 | 25.3 软件测试的研究方向 | 554 |
| 24.3.1 日常交流 | 537 | 25.3.1 软件测试中的数学 | 554 |
| 24.3.2 专门的交流 | 538 | 25.3.2 软件测试工具设计 | 555 |
| 24.3.3 与开发人员的交流 | 538 | 25.3.3 其他研究方向 | 555 |
| 24.3.4 定义好自己的角色 | 539 | 25.4 小结 | 555 |
| | | 25.5 新手入门须知 | 556 |
| | | 25.6 模拟面试问答 | 556 |