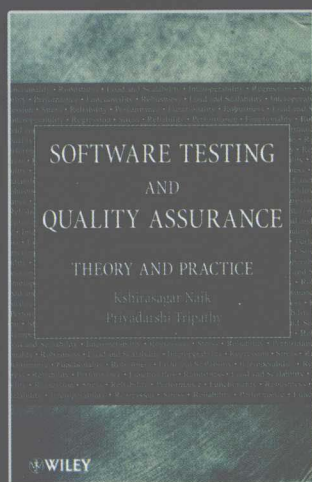




软件测试与质量保证

——理论与实践

Software Testing and Quality Assurance
Theory and Practice



[加] Kshirasagar Naik 著
[美] Priyadarshi Tripathy

郁 莲 等译

国外计算机科学教材系列

软件测试与质量保证

——理论与实践

Software Testing and Quality Assurance
Theory and Practice

[加] Kshirasagar Naik
[美] Priyadarshi Tripathy 著

郁 莲 等译

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是有关软件测试与质量保证的一本理想入门教材,全书共分18章,主要内容包括测试的基本概念、程序测试理论、单元测试、控制流测试、数据流测试、域测试、系统集成测试、系统测试分类、功能测试、从有限状态机模型生成测试、系统测试设计、系统测试计划与自动化、系统测试的执行、验收测试、软件可靠性、测试团队的组织结构、软件质量及成熟度模型等。本书的两位作者在软件测试方面有着丰富的实践经验,全书既有侧重测试理论的概念讲解,也有实际应用的经验总结。

本书既可作为高等院校相关专业软件测试、质量保证、软件工程等课程的教材,也可作为软件测试爱好者的自学用书。对于那些希望增强软件测试方面知识的程序员、软件项目经理和软件开发团队的其他人员,本书也具有很好的参考价值。

Software Testing and Quality Assurance: Theory and Practice, Kshirasagar Naik, Priyadarshi Tripathy
Copyright © 2008 by John Wiley & Sons, Inc.

All rights reserved. This translation published under license.

AUTHORIZED TRANSLATION OF THE EDITION PUBLISHED BY JOHN WILEY & SONS, INC.

No part of this book may be reproduced in any form without the written permission of John Wiley & Sons, Inc.

Copies of this book sold without a Wiley sticker on the back cover are unauthorized and illegal.

本书简体中文版专有翻译出版权由美国 John Wiley & Sons, Inc. 授予电子工业出版社。未经许可,不得以任何手段和形式复制或抄袭本书内容。

本书封底贴有 John Wiley & Sons, Inc. 防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2009-0639

图书在版编目(CIP)数据

软件测试与质量保证:理论与实践/(加)奈克(Naik, K.), (美)特里帕蒂(Tripathy, P.)著;郁莲等译.

北京:电子工业出版社,2013.1

(国外计算机科学教材系列)

书名原文:Software Testing and Quality Assurance: Theory and Practice

ISBN 978-7-121-19013-1



I. ①软… II. ①奈… ②特… ③郁… III. ①软件-测试-高等学校-教材 ②软件质量-质量管理-高等学校-教材
IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2012)第 278234 号

策划编辑:马 岚

责任编辑:冯小贝

印 刷:

装 订:北京季蜂印刷有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:26.5 字数:748 千字

印 次:2013 年 1 月第 1 次印刷

定 价:69.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

译者序

越来越多的软件项目呈现规模大型化、技术复杂化的趋势，包含几十人、上百人的开发团队，成千上万的模块和接口，跨地域、跨拥有权域等情况屡见不鲜；社交网络、移动互联网、云计算、物联网等各种新技术、新应用层出不穷。然而，软件质量保证、软件测试专业人才，尤其是合格的专业人才的严重匮乏日益显现。如何培养合格的软件测试及软件质量保证人才成为大家关注的焦点。

该书致力于为软件工程师、软件质量专业人员、软件开发人员和学生提供测试理论和通用测试方法的基础性培养。以有价值的工作成果巧妙地平衡理论与实践，并提供了丰富的教学辅助内容，包括习题、示例、教学建议和章节总结，该书对于专业人员来说是有价值的、自成体系的参考手册，对于软件测试、质量保证和软件工程的课程来说是一部理想的入门教材。

该书有3种不同的学习模式：(1)只对学习软件测试概念及运用这些概念感兴趣的人来说，建议阅读第1章(“基本概念和预备知识”)、第3章(“单元测试”)、第7章(“系统集成测试”)、第8章至第14章(与系统测试有关)。(2)对致力于提高团队的测试效果的测试经理来说，可以阅读第1章、第3章、第7章、第8章至第14章、第16章(“测试团队的组织结构”)、第17章(“软件质量”)、第18章(“成熟度模型”)等章节。(3)初学者应该从头到尾阅读本书。

该书的两位作者有着丰富的行业实践经验和大学教学经验，为教材的优良品质奠定了坚实的基础。第一作者 Kshirasagar Naik 博士曾是印度班加罗尔地区 Wipro 科技公司的一名软件开发工程师，现在是加拿大安大略省滑铁卢大学电气与计算机工程学院的副教授，在软件测试领域的很多杂志及会议上发表过文章。第二作者 Priyadarshi Tripathy 博士曾在 Nortel Networks、Cisco Systems 和 Airvana 等公司从事软件测试和质量保证的工作，现在是 NEC 美国实验室有限公司的一名资深经理，负责基于网格存储的应用程序的软件测试，他也在软件测试领域发表过很多文章。

参加本书翻译工作的有北京大学软件与微电子学院的李磊、孔常柱、张乐、相慧如、范向东、易玥、赵文博、曹宇奇、鹿昕姝、赵静等人，全书最后由郁莲统稿和审校。由于时间仓促，错误之处在所难免，敬请读者不吝赐教。

前 言

karmany eva dhikaras te; ma phalesu kadachana; ma karmaphalahetur bhur; ma te sango stv akarmani.

Your right is to work only; but never to the fruits thereof; may you not be motivated by the fruits of actions; nor let your attachment to be towards inaction.

(你的权利就只是工作,但从不因此给你带来成果。不要因为一项工作的成果而去完成它,也不要因此而无所作为。)

——*Bhagavad Gita*(薄伽梵歌)

在过去的二十多年里,我们已经见证了软件行业的迅猛发展。软件应用已由最初的数据处理和科学计算领域衍生到我们的日常生活中,只不过我们没有意识到,即使像打电话、启动汽车、打开微波炉或者使用借记卡付款这类看似普通的事情,其实已经执行了某种软件程序。软件生产过程必须面临两大挑战。第一,过程必须在短时间内生产出低成本软件,以便公司能够保证自己的竞争力;第二,过程必须生产实用、可靠并且安全的软件,这些属性通常称为质量属性。软件质量影响我们日常生活中的一些重要方面,例如经济、个人与国家安全及身体健康。

二十多年前,在一个软件开发项目中,测试占据了约50%的总开发时间,以及总耗费资金的50%以上,现在依旧如此。那时软件行业还比较小,学术界为本科生设立了一门叫做“软件工程”的综合课程,教授一些软件开发的基础知识。虽然几十年里软件测试已成为经典软件工程文献的一部分,但很少将软件工程课程纳入主流的学科中。一些大学已经开始提供三个软件工程的专业课程选项,包括需求规范、软件设计及测试与质量保证。除此之外,一些高校推出了完整的软件工程领域的本科生和研究生的学位课程。

考虑到软件质量的影响,我们注意到软件测试的教学并没有得到其应有的位置。在理想情况下,研究工作应引领相关测试工具与方法的开发,以生产出低成本、高质量的软件,而且学生应进行测试基础的学习。换句话说,软件测试的研究不应该仅仅是学术性的,而必须努力对该行业的消费者具有实用价值。然而在现实中,产业所需要的测试技术与我们在大学中所教授和研究的测试技术相差甚远。

我们的目标是为学生与老师提供一系列全面的教学资料,包括测试理论的基础培养及行业中常用的测试实践的基础培养。因为软件质量的概念相当广泛,我们打算为学生提供测试与质量保证的“宏图”。不同的软件系统有着自己错综复杂的特点。我们并没有针对不同的复杂系统展开详细的测试过程讨论。相反,我们提出的测试理论和实践成为了宽阔的垫脚石,这将使学生了解更复杂的系统及相应的测试方法。

我们决定根据在软件测试和质量保证上的教学和行业经验来编写本书,在过去的十几年间,Sagar已经定期讲授软件工程与软件测试的课程,而Piyu一直活跃在执行测试与管理测试团队上,包括测试路由器、交换机、无线数据网络、存储网络和入侵防御设备。相关的经验帮助我们选择和构建这本书的内容,使其适合作为一本教材。

读者对象

我们编写本书是为了向学生和软件专业人员介绍测试理论、测试技术、测试实践及质量保证的基本思想。对于在软件工程、计算机科学及计算机工程上没有软件行业经验的本科生，我们将使用循序渐进的方式一步步为其介绍主题内容。从业人员也将受益于结构性的介绍和综合性的材料。研究生可以使用这本书作为参考文献。完整阅读本书，读者将会对下列主题有一个透彻的了解：

- 测试理论和基础概念
- 支持高质量软件生产的实践
- 软件测试技术
- 需求、缺陷、测试用例和测试结果的生命周期模型
- 单元测试、集成测试、系统测试和验收测试的过程模型
- 建立测试团队，其中包括招聘、培训测试工程师和保留人才的方法
- 质量模型、能力成熟度模型、测试成熟度模型及测试过程改进模型

阅读建议

本书的目的是教会读者怎样进行软件测试。在第1章我们介绍了一些非常重要的背景材料，把软件质量问题的阐述放到本书的后面。对于初学者来说，在对软件测试是做什么的有一个明确认识之前，讨论软件质量的定义是有困难的。但是，对于有着丰富测试经验的从业人员来说，他们可以直接从第1章跳到第17章“软件质量”去阅读。

根据一个人的兴趣，本书有3种不同的阅读方式。第一种情况，对于那些只对软件测试概念感兴趣，并且想要运用这些概念的读者来说，应该阅读第1章（“基本概念和预备知识”）、第3章（“单元测试”）、第7章（“系统集成测试”）、第8章至第14章（与系统测试有关）。第二种情况，对于测试经理来说，兴趣点在于提高团队的测试效果，可以阅读第1章、第3章、第7章、第8章至第14章、第16章（“测试团队的组织结构”）、第17章（“软件质量”）、第18章（“成熟度模型”）等章节。第三种情况，初学者应该从头到尾阅读本书。

教师参考

本书可以作为软件测试基础与质量保证入门课程的教材，在滑铁卢大学，本书的作者之一在“软件测试与质量保证”的研究生课程中使用本书的内容进行教学已有数年之久。软件测试的一门基础课程可以涵盖除第16章之外的大部分内容。对于一门更加强调测试技术而不是过程的课程而言，我们推荐选择第1章（“基本概念和预备知识”）至第15章（“软件可靠性”）来授课。当选择本书作为软件工程课程的补充教材时，在以下章节中选择一部分可以帮助学生理解软件测试的重要基础概念。

- 第1章：基本概念和预备知识
- 第3章：单元测试
- 第7章：系统集成测试

- 第 8 章:系统测试分类
- 第 14 章:验收测试

致谢

在撰写本书的过程中,我们得到了很多人的大力支持,包括出版社、我们的亲人、朋友和同事。他们给予了我们各种各样的帮助。首先,我们想感谢本书的编辑:Anastasia Wasko、Val Moliere、Whitney A. Lesch、Paul Petralia 及 Danielle Lacourciere,他们提出了很多专业的指导意见,并且耐心回答了我们的各种问题。我们的朋友 Alok Patnaik 博士阅读了全部草稿,给出了非常多的建议,大大提高了本书的质量,在此感谢他的所有努力及鼓励。第二作者 Piyu Tripathy 想感谢他之前在 Nortel Networks、Cisco Systems 和 Airvana 等公司的同事们,以及现在 NEC 美国实验室有限公司的同事们。

最后,我们需要特别提到父母、岳父母及伴侣给予的支持。我, Piyu Tripathy, 要感谢亲爱的妻子 Leena, 她独自承担了许多家务和家庭责任,从而给予我写作这本书的时间。我, Sagar Naik, 要感谢我可爱的妻子 Alaka 一直在身边给予我宝贵的支持。还要感谢我漂亮迷人的女儿们 Monisha 和 Sameeksha, 还有活泼的儿子 Siddharth, 他们在我撰写这本书时给予了理解。我非常感谢我的哥哥 Gajapati Naik 给予我的所有支持。我们非常高兴现在有更多的时间和家人及朋友们相处。

Kshirasagar Naik
滑铁卢大学
滑铁卢

Priyadarshi Tripathy
NEC 美国实验室有限公司
普林斯顿

目 录

第 1 章 基本概念和预备知识	1
1.1 质量革命	1
1.2 软件质量	4
1.3 软件测试的作用	4
1.4 验证和确认	5
1.5 失败、错误、故障和缺陷	6
1.6 软件可靠性的概念	7
1.7 测试的目标	7
1.8 什么是测试用例	7
1.9 期望结果	8
1.10 完备测试的概念	9
1.11 测试的核心问题	9
1.12 测试活动	9
1.13 测试阶段	11
1.14 测试用例选择的信息来源	12
1.15 白盒测试与黑盒测试	13
1.16 测试计划和设计	14
1.17 监测并度量测试执行	15
1.18 测试工具和自动化	16
1.19 测试团队的组织和管理	17
1.20 本书概述	18
参考文献	19
习题	21
第 2 章 程序测试理论	22
2.1 测试理论中的基本概念	22
2.2 Goodenough 和 Gerhart 理论	22
2.2.1 基本概念	23
2.2.2 测试理论	24
2.2.3 程序错误	24
2.2.4 可靠性条件	25
2.2.5 理论的缺点	26
2.3 Weyuker 和 Ostrand 理论	26
2.4 Gourlay 理论	27
2.4.1 几个定义	28
2.4.2 测试方法的能力	29
2.5 测试的足够性	30

2.6	测试的局限性	31
2.7	总结	32
	文献综述	33
	参考文献	33
	习题	34
第3章	单元测试	35
3.1	单元测试的概念	35
3.2	静态单元测试	36
3.3	缺陷预防	41
3.4	动态单元测试	42
3.5	变异测试	44
3.6	调试	46
3.7	极限编程中的单元测试	48
3.8	JUnit:单元测试框架	49
3.9	单元测试工具	52
3.10	总结	55
	文献综述	55
	参考文献	57
	习题	59
第4章	控制流测试	60
4.1	基本概念	60
4.2	控制流测试概述	60
4.3	控制流图	61
4.4	控制流图中的路径	64
4.5	路径选择标准	65
4.5.1	全路径覆盖原则	66
4.5.2	语句覆盖原则	66
4.5.3	分支覆盖原则	67
4.5.4	谓词覆盖原则	68
4.6	生成测试输入	69
4.7	选择测试数据的示例	73
4.8	包含无效路径	74
4.9	总结	74
	文献综述	75
	参考文献	75
	习题	76
第5章	数据流测试	78
5.1	一般概念	78
5.2	数据流异常	78
5.3	动态数据流测试概述	80

5.4	数据流图	81
5.5	数据流术语	83
5.6	数据流测试标准	84
5.7	数据流测试选择标准的比较	86
5.8	可行路径和测试选择标准	87
5.9	测试技术的比较	89
5.10	总结	90
	文献综述	90
	参考文献	92
	习题	93
第6章	域测试	95
6.1	域错误	95
6.2	域错误测试	97
6.3	域的源	97
6.4	域错误的类型	99
6.5	ON 点和 OFF 点	100
6.6	测试选择标准	102
6.7	总结	106
	文献综述	107
	参考文献	108
	习题	108
第7章	系统集成测试	110
7.1	集成测试的概念	110
7.2	接口的不同类型和接口错误	111
7.3	系统集成测试的粒度	113
7.4	系统集成技术	113
7.4.1	增量法	114
7.4.2	自顶向下	116
7.4.3	自底向上	118
7.4.4	三明治方式和大爆炸方式	119
7.5	软件和硬件的集成	120
7.5.1	硬件设计验证测试	120
7.5.2	硬件/软件兼容性矩阵	122
7.6	系统集成的测试计划	124
7.7	现成组件集成	126
7.7.1	现成组件的测试	127
7.7.2	内置测试	128
7.8	总结	128
	文献综述	129
	参考文献	129
	习题	130

第 8 章 系统测试分类	131
8.1 系统测试分类法	131
8.2 基础测试	132
8.2.1 启动测试	132
8.2.2 升级/降级测试	132
8.2.3 发光二极管测试	132
8.2.4 诊断程序测试	133
8.2.5 命令行界面测试	133
8.3 功能性测试	133
8.3.1 通信系统测试	134
8.3.2 模块测试	134
8.3.3 记录和跟踪测试	134
8.3.4 元素管理系统(EMS)测试	135
8.3.5 管理信息基础测试	137
8.3.6 图形用户界面测试	137
8.3.7 安全性测试	137
8.3.8 特征测试	138
8.4 健壮性测试	138
8.4.1 边界值测试	139
8.4.2 电力循环测试	140
8.4.3 在线插入和删除测试	140
8.4.4 高可用性测试	140
8.4.5 降级节点测试	140
8.5 互操作性测试	141
8.6 性能测试	141
8.7 可扩展性测试	142
8.8 压力测试	143
8.9 负载和稳定性测试	144
8.10 可靠性测试	145
8.11 回归测试	145
8.12 文档测试	146
8.13 规章测试	146
8.14 总结	147
文献综述	148
参考文献	149
习题	149
第 9 章 功能测试	151
9.1 Howden 的功能测试的概念	151
9.1.1 变量的不同类型	152
9.1.2 测试向量	156
9.1.3 在上下文中测试一个函数	157

9.2	应用功能测试的复杂度	157
9.3	成对测试	159
9.3.1	正交矩阵	159
9.3.2	参数顺序算法	162
9.4	等价类划分	164
9.5	边界值分析	167
9.6	判定表	167
9.7	随机测试	170
9.8	错误猜测	173
9.9	范畴划分	173
9.10	总结	175
	文献综述	176
	参考文献	177
	习题	178
第 10 章	从有限状态机模型生成测试	180
10.1	面向状态的模型	180
10.2	控制和观察点	182
10.3	有限状态机	183
10.4	从一个 FSM 中产生测试	185
10.5	迁移遍历方法	185
10.6	带有状态验证的测试	188
10.7	唯一输入-输出序列	189
10.8	区分序列	192
10.9	表征序列	195
10.10	测试架构	197
10.10.1	本地架构	198
10.10.2	分布式架构	199
10.10.3	协调架构	199
10.10.4	远程架构	200
10.11	测试和测试控制标记法第 3 版(TTCN-3)	200
10.11.1	模块	200
10.11.2	数据声明	201
10.11.3	端口及组件	202
10.11.4	测试用例裁定	203
10.11.5	测试用例	203
10.12	扩展的有限状态机	205
10.13	从 EFSM 模型中生成测试用例	209
10.14	为系统测试增加的覆盖标准	213
10.15	总结	215
	文献综述	215

参考文献	216
习题	217
第 11 章 系统测试设计	219
11.1 测试设计要点	219
11.2 需求识别	220
11.3 可测需求的特征	225
11.4 测试目标识别	227
11.5 示例	228
11.6 建立测试设计流程模型	235
11.7 建立测试结果模型	236
11.8 测试设计准备度量	238
11.9 测试用例设计有效性	238
11.10 总结	239
文献综述	239
参考文献	240
习题	241
第 12 章 系统测试计划与自动化	242
12.1 系统测试计划的结构	242
12.2 导言与特征描述	243
12.3 假设前提	243
12.4 测试方法	243
12.5 测试套件结构	243
12.6 测试环境	244
12.7 测试执行策略	245
12.7.1 多周期系统测试策略	246
12.7.2 测试周期特性	247
12.7.3 首次测试周期的准备	249
12.7.4 最终测试周期的测试用例选择	251
12.7.5 测试用例优先级排序	252
12.7.6 三个测试周期的细节	253
12.8 测试工作量评估	256
12.8.1 测试用例数量	257
12.8.2 创建测试用例工作量	261
12.8.3 测试用例执行工作量	261
12.9 安排进度与测试里程碑	262
12.10 系统测试自动化	266
12.11 评价和选择自动化测试工具	266
12.12 自动化测试选择指南	268
12.13 自动化测试用例的特性	269
12.14 自动化测试用例的结构	270
12.15 测试自动化的基础设施	271

12.16	总结	272
	文献综述	273
	参考文献	275
	习题	276
第 13 章	系统测试的执行	277
13.1	基本观点	277
13.2	缺陷建模	277
13.3	系统测试开始前的准备工作	282
13.4	跟踪系统测试的度量指标	285
13.4.1	监测测试用例执行的度量指标	285
13.4.2	测试执行度量指标的示例	286
13.4.3	监测缺陷报告的度量指标	288
13.4.4	缺陷报告度量指标示例	289
13.5	正交缺陷分类	292
13.6	缺陷原因分析	294
13.7	beta 测试	296
13.8	首次客户运送(FCS)	297
13.9	系统测试报告	298
13.10	产品支持	299
13.11	测量测试的有效性	300
13.12	总结	302
	文献综述	303
	参考文献	303
	习题	304
第 14 章	验收测试	306
14.1	验收测试的类型	306
14.2	验收标准	307
14.3	验收标准的选择	313
14.4	验收测试计划	313
14.5	验收测试的执行	314
14.6	验收测试报告	315
14.7	极限编程中的验收测试	316
14.8	总结	317
	文献综述	317
	参考文献	318
	习题	318
第 15 章	软件可靠性	320
15.1	什么是可靠性	320
15.1.1	缺陷和故障	321
15.1.2	时间	321

15.1.3	故障之间的时间间隔	322
15.1.4	在周期性时间间隔中的故障计数	322
15.1.5	故障密度	323
15.2	软件可靠性的定义	324
15.2.1	软件可靠性的第一个定义	324
15.2.2	软件可靠性的第二个定义	325
15.2.3	比较软件可靠性的定义	325
15.3	影响软件可靠性的因素	325
15.4	软件可靠性的应用	326
15.4.1	比较软件工程技术	326
15.4.2	衡量系统测试的进度	326
15.4.3	控制运行中的系统	327
15.4.4	深入洞察软件开发过程	327
15.5	操作概要	327
15.5.1	操作	327
15.5.2	操作概要的表示	327
15.6	可靠性模型	329
15.7	总结	333
	文献综述	334
	参考文献	335
	习题	335
第 16 章	测试团队的组织结构	337
16.1	测试团队	337
16.1.1	集成测试组	337
16.1.2	系统测试组	337
16.2	软件质量保证组	338
16.3	系统测试团队的组织结构	339
16.4	有效的测试工程师人员编制	340
16.5	招聘测试工程师	342
16.5.1	职位需求	342
16.5.2	职位概要	342
16.5.3	浏览简历	343
16.5.4	协调面试团队	343
16.5.5	面试	344
16.5.6	决策	347
16.6	保留测试工程师	347
16.6.1	职业前景	347
16.6.2	培训	347
16.6.3	奖励制度	348
16.7	团队建设	348
16.7.1	期望	348

16.7.2	一致性	348
16.7.3	信息共享	349
16.7.4	标准化	349
16.7.5	测试环境	349
16.7.6	酬谢	349
16.8	总结	349
	文献综述	350
	参考文献	350
	习题	351
第 17 章	软件质量	352
17.1	软件质量的 5 个观点	352
17.2	McCall 的质量因子和标准	354
17.2.1	质量因子	354
17.2.2	质量标准	357
17.2.3	质量因子和质量标准之间的关系	358
17.2.4	质量度量	359
17.3	ISO 9126 质量特性	359
17.4	ISO 9000:2000 软件质量标准	361
17.4.1	ISO 9000:2000 基本法则	362
17.4.2	ISO 9001:2000 需求	363
17.5	总结	366
	文献综述	367
	参考文献	368
	习题	368
第 18 章	成熟度模型	369
18.1	软件过程中的基本概念	369
18.2	能力成熟度模型	370
18.2.1	CMM 架构	371
18.2.2	5 个成熟度级别和核心过程领域	371
18.2.3	核心实践的公共特性	373
18.2.4	CMM 应用	374
18.2.5	能力成熟度集成(CMMI)	374
18.3	测试过程改进	375
18.4	测试成熟度模型	383
18.5	总结	389
	文献综述	390
	参考文献	390
	习题	391
术语表		392

第 1 章 基本概念和预备知识

Software is like entropy. It is difficult to grasp, weighs nothing, and obeys the second law of thermodynamics, i. e., it always increases. (软件像是熵, 很难把握且无重量, 服从热力学第二定律, 也就是说, 它总是会增加的。)

—Norman Ralph Augustine

1.1 质量革命

对于各种人工制造的手工艺品或称工件(artifact), 人们都会追求质量。当然, 质量的概念并不是伴随着各种软件系统的产生而出现的。实际上, 质量的概念如同人类大规模生产制品、大型物品的历史一样古老。在过去的几十年里, 伴随着 Internet 的爆炸式增长, 一种有关质量的变革已在全世界快速蔓延开来。全球竞争、外包、离岸外包、增加的客户期望, 已经将质量的概念提高到重要的地位。在紧张的工期里开发出高质量的产品, 对于一个公司成功屹立于新的全球经济之中是至关重要的。传统上, 提高质量的需求在产品开发周期的末期成为核心, 依靠于加强各种缺陷的发现与改正。与此相反, 增强质量的新方法包含了一个产品开发过程的各个方面——从需求分析到向客户的提交最终产品。开发过程中的每一步都必须做到尽可能的最高标准。一个有效的质量过程必须要注意^[1]:

- 更加专注于客户的需求
- 采取措施持续地提高质量
- 使度量过程与产品设计、开发融为一体
- 将质量概念推广到整个组织的最底层
- 建立一种对方法、过程高度关注的系统级的视图
- 通过持续的质量提高来消除浪费

由 William Edwards Deming、Joseph M. Juran 和 Kaoru Ishikawa 领导的一场关于质量的运动, 于 20 世纪 40 ~ 50 年代在日本开始。大约在 1947 年, W. Edwards Deming“访问了印度, 然后继续来到日本, 在那里他被邀请加入一个统计组织, 负责计划日本 1951 年的人口普查”^[2]。在他去日本访问的期间, Deming 邀请了一些统计专家参加一个晚餐会议, 并且强调他们是何等重要, 以及他们能为日本做什么^[3]。1950 年 3 月, 受日本科学家与工程师联盟(JUSE)管理董事 Kenichi Koyanagi 的邀请, Deming 回到了日本, 并为日本的研究者、生产人员、管理人员、工程师们讲授关于统计质量控制方法的课程。统计质量控制(SQC)是一门以度量和统计为基础的技术。决策的制定和计划的建立, 都是依照指标形式建立在收集和评估实际数据的基础上, 而不是建立在直觉和经验之上。SQC 方法采用了 7 种基本的质量管理工具: Pareto 分析法、因果图、流图、趋势图、柱状统计图、散点图、控制图^[2]。

1950 年 7 月, Deming 为日本的工程师和高管举办了 8 天基于 Shewhart 方法的统计质量控制的讲座^[4, 5]。在研讨会上他介绍了 PDCA (plan-do-check-act) 循环, 称之为 Shewhart 循环。Shewhart 循环(见图 1.1)例证了以下的活动序列: 设立目标, 为每个目标确定可测量的里程碑, 基于这些里程碑对过程进行评估。Deming 于 1950 年的演讲笔记形成了关于 SQC 方法系列讲座(JUSE