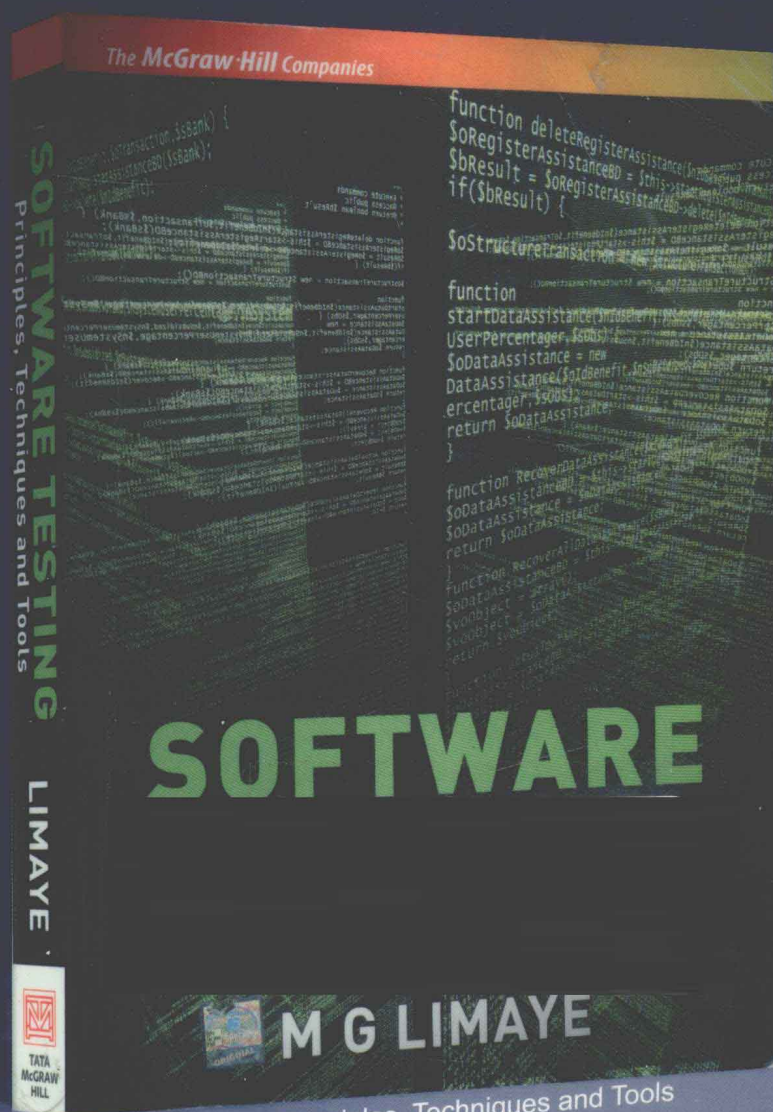


软件测试 原理、技术及工具

(美) M G Limaye 著

黄晓磊 曾琼 译

软件质量分析师、IRCA注册审核员



Software Testing Principles, Techniques and Tools

Education

清华大学出版社

国外计算机科学经典教材

软件测试原理、 技术及工具

(美) M G Limaye 著

黄晓磊 曾 琼 译

清华大学出版社

北 京

M G Limaye

Software Testing: Principles, Techniques and Tools

ISBN: 978-0-07-013990-9

Copyright © 2009 by Tata McGraw-Hill Publishing Company Limited.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education (Asia) and Tsinghua University Press. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2011 by McGraw-Hill Education (Asia), a division of the Singapore Branch of The McGraw-Hill Companies, Inc. and Tsinghua University Press.

版权所有。未经出版人事先书面许可，对本出版物的任何部分不得以任何方式或途径复制或传播，包括但不限于复印、录制、录音，或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和清华大学出版社合作出版。此版本经授权仅限在中华人民共和国境内(不包括香港特别行政区、澳门特别行政区和台湾)销售。

版权©2011 由麦格劳-希尔(亚洲)教育出版公司与清华大学出版社所有。

北京市版权局著作权合同登记号 图字：01-2010-2536

本书封面贴有 McGraw-Hill 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

软件测试原理、技术及工具/(美) 利马耶(Limaye M G) 著；黄晓磊，曾琼 译。

—北京：清华大学出版社，2011.3

书名原文：Software Testing: Principles, Techniques and Tools

(国外计算机科学经典教材)

ISBN 978-7-302-24651-0

I. 软… II. ①利… ②黄… ③曾… III. 软件—测试 IV. TP311.5

中国版本图书馆 CIP 数据核字(2011)第 014758 号

责任编辑：王 军 于 平

装帧设计：孔祥丰

责任校对：胡雁翎

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：北京密云胶印厂

装 订 者：北京市密云县京文制本装订厂

经 销：全国新华书店

开 本：185×260 印 张：32 字 数：779 千字

版 次：2011 年 3 月第 1 版 印 次：2011 年 3 月第 1 次印刷

印 数：1~3000

定 价：59.00 元

产品编号：036888-01

出版说明

近年来,我国的高等教育特别是计算机学科教育,进行了一系列大的调整 and 改革,亟需一批门类齐全、具有国际先进水平的计算机经典教材,以适应我国当前计算机科学的教学需要。通过使用国外优秀的计算机科学经典教材,可以了解并吸收国际先进的教学思想和教学方法,使我国的计算机科学教育能够跟上国际计算机教育发展的步伐,从而培养出更多具有国际水准的计算机专业人才,增强我国计算机产业的核心竞争力。为此,我们从国外多家知名的出版机构 Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 等精选、引进了这套“国外计算机科学经典教材”。

作为世界级的图书出版机构, Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 通过与世界级的计算机教育大师携手,每年都为全球的计算 机高等教育奉献大量的优秀教材。清华大学出版社和这些世界知名的出版机构长期保持着紧密友好的合作关系,这次引进的“国外计算机科学经典教材”便是出自上述这些出版机构。同时,为了组织该套教材的出版,我们在国内聘请了一批知名的专家和教授,成立了专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动,各位委员根据国内外高等院校计算机 科学及相关专业的现有课程体系,并结合各个专业的培养方向,从上述这些出版机构出版的计算机系列教材中精心挑选针对性强的题材,以保证该套教材的优秀性和领先性,避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量,我们为 该套教材配备了一批经验丰富的编辑、排版、校对人员,制定了更加严格的出版流程。本套教材的译者,全部由对应专业的高校教师或拥有相关经验的 IT 专家担任。每本教材的责编在翻译伊始,就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华,在经过翻译、排版和传统的三审三校之后,我们还请编审委员或相关的专家教授对文稿进行审读,以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限,该套教材在出版过程中很可能还存在一些遗憾,欢迎广大师生来电来信批评指正。同时,也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材,共同为 我国高等院校计算机教育事业贡献力量。

清华大学出版社

国外计算机科学经典教材

编审委员会

主任委员:

孙家广 清华大学教授

副主任委员:

周立柱 清华大学教授

委员(按姓氏笔画排序):

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

译者序

软件测试是软件开发过程中必不可少的阶段。软件测试的研究结果表明，软件中存在的问题发现越早，其软件开发费用就越低；软件质量越高，软件发布后的维护费用就越低。另据业界统计，在一个典型的软件项目中，软件测试的费用会占到整个软件工程所有研发费用的 50% 以上。

在现实生活中，尤其是在国内，业界人员往往重视高效的软件开发技能，却忽视了软件测试技能。在很多计算机开发公司中，往往不会把精锐力量投入到软件测试部门中，而测试部分的工作人员也从来不认为自己属于公司的核心部门。但是，随着客户对高质量软件产品需求的日益增加，软件测试越来越得到业界重视。对软件企业来说，不仅要提高对软件测试的认识，同时还要建立起独立的软件测试机构，采用先进的测试技术，充分运用自动化和半自动化测试工具，不断改善软件开发和测试流程，建立完善的软件质量保证体系。只有这样，才能达到软件开发的预期目标，降低软件开发的成本和风险，确保开发出高质量的软件产品。

本书不仅详细介绍了软件测试的原理、技术和工具，同时还用两个具体形象的示例说明了测试场景、测试案例和测试文档，使读者可以完整地了解测试管理、技术和过程需求活动相集成的方法。本书最后附有常用的、测试信息资源、常见问题解答、中英文术语对照。

本书主要由黄晓磊和曾琼翻译完成，全书由李化统稿。由于译者的水平有限，书中可能存在错误与疏漏，敬请读者批评指正。

译者

前言

每个程序开发过程都需要测试。测试可确保程序质量、实现验证和确认或者对可靠性进行评估,从而确保正确实现用户需求。软件测试人员为软件程序和整个系统的效能和效率提供独立的观点。所以,测试团队(程序最终的“质量关卡”)在成功的程序开发中扮演着重要角色。虽然有时将软件测试看作软件开发的一部分,但它自身正在迅速成长成为一门独立的学科。随着软件测试作为一门技术和一种职业逐步被人们接受,现在它正变得更为普及。

本书背景 据观察,人们通常是在未接受任何正规教育的前提下开始软件测试职业生涯的(作为领域专家或软件开发人员)。大学里很少会将“测试”作为专门的培训课程并提供相应的工作方面的培训。这些培训对于掌握软件测试以及软件质量保证和控制中的基本和最新的概念非常重要。此外,目前图书市场上主流的图书无法反映出当前业界的做法,所以缺少软件测试的实践方法。从学校或通过 Internet 获得的知识通常比较简明,无法阐明基本概念或提供全面的解释。

通过对工程和软件中各种机构的质量控制活动进行调查,编写本书的时机已经成熟。软件领域中质量专家的经历使本书作者具备了关于各种质量模型和标准的知识。作为 ERP 和银行软件领域的专家,本书作者开始向软件测试转型。通过不断摸索提高,逐渐掌握了软件测试的技巧。在本书中,作者通过质量保证活动来跟读者分享软件质量控制活动方面的经验。此外,在培训学生参加富有竞争力的考试时,通过跟他们交互,作者还获得了一些特有的感悟,这里将一并分享。

目标读者 本书可作为计算机科学与工程以及信息技术专业本科生的教材,也可以作为 MCA 和 MS(计算机科学)有益的指南。本书反映了软件标准和开发方面最新的趋势,对于希望提高知识基础的质量保证从业人员以及富有抱负的专家来说,本书很有吸引力。本书还可作为软件测试中富有竞争力的考试的参考书。其中软件测试还包括质量和测试管理等内容,例如由各种国际权威机构(像美国的 QAI(Quality Assurance Institute)以及英国的 ISTQB(International Software Testing Qualifications Board)等提供的 Certified Software Test Engineer。

本书根据主流大学以及由国际机构组织的富有竞争力的考试的教学大纲编写而成。跟具体的技术、零售的程序或软件参考书不同,本书遵循了软件测试过程中通用的方法,从而对基本概念、工具和质量保证进行了全面综合。本书使用的术语跟当前业界的规定保持一致。

显著特征 本书的一项重要特征是介绍了一种将测试管理、技术和过程需求活动相

集成的方法。它展示了一种实用的、使用测试工具、测试案例设计和测试过程实现软件测试的方法。面向管理、技术和过程的测试域包括电子商务和移动通信等新技术。本书覆盖了风险分析、最小化、回归测试、数据流测试和评审程序等内容。除了范围广泛的测试话题外,本书还演示了测试的重要性以及测试人员在机警、能干、富有技巧和创造性等方面需要具备的素质。

本书涵盖的内容

- 质量概念,软件测试的基础
- 产品风险和基于测试的风险概念
- 关于测试计划、测试案例编写和缺陷管理的详细介绍
- 不同类型系统的不同测试方法
- 验收测试的详细分析
- 定量分析和尺度的详细介绍
- 评审、走查和检查的详细介绍
- 测试过程改进模型

丰富的教学方法包括:

- 100 多个具体示例
- 200 多道课后练习题
- 50 多个工作访谈(Job Interview)中常见问题解答(FAQ)
- 6 个现实生活中的测试计划

注意:流程图和模板只用于演示目的,具体因场合、机构和客户而异。

组织结构 本书对与软件测试管理和过程(对于交付高质量软件非常关键)有关的话题进行了可行性讨论。本书介绍了将要使用的技术和工具、遵循的过程、测试管理、风险分析、质量保证和与整个软件测试相关的成本。

本书分为 5 部分,共 17 章,另外包含术语表、工作访谈中常见问题解答和附录(包含案例研究)。

第 I 部分 质量保证 该部分介绍软件质量的基本概念。

第 1 章 质量简介 该章介绍质量的基本定义和质量管理的 basic 前提(不同利益相关方的不同的产品质量观),该章还定义了各种质量方法。

第 2 章 软件质量 该章简单介绍了不同产品的质量期望是不同的,它们以质量管理方法为基础。

第 II 部分 软件测试的基本概念 该部分阐明了将软件测试作为一门学科研究时必须了解的基本内容。

第 3 章 软件测试基础 该章介绍“TQM 测试”和“Big Bang”方法之间的差异,定义了测试中使用的不同的方法,例如“黑盒测试”、“白盒测试”和“灰盒测试”。该章最后介绍了测试过程,包括定义测试政策、测试策略和测试计划的过程。

第 4 章 配置管理 该章对软件开发生命周期(SDLC)中工作产品遵循的配置管理过程进行了概述。

第 5 章 风险分析 该章介绍产品风险管理。对使用自动系统替代手动操作时可能的风险进行了讲解。该章还介绍了风险管理过程和用于降低风险的控制机制。

第6章 软件验证与确认 该章介绍验证和确认，它们属于软件测试的两个部分，互为补充。该章还介绍了不同的 SDLC 阶段中验证和确认进行的各种活动的方法。

第7章 V 测试模型 该章建立了“V 模型”或“确认模型”与“VV 模型”或“验证和确认”模型。该章还定义了软件开发中三种关键实体的角色和责任。

第8章 缺陷管理 该章定义缺陷管理过程以及如何使用它们对过程进行改进，该章阐明了整个缺陷生命周期。

第Ⅲ部分 测试技术和工具 该部分介绍 SDLC 过程中各种测试级别以及特殊的功能性测试。此外还介绍了测试生命周期中使用的工具。

第9章 测试层次 该章详细讲述与 SDLC 各个阶段相关的验证和确认活动，同时还包括集成测试的各种形式。

第10章 验收测试 该章详细讲述了验收测试的流程。首先从测试计划和适用于各类系统的测试标准开始，接着对验收测试的流程和各种验收测试方法进行了概括，包括 alpha 测试、beta 测试和 gamma 测试。

第11章 特殊测试(第 I 部分) 该章主要讲述系统测试以外的特殊测试技术，例如 GUI 测试、兼容性测试、国际化测试、安全性测试、性能测试、恢复测试和安装测试。

第12章 特殊测试(第 II 部分) 该章介绍 OO 测试、移动测试、电子商务和电子化企业测试、控制测试、COTS 测试、客户端服务器测试、Web 应用测试和敏捷测试等特殊测试。

第13章 测试工具 该章讲述从外部获得测试工具的过程，即 COTS(商业现货供应)或者厂商自己开发的工具，同时还讲述了选择工具的指导原则，以及各种工具的优点和缺点。

第Ⅳ部分 测试过程 该章介绍真正的测试过程——先是测试政策、测试策略、测试计划和测试案例，最后是测试报告和定量分析。

第14章 测试计划 该章主要讲述在各个层次的测试中产生和使用的测试工件的基础知识，这些测试工件也被称作“测试用具”或者“测试套件”。该章详细讲述了如下概念：

- 测试政策
- 测试策略
- 测试计划
- 测试场景定义
- 测试案例定义
- 测试数据定义

该章还讲述了测试估算技术，包括在机构层面、国家层面、客户层面和国际层面上各种标准的使用。

第15章 测试度量和测试报告 该章介绍在不同验证和确认活动级别中准备的各种报告，同时还讨论了机构中的基准线和测量过程部署。

第16章 定性和定量分析 该章讲述定性和定量的数据收集，以及如何使用定性和定量工具进行数据分析并制定决策。

第Ⅴ部分 测试过程管理

第17章 测试过程改进 该章主要讲述测试过程改进的原因，并详细阐述了各种测试过程改进模型。

在线学习中心 本书对应的 Web 站点为 <http://www.mhhe.com/limaye/st>, 该站点会定期更新, 为教师和学生提供丰富的资源。教师可以从该站点获得习题解答、每章带图表的幻灯片、补充讲解演示的笔记和附加的练习。学生可以访问大量的资源, 例如样例章节、每章的练习、参考资源超链接、附加的测试案例和大量模拟练习。

致谢 特别感谢所有为本书做出贡献的评审人员, 他们对本书进行了深度评论并提出了建设性意见。

为使本书不断得到改进, 欢迎读者提出有价值的建议。请发邮件到 wkservice@vip.163.com。

结构

目标: 快速预览当前章将要讨论的概念。

简介: 介绍基本背景信息。

图和表: 不错的研究工具, 通过可视的图和表来加深对文字材料的理解和记忆。

节和小节: 将每一章细分为节和小节, 这样可使学生通过理论和概念的逻辑组合来掌握主题。

工作示例: 通过贯穿本书的具体示例来加深对关键主题的理解。

实例: 通过丰富的相关示例来演示软件测试概念的应用。

提示: 实用的指导有助于做出正确的决定或者避免软件测试中常见问题的发生。

小结: 简短介绍每一章中重要的概念。

本章回顾: 激励学生应用并集成当前章的内容。

附录 A: 案例研究, 包含详细的测试计划和使用实际例子编写的测试案例。

附录 B: 常见问题解答: 常见问题列表, 包含详细的问题解答(重点关注工作访谈)。

术语表: 在本书末尾的术语表中, 我们按字母顺序定义了 84 个关键词。

目 录

第 I 部分 质量 保 证

第 1 章 质量简介	3
1.1 简介	3
1.2 质量的历史展望	4
1.3 质量的概念(它是事实还是感知)	4
1.4 质量定义	6
1.5 质量的关键要素	7
1.5.1 质量以客户获得产品的满足感为基础	7
1.5.2 机构在实现产品质量之前必需定义好质量参数	8
1.5.3 管理者在机构改进中起领导作用	8
1.5.4 连续(持续)过程改进是必要的	9
1.6 质量观点	10
1.6.1 客户的质量观	11
1.6.2 供应商的质量观	12
1.6.3 用户差异/需求差异	13
1.6.4 生产者差异/开发差异	14
1.7 经济方面的质量	14
1.7.1 制造成本	15
1.7.2 质量成本	15
1.8 质量定义	16
1.9 客户、供应商和过程	16
1.10 全面质量管理	17
1.11 TQM 的质量原则	18
1.11.1 开发各种首创内容定义和部署目的的恒定性	18
1.11.2 通过建立信心和联系来适应管理人员/利益相关方的新哲学	18

1.11.3 从对所得内容和产生的输出进行的大量检查中摆脱出来	18
1.11.4 停止与标有最低价的供应商签订合同	18
1.11.5 改进开发和产品测试中使用的所有过程	19
1.11.6 将对整个机构中所有人的培训形成制度	19
1.11.7 将机构中所有层次的领导能力制度化	19
1.11.8 驱散员工对失败的恐惧	19
1.11.9 打破功能/部门间的障碍	19
1.11.10 消除数字、目标、目的式规劝	20
1.11.11 消除随意的、不被过程支持的数字化目标	20
1.11.12 让员工为工艺而自豪	20
1.11.13 鼓励新技能和新技术的教育	20
1.11.14 高层管理者的承诺和持续改进活动	20
1.12 通过统计过程控制进行质量管理	21
1.12.1 所有层次的质量计划	21
1.12.2 质量控制	21
1.12.3 质量改进	21
1.13 通过文化变化进行质量管理	21
1.14 持续改进周期	22
1.15 不同区域的质量	23
1.16 基准和尺度	24
1.17 解决问题的技术	24

1.18 解决问题的软件工具	25	2.11.2 每次开发的产品一般都是 唯一的	43
1.18.1 使用软件工具进行分析并 作出决策的优点	25	2.11.3 开发过程中产品存在不确定 性,方法也是多种多样	43
1.18.2 使用计算机工具进行分析 并作出决策的缺点	25	2.11.4 不可能进行全面的检查	43
1.18.3 工具	26	2.11.5 无法立刻了解差质量的 效果	44
1.18.4 技术	26	2.11.6 将质量嵌入在产品中	44
1.19 本章小结	26	2.11.7 根据产品/客户的不同,质量 目标也会存在差异	44
1.20 本章回顾	27	2.12 软件质量管理	46
第2章 软件质量	28	2.13 为什么软件会存在缺陷	46
2.1 概述	28	2.14 与软件质量相关的过程	47
2.2 软件产品质量评估的限制	28	2.14.1 愿景	47
2.3 客户是上帝	29	2.14.2 任务	47
2.4 质量和生产率之间的关系	30	2.14.3 政策	47
2.5 产品需求	32	2.14.4 目标	48
2.6 机构文化	33	2.14.5 策略	48
2.7 软件的特征	35	2.14.6 目的	48
2.8 软件开发过程	35	2.14.7 价值	48
2.8.1 瀑布开发方法/模型	35	2.15 质量管理系统的结构	48
2.8.2 迭代开发方法/模型	36	2.15.1 第一层——质量政策	49
2.8.3 增量开发方法/模型	36	2.15.2 第二层——质量目标	49
2.8.4 螺旋开发方法/模型	37	2.15.3 第三层——质量指南	49
2.8.5 原型开发方法/模型	38	2.16 质量管理系统的支柱	49
2.8.6 快速应用开发方法/模型	38	2.16.1 质量过程/质量流程/工作 指令	49
2.8.7 敏捷开发方法/模型	38	2.16.2 指导方针和标准	49
2.8.8 维护开发方法/模型	39	2.16.3 格式和模板	50
2.9 产品类型	39	2.17 质量管理重要的方面	50
2.9.1 生活必需品	40	2.17.1 机构层次的质量计划	50
2.9.2 影响收入的产品	40	2.17.2 项目层次的质量计划	50
2.9.3 只能使用模拟器测试的产品	40	2.17.3 资源管理	50
2.9.4 其他产品	40	2.17.4 工作环境	51
2.10 部分其他重要性定义方案	41	2.17.5 与客户相关的过程	51
2.10.1 从用户角度看	41	2.17.6 质量管理体系文档和数据 控制	51
2.10.2 另外一种来自用户角度的 方法	41	2.17.7 验证和确认	51
2.10.3 开发人员眼中的重要性	41	2.17.8 软件项目管理	51
2.11 软件开发生命周期中存在问题 的领域	42		
2.11.1 需求阶段存在的问题	42		

2.17.9 软件配置管理	52	3.7.2 需求跟踪矩阵存在的问题	71
2.17.10 软件尺度和测量	52	3.7.3 水平跟踪性	71
2.17.11 软件质量审查	52	3.7.4 双向跟踪性	71
2.17.12 分包合同管理	53	3.7.5 垂直跟踪性	72
2.17.13 信息安全管理	53	3.7.6 风险跟踪性	72
2.17.14 管理检查	53	3.8 软件测试的本质	72
2.18 本章小结	54	3.9 工作台	73
2.19 本章回顾	54	3.10 测试过程的重要特征	75
第 II 部分 软件测试的基本概念		3.11 关于测试的误解	77
第 3 章 软件测试基础	59	3.12 软件测试的原则	77
3.1 简介	59	3.13 良好测试的显著特征	78
3.2 软件测试的历史展望	59	3.14 测试政策	79
3.2.1 面向调试的测试	60	3.15 测试策略或测试方法	79
3.2.2 面向演示的测试	60	3.16 测试计划	79
3.2.3 面向破坏的测试	60	3.17 测试过程以及测试中可以发现的缺陷数	81
3.2.4 面向评价的测试	60	3.18 测试团队的效率	81
3.2.5 面向预防的测试	60	3.19 变异测试	82
3.3 测试的定义	61	3.20 测试中的挑战	83
3.4 测试方法	62	3.21 测试团队方式	84
3.4.1 Big bang 测试方法	62	3.21.1 测试团队在机构中的位置	84
3.4.2 全面质量管理方法	62	3.21.2 开发人员变为测试人员	85
3.4.3 全面质量管理与 Big bang 方法	63	3.21.3 独立测试团队	86
3.4.4 成本观中的 TQM	64	3.21.4 由领域专家参与的软件测试	86
3.4.5 Big bang 方法的特征	64	3.22 测试面临的过程问题	87
3.5 流行的测试定义	65	3.23 测试成本方面	89
3.5.1 传统的测试定义	65	3.23.1 测试成本评估	90
3.5.2 什么是测试	66	3.23.2 测试中的预防成本	91
3.5.3 经理的软件测试观	67	3.23.3 测试中的评价成本	91
3.5.4 测试人员的软件测试观	67	3.23.4 测试中的失效成本	91
3.5.5 客户的软件测试观	67	3.24 建立测试政策	92
3.5.6 测试目标	67	3.25 方法	92
3.5.7 测试的基本原则	68	3.26 结构化测试方法	93
3.5.8 成功的测试人员	68	3.27 缺陷的分类	93
3.5.9 成功的测试案例	68	3.27.1 以需求/设计规范为基础	93
3.6 开发生命周期中的测试	69	3.27.2 缺陷类型	93
3.7 需求跟踪矩阵	70	3.27.3 产生缺陷的根本原因	94
3.7.1 需求跟踪矩阵的优点	70	3.28 软件中的缺陷、错误或失误	94

3.29	开发测试策略	94	4.6	配置项	114
3.30	开发测试方法(测试计划)	95	4.7	基线	115
3.30.1	按照以前的定义获取并研究 测试策略	95	4.8	几个与配置库相关的概念	115
3.30.2	确定正在执行的开发项目的 类型	95	4.9	库中配置项的存储	115
3.30.3	确定正在开发的软件系统的 类型	96	4.9.1	分支	115
3.30.4	识别与开发有关的策略性 风险	96	4.9.2	合并	116
3.30.5	确定测试在生命周期中发生 的时机	96	4.9.3	查找历史	116
3.30.6	开发客户化测试策略的 步骤	97	4.9.4	比较	116
3.30.7	开发方法类型影响测试计划 决定	97	4.9.5	回滚	116
3.31	测试过程	98	4.9.6	贴标签	117
3.32	测试态度(常见的人的问题)	99	4.9.7	锁定	117
3.33	测试方法	100	4.9.8	共享	117
3.33.1	黑盒测试(领域测试/规范 测试)	100	4.9.9	冲突处理	117
3.33.2	白盒测试	101	4.10	使用自动配置工具	118
3.33.3	灰盒测试	102	4.10.1	使用工具的优点	118
3.34	软件测试中人的挑战	103	4.10.2	使用工具的缺点	118
3.35	提高测试管理意识	103	4.10.3	常见的自动配置管理 工具	118
3.36	测试人员需要的技能	104	4.11	配置管理计划	119
3.36.1	一般技能	104	4.11.1	引言	119
3.36.2	测试技能	105	4.11.2	范围	119
3.37	本章小结	106	4.11.3	配置管理活动和责任	119
3.38	本章回顾	106	4.11.4	访问特权	120
第4章	配置管理	108	4.11.5	配置管理流程	120
4.1	简介	108	4.11.6	包含命名规范的配置项	120
4.2	配置管理	108	4.11.7	变化控制	120
4.2.1	可配置和不可配置工件	109	4.11.8	工作文件夹定义	121
4.2.2	受管理工件和受控工件	109	4.11.9	注释标准	121
4.3	配置管理周期	109	4.11.10	贴标签标准	121
4.4	配置管理过程	111	4.11.11	工具和用途	121
4.5	审查配置库	114	4.11.12	假定和风险	121
			4.11.13	配置状态报告	121
			4.11.14	跟踪矩阵	122
			4.11.15	集成和构建流程	122
			4.11.16	配置管理审查	122
			4.11.17	维护活动和配置管理	122
			4.11.18	配置管理测量	122
			4.12	本章小结	123
			4.13	本章回顾	123

第 5 章 风险分析	124	5.9 软件实现风险	139
5.1 简介	124	5.9.1 不恰当的进度表和预算	140
5.2 自动化系统的优势	124	5.9.2 开发过程/测试过程资格	140
5.3 自动化系统的弊端	125	5.10 风险识别	140
5.4 风险	125	5.11 软件风险的类型	142
5.5 约束(警告)	126	5.12 测试中风险的处理	143
5.6 项目风险(项目管理风险)	126	5.12.1 固有风险和剩余风险	143
5.6.1 组织因素造成的风险	126	5.12.2 依照现状接受风险	143
5.6.2 政治问题引发的风险	127	5.12.3 避开/避免风险	143
5.6.3 技术问题造成的风险	127	5.12.4 风险预防	144
5.6.4 供应商问题引发的风险	127	5.12.5 风险缓解	144
5.7 产品风险	128	5.12.6 化解风险	144
5.8 因为软件系统而面临的风险	129	5.12.7 意外事件计划	144
5.8.1 开发过程中技术的不合理 使用	129	5.13 风险控制管理措施的类型	144
5.8.2 错误反复出现	129	5.14 风险和测试	146
5.8.3 错误的级联效应(连锁反应)	129	5.14.1 可能性	146
5.8.4 不合逻辑事务的处理	130	5.14.2 严重性	146
5.8.5 缺乏把用户需求转化为系统 技术需求的能力	130	5.15 测试中的假设	147
5.8.6 对开发/使用所选技术缺少 控制能力	131	5.16 可看作风险降低项目的测试 (测试中的优先级)	147
5.8.7 系统中输入错误的数据	131	5.17 测试风险	148
5.8.8 数据聚集在少数几个地方	132	5.17.1 杀虫剂悖论效应	148
5.8.9 缺少对用户需求变化做出快速 反应的能力	132	5.17.2 测试工具的潜在益处和 风险	149
5.8.10 缺少灾难后校验处理的 能力	132	5.18 本章小结	151
5.8.11 人员责任的集中	133	5.19 本章回顾	152
5.8.12 系统接受错误或伪造的输入 数据	133	第 6 章 软件验证与确认	153
5.8.13 系统授权用户误用权限	134	6.1 简介	153
5.8.14 用户/外来者不受控制的系统 访问	134	6.2 验证	154
5.8.15 失败的系统安全设计和用户 隐私行为	135	6.2.1 验证的优点	154
5.8.16 生产/操作过程中的流程 错误	136	6.2.2 验证的缺点	154
5.8.17 通信系统缺陷/失败	138	6.2.3 验证的前提	154
		6.3 验证工作台	155
		6.4 验证的方法	155
		6.4.1 自评	156
		6.4.2 同行评审	156
		6.4.3 上级评审	157
		6.4.4 走查	158

6.4.5	检查	158
6.4.6	审计	161
6.5	阶段性评审的类型	162
6.5.1	过程中评审	162
6.5.2	实现后评审	164
6.6	验证涉及的具体实体	165
6.7	测试生命周期中的评审	165
6.7.1	测试就绪评审	165
6.7.2	测试结束评审	167
6.8	验证中的覆盖(测试设计)	167
6.8.1	语句覆盖	167
6.8.2	路径覆盖	167
6.8.3	判断覆盖	168
6.8.4	判断到判断的路径覆盖 (DD 路径覆盖)	168
6.8.5	好的域和差的域	169
6.9	验证的关注点	169
6.9.1	使用正确的验证技术	169
6.9.2	在 SDLC 中集成验证活动	169
6.9.3	验证可用的资源和技能	170
6.10	确认	170
6.10.1	确认的优点	170
6.10.2	确认的缺点	170
6.10.3	确认的前提	171
6.11	确认工作台	171
6.12	确认层次	172
6.12.1	单元测试	172
6.12.2	集成测试	172
6.12.3	接口测试	172
6.12.4	系统测试	172
6.12.5	因果图	173
6.12.6	路径表达式和正则 表达式	173
6.13	确认中的覆盖范围(优先级/ 基于片的测试)	173
6.14	验收测试	175
6.15	验证和确认(V&V)管理	175
6.15.1	定义验证和确认过程	176
6.15.2	为执行过程准备计划	176

6.15.3	启动实现计划	176
6.15.4	监视计划执行	177
6.15.5	分析执行过程中发现的 问题	177
6.15.6	报告过程进展	177
6.15.7	确保产品满足需求	177
6.16	软件开发验证和确认活动	177
6.17	本章小结	179
6.18	本章回顾	179
第 7 章	V 测试模型	180
7.1	概述	180
7.2	软件的 V 模型	181
7.2.1	结构化测试方法	181
7.2.2	软件开发生命周期每一 阶段的活动	181
7.2.3	分析开发阶段针对充分性和 可测性生成的结构	181
7.2.4	生成基于结构的测试集	181
7.2.5	设计和编码过程中附加的 活动	182
7.2.6	确定结构与之前生成的结构 相一致	182
7.2.7	精练并重定义之前生成的 测试集	182
7.3	提议阶段的测试	182
7.4	需求阶段的测试	183
7.5	测试计划阶段的测试	184
7.6	设计阶段的测试	184
7.7	编码过程中的测试	186
7.8	VV 模型	186
7.9	关键角色和责任	188
7.10	本章小结	189
7.11	本章回顾	190
第 8 章	缺陷管理	191
8.1	概述	191
8.1.1	缺陷的根本原因	191
8.1.2	缺陷的影响	192
8.2	缺陷分类	192

8.2.1 需求缺陷	192	9.15 本章小结	222
8.2.2 设计缺陷	192	9.16 本章回顾	222
8.2.3 编码缺陷	193	第 10 章 验收测试	223
8.2.4 测试缺陷	193	10.1 简介	223
8.3 缺陷管理过程	193	10.2 验收测试标准	224
8.4 缺陷生命周期	195	10.3 验收测试标准的重要性	225
8.5 缺陷模板	196	10.4 一些著名的验收标准	225
8.6 缺陷管理过程(缺陷修复及其 根本原因)	199	10.4.1 测试小时法	225
8.7 评估缺陷预期的影响	201	10.4.2 测试案例数目	226
8.8 为什么风险管理需要风险 讨论	203	10.4.3 缺陷密度法	226
8.9 缺陷发现技术	203	10.4.4 缺陷密度的 P-S 矩阵	226
8.10 缺陷报告	203	10.4.5 容许缺陷标准	227
8.11 本章小结	204	10.4.6 缺陷发现率	227
8.12 本章回顾	205	10.4.7 基于覆盖的验收标准	228
第 III 部分 测试技术和工具		10.4.8 生命周期验收	228
第 9 章 测试层次	209	10.5 alpha 测试	229
9.1 简介	209	10.5.1 alpha 测试的优点	229
9.2 方案测试	210	10.5.2 alpha 测试的缺点	229
9.3 需求测试	210	10.6 beta 测试	230
9.4 设计测试	212	10.6.1 beta 测试的优点	230
9.5 代码审查	212	10.6.2 beta 测试的缺点	230
9.6 单元测试	213	10.7 gamma 测试	230
9.7 模块测试	214	10.7.1 gamma 测试的优点	231
9.8 集成测试	214	10.7.2 gamma 测试的缺点	231
9.8.1 自底向上测试	214	10.8 软件开发阶段的验收测试	231
9.8.2 自顶向下测试	216	10.9 alpha 和 beta 验收测试过程的 一些思考	232
9.8.3 改进的自顶向下方法	217	10.9.1 选择执行测试的用户	232
9.9 大棒测试	218	10.9.2 选择场所	233
9.10 三明治测试	219	10.9.3 测试持续时间	233
9.10.1 三明治测试的流程	219	10.9.4 双边计划和协定	233
9.10.2 三明治方法的优点	219	10.10 软件验收的作用	233
9.10.3 三明治方法的缺点	220	10.11 客户在验收测试中的职责	233
9.11 关键路径优先	220	10.12 验收测试匹配	234
9.12 子系统测试	220	10.13 定义验收标准	235
9.13 系统测试	220	10.14 需求的关键程度	236
9.14 测试阶段	221	10.15 影响需求关键程度的因素	236
		10.16 制定验收测试计划	238