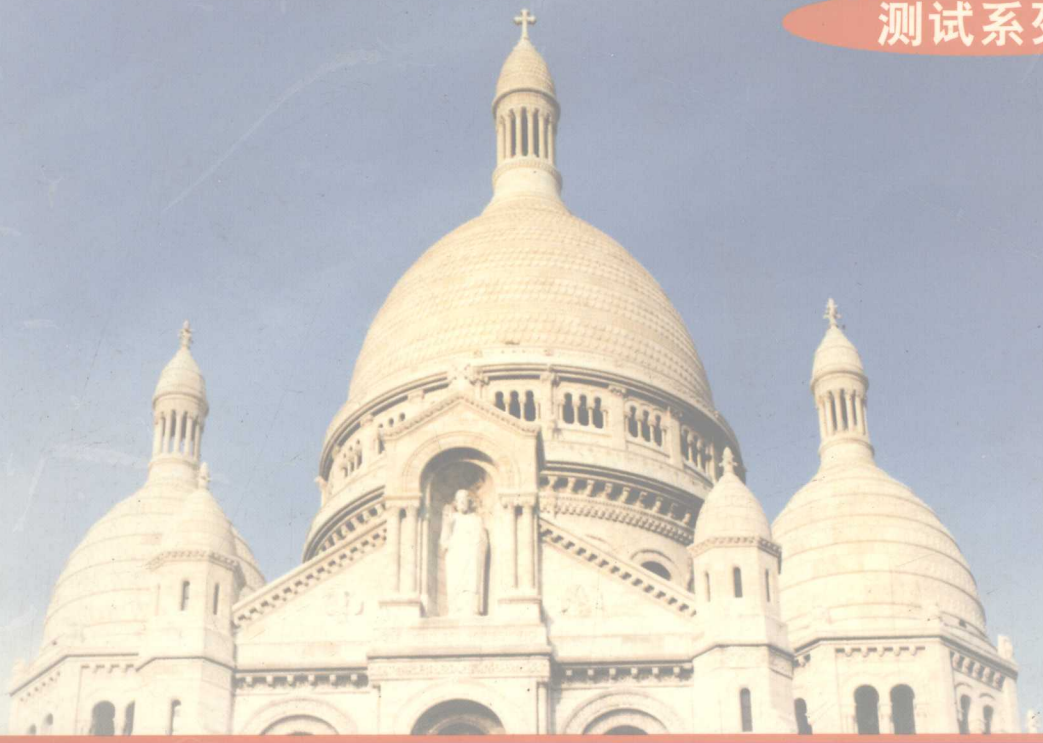
HZ BOOKS
软 件 工 程 技 术 丛 书

测试系列



计算机软件测试

（原书第2版）

Testing Computer Software

(Second Edition)

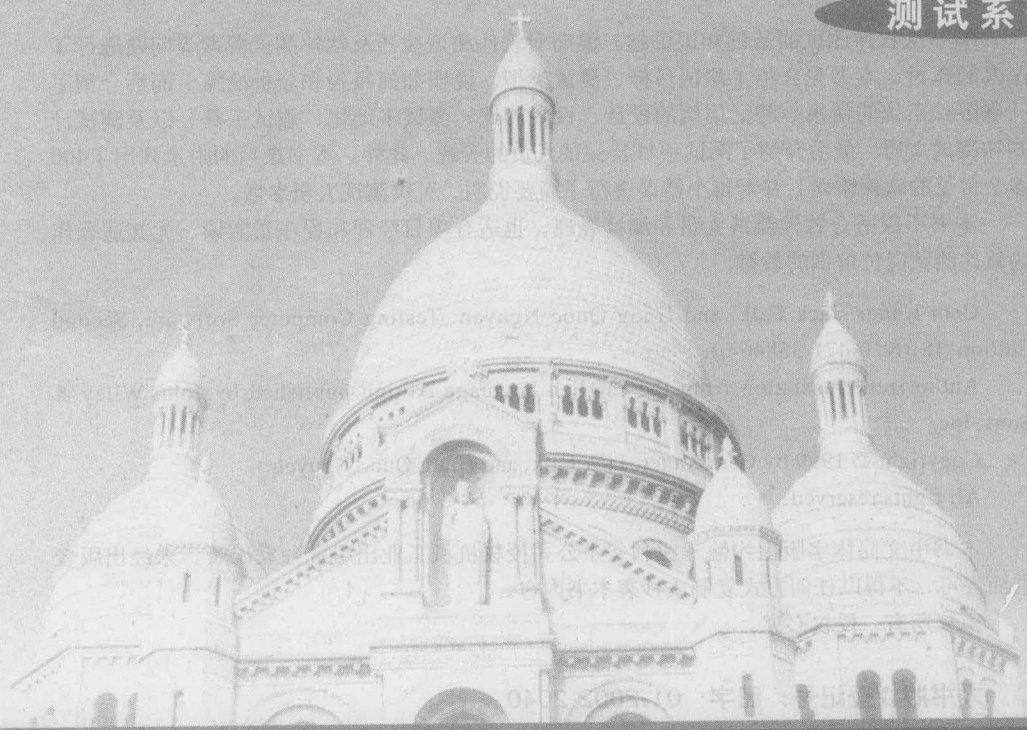
Cem Kaner
（美） Jack Falk 著
Hung Quoc Nguyen

王峰 陈杰 喻琳 译

机械工业出版社 China Machine Press
中信出版社 CITIC PUBLISHING HOUSE

软 件 工 程 技 术 丛 书

测试系列



计算机软件测试

（原书第2版）

Testing Computer Software

Second Edition

Cem Kaner
（美） Jack Falk 著
Hung Quoc Nguyen

王峰 陈杰 喻琳

机械工业出版社 China Machine Press
中信出版社 CITIC PUBLISHING HOUSE

本书从软件测试的基础知识讲起，继而对软件测试技巧及软件测试管理等问题进行了深入的探讨。本书先介绍了测试目标、测试类型，说明如何报告和分析故障；而后介绍了问题跟踪系统的使用、测试用例的设计、设备测试，测试本地化、测试工具，以及测试计划和测试文档；最后介绍了测试项目及测试人员的管理。此外，本书最后的附录列出了400多个常见的软件错误，并对每个错误进行了简要说明，可供测试人员参考。

本书不仅适合软件测试人员和测试经理，也适合项目经理和程序员阅读，尤其适合作为软件测试岗位培训的教材。

Cem Kaner, Jack Falk, and Hung Quoc Nguyen: Testing Computer Software, Second Edition (ISBN: 0-471-35846-0).

Authorized translation from the English language edition published by John Wiley & Sons, Inc.

本书版权登记号：图字：01-2002-2040

图书在版编目（CIP）数据

计算机软件测试（原书第2版）/（美）卡尼尔（Kaner, C.）等著；王峰等译. －北京：机械工业出版社，2004.5

（软件工程技术丛书　测试系列）

书名原文：Testing Computer Software, Second Edition

ISBN 7-111-14246-2

Ⅰ.计…　Ⅱ.①卡…②王…　Ⅲ.软件－测试　Ⅳ.TP311.5

中国版本图书馆CIP数据核字（2004）第027653号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：杨海玲

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2004年5月第1版第1次印刷

787mm × 1092mm 1/16 · 26.25 印张

印数：0 001 - 5 000 册

定价：39.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

译者序

从计算机诞生起，软件问题就伴随其左右。层出不穷的软件故障以及由此带来的尴尬局面给软件产业敲响了警钟。在大量现实的不良后果面前，痛定思痛，人们开始重新认识软件测试：软件测试与软件开发对软件质量具有同等重要的意义。

近年来，有不少好书讲解有高可靠性要求的软件产品的测试方法，细致而专业，介绍了如何进行完整的软件测试工作。它们通常是针对重要行业（如军事、医疗、金融等）的应用系统，这些系统事先都进行了详细的定义和设计，质量保证和测试活动的经费充足，软件测试人员能够彻底接触系统源代码。而本书侧重于不会造成人命关天后果的消费软件、普通商业软件、学术研究软件和个人软件等，它们不要求很高的可靠性。

本书作为面向学生的教材，主要是为了提供培训和工作指南而撰写。书中介绍了如何在通常的业务条件下测试消费软件和商业软件，以生动的语言和实例说明如何在有限的测试预算、紧迫的时间进度以及缺乏足够人手的条件下进行有效测试，保证产品具有令人满意的质量。书中介绍的测试不要求读者按照某种特定的“规则”或“规定”行事，它更强调在测试活动中每个人的力量及相互间的协同合作，认为软件质量的保证在于参与测试的每个人追求卓越品质的承诺、熟练使用工具的技巧以及协同工作的能力，而不是死抠“规定”。书中指出，开发阶段形成的一系列规定（如规格说明）不是一成不变的东西，会随着测试工作的开展，根据产品功能及其他品质的修正而做出相应改变。测试人员的工作就是要在产品发生后期变更的有限时间内尽可能平稳地处理变更状况，以最大的个人积极性来改进产品质量，而不是刻板地建立规章制度来抑制变更的产生。

本书选择了反映出读者兴趣及对读者有益的主题，摒弃某些学术性内容（如程序正确性证明），讨论在一般教科书中并未强调的内容。例如，书中讨论了人际关系和企业文化，提出了一些建议来让测试人员避免陷入企业内部的政治漩涡中，从而提高工作效率；讨论了项目管理问题，指出软件测试估算、计划和时间安排的重要性，提出优先级和效率是做出权衡的重点考虑对象；探究了法律问题，指出测试人员应如何避免产生各类法律纠纷以便更好地完成工作。

本书的很多修订为在校学生而做，希望本书能够填补大学教育中有关测试知识课程的空白，使更多具备相当计算机软件测试知识的学生能够投入到这一领域中来。然而在此我们要指出“师傅领进门，修行在个人”，我们希望本书的读者不仅仅满足于本书的内容，要更系统地学习相关的大学计算机课程，才能拓展思维，更好地领会和实践。

本书以一个样例测试系列开始，引出测试活动的基础知识，继而对测试技巧、测试管理等问题进行了探讨。全书分为三部分。第一部分是基本原则：讲解测试目标和局限、测试类型，简要介绍常见的软件问题，并说明如何报告和分析故障，这是每位读者都应仔细阅读的部分。第二部分是特殊的测试技巧：问题跟踪系统的使用，测试用例的设计，设备测试，本地化测试，用户手册测试，有用的测试工具，以及测试过程中产生的测试计划和测试文档。读者可以根据

自身知识水平选择相应章节阅读，或者按任意顺序阅读。最后一部分是测试项目和测试人员的管理：测试开发进程时间基线、法律问题和管理问题。这部分主要针对高级测试人员和测试经理。另外，本书在附录中分门别类地列出了400多个常见的软件错误，并对每个错误进行了简要说明，以备测试人员作为参考来查对被测软件的错误，还可以根据实际情况增删故障现象以形成自己的故障数据库。

本书由王峰、陈杰、喻琳翻译，全书由王峰统稿。鉴于译者水平有限，在翻译过程中难免出现失误和不足，请广大读者不吝赐教。

前　言

本书介绍了如何在通常的业务条件下测试消费软件和商业软件，实在而且实用。我们曾经为硅谷中快速成长的著名软件公司测试软件，管理其测试人员。我们撰写本书的目的在于为员工提供培训和工作指南。

已经有很多很好的书在讲授有高可靠性要求的软件产品的测试方法。人命关天的和金融领域的应用程序，事先都得到了详细的定义和设计，质量保证和测试活动的经费充足，软件测试人员能够完全接触系统源代码，并有充分的时间阅读它们。在这样的前提下，这些书籍介绍了如何进行完整的软件测试工作。

计算行业的大多数人，包括个人计算机业的绝大部分人，都在努力开发有用且可靠的软件产品，但这些软件并不要求很高的可靠性。就像买车你可以选性能良好的经济型轿车或中档轿车一样，大多数的程序不必成为劳斯莱斯（而且大多数人是买不起劳斯莱斯的）。消费软件、普通商务软件、学术研究软件和个人软件的开发人员的测试预算相当有限，时间紧迫，人手短缺，然而很多程序却具有令人满意的质量。他们是如何做到的呢？本书介绍了其中的测试工作部分。

书不能解决全部问题

有些书认为，如果一个软件项目没有得到“合理”的控制，书面规格说明始终没有完成且不是最新的，代码没有依据所谓流行的方法进行适当组织的话，那么必须要努力做到。这些书讲到的测试是建立在每个人都依据“规定”行事的基础之上的。

> 本书介绍的测试，假定的前提是你的同伴现在没有、将来也不会并且也没必要遵循这些规定。

消费软件项目通常的特点是，预算非常有限、人手非常少、交付时间非常紧迫而且不可能推迟、开发人员之间拥有共同的认识和承诺。

大型软件产品的质量掌握在那些负责设计、编程、测试、撰写文档的个人手上，每个人都很重要。无论是标准、规格说明、指导/协调委员会和变更控制都不能保证质量，软件公司也没有指望它们起到这样的作用。能够确保软件质量的，是个人追求卓越的承诺、熟练使用工具的技巧和协同工作的能力，而不是这些规定。

开发小组对所要创建的东西有了认识，对尽可能实现梦想达成了承诺，并且意识到不得不通过试验和错误来充实这些细节。当产品某一部分的所有细节工作都完成时，他们会形成一个只有一两个人能完全理解的工作说明。这个说明就是所谓的“规格说明”。“规格说明”并不是刻在石头上一成不变的，随后会被评审和修正，以便与系统其他部分保持一致。大多数的修正是根据软件测试人员的建议进行的。

在现实世界中，产品变更一般发生在开发阶段的后期。当你正开发一个公开发售的软件产品时，你的客户或潜在客户不一定会同意你的设计。如果竞争对手开发出更具吸引力的产品，你要马上作出反应。

由于没有时间实现，预想的功能往往被取消掉。代码重写也被搁置一边，即使是需要完成这些代码重写工作把勉勉强强提交的第一版本改得稍微专业一些。认真负责的程序员也许会自觉地，甚至是“偷偷地”完成这项工作。在项目的后期，他可能在没有通知其他任何人的情况下对软件包做了重要修改。这些努力是在每周40小时工作时间之外完成的，可能会极大地改进软件质量，但也可能使软件质量变得极其糟糕。不管结果如何，软件质量的改进是通过个人的积极性来实现的。

后期的变更是不可避免的。我们的目标是尽可能平稳地来处理这些必要的变更。我们不打算建立什么规章制度来抑制变更的产生。作为测试人员，你如果处理不好变更，就会有麻烦。解决之道不在于单纯抱怨或是竭力阻止它们。

本书的读者对象

本书的读者对象是从事软件测试且常常测试他人代码的人员。我们认为所选主题和讨论水平反映了读者感兴趣或对读者有益处的内容。因此我们摒弃某些学术性内容（如程序正确性证明），讨论通常在测试教科书中并未强调的内容。例如，我们经常谈论到人际关系和企业文化问题。连最初级的软件测试人员也必须评判他人的工作——这就是测试工作的根本。在将他们的判断与别人进行交流的过程中，测试人员容易受到指责（有时也是“罪有应得”）。他们工作的重要程度往往与其地位不相称，从而导致他们容易成为人际矛盾的靶子或替罪羊。我们没有办法解决其地位问题，但我们确实提出了一些建议来提高测试效率和避免某些陷阱。

我们也讨论项目管理问题。软件测试的估算、计划和时间安排很棘手，因为软件测试没有真正的终点，测试总是没完没了，如果不做就会有更多的风险。每个测试人员都必须明白这种平衡关系。他们需要安排时间来充分考虑这些因素。例如，某个测试人员可能为了更好地完成其他任务而推迟了重要的测试，但是，后来发现由于时间已用完，再也无法进行这些测试了。本打算将工作做得更好，却弄巧成拙。这种情况很常见，而且非常令人沮丧。因此，优先级和效率是本书主要关注的问题。

测试人员还要处理设计错误。按照糟糕的规格说明实现的程序，其质量必定是糟糕的。大多数有关软件可靠性和测试的书籍没有涉及用户界面，而将其作为人为因素分析员的工作范围。我们不同意这种观点。（我们之中就有一位人为因素分析专家，他也不赞同。）系统的可靠性取决于其各部分是如何协同工作的，包括使用它的人员在内。

作为测试人员，你的任务是发现并指出产品中存在的问题，为改进产品质量服务。无论问题的源头出在何处，关于人－机系统可靠性差的报告都是适当和重要的。**你是少数几个能在产品发布前详细检验整个产品的人之一**。还有哪些地方能产生这种反馈呢？

我们对用户界面问题的讨论并不能让你成为专家，你仍然会遗漏很多重要问题。你的一些设计建议可能是笨拙的，但你依然应该递交这些报告，它们很重要。其中一些报告将导致产品改进，除此之外，别无他法。

人命关天的软件

在某些条件下，编程人员或他们的经理如果违背标准化的方法是绝对不能接受的。核反应堆的控制程序必须文档齐备，并得到彻底地详细说明，仅在精确计算之后方可进行变更。如果发生了失误，测试文档将是非常重要的法律证据。像这样的项目，负担得起大批质量保证人员和巨大的测试预算。对于这种项目的质量保证人员，测试方面的其他书籍比本书更适合他们。

然而，即使在这些项目中，最后常会有一个验收测试过程。这些测试人员不属于开发队伍的核心，他们得到的预算非常少，他们的时间限期简直是不可能的。整个质量保证机构看不起他们的工作。他们接触产品的时间有限，无法采取“正确”的方式来进行“真正”的测试。如果你是在这种环境下工作，我们认为本书比其他经典教科书更适合你。

这是一本教科书吗

我们曾经面试并雇用了很多测试人员。但我们还没有见到在大学里学到了足够测试知识的计算机专业的毕业生。

1991年，美国计算机协会（Association for Computing Machinery，ACM）与IEEE计算机学会出版了《计算课程1991》（Computing Curricula 1991），在之后的十年中深刻影响了大学计算机学位课程的设计。但是，它给软件测试技术的必修时间少得可怜——四年学习期间只有几个小时而已。这一课程指导建议设立一个选修课程“高级软件工程”，其中包括关于软件测试的主题，但它没有提及是否开设一门以测试为主要内容的课程。

往后十年，我们也不能期望计算机专业的毕业生在大学中能学到足够的软件测试知识。

我们不知道为什么学校不去填补这个空白。我们认为，学过几门测试课程、同时又辅修了一些入门的编程和项目管理课程的理工科毕业生，应该能够很容易找到工作。测试人员的起薪也比较高。我们认为，大学教育应为这样的工作培养人才。

本书中所做的很多修订是为在校学生所做的，他们可能在头脑中从未形成对软件测试实验室的认识。对于那些尚未设置通用商业软件测试课程的院校，我们希望本书能对他们设置课程提供帮助。

如果你拥有理工科学位，我提醒你可以考虑从事测试工作。具备理工科教育背景并学过一些测试课程便能入门，但仅满足于此还不够。我们强烈建议你在适应了基本工作之后，最好在业余时间立即去修相关课程。

关于本书结构和布局的说明

对本书结构的一些说明

这是一本培训教材。有的读者可能是初次接触测试这个领域，而有的读者可能已是有经验的测试人员，他们已经通过岗位培训掌握了工作知识，这是他们的第一本测试教材。本书的结构正是为服务于这些读者而设的。

本书分层次来介绍有关知识。同样的问题在本书的不同章节以不同难度、广度和深度进行阐释。例如，第1章和第7章都讨论了边界条件。第3章和第13章都在讨论项目时间基线，以及如何按项目时间基线来安排测试任务。我们不在一个地方讨论某个主题的方方面面，而是随着读者了解了更多的背景和来龙去脉，逐步展开讨论该领域核心问题的更多内容。

第一部分——基础知识

前五章从测试新手的角度介绍了测试领域的基础知识，每个人都应阅读这几章。

给测试新手的建议

如果你详细阅读了第1、2、4、5章，并浏览了第3章，你就会非常喜欢这本书。然后，如有可能，花几周时间实际做些测试，报告一些缺陷，并收集一些反馈信息，之后再继续读第6章和以后的内容。

给教师的建议

如果学生们没有工作经验，在继续第6章之前，布置一个小型的测试项目（大约10~20小时的工作）。例如，在即将发布的软件中寻找缺陷（这类软件产品中存在很多缺陷）。不要强行要求对缺陷进行分析，只要记录这些缺陷和设计建议，并模拟写出其中的一些反馈信息。在练习的最后，重温一下测试涉及的范围，并指出明显存在缺陷的地方，或未做的重要测试。

第二部分——专门的测试技巧

第5章到12章重点讨论专门的测试技巧或问题。你可以独立地阅读任意一章或按任意顺序阅读（但在第6章前一定要先读第5章）。这些章对所有测试人员都很有益。我们写作这几章时设定的难度要稍高一些，目标读者是写过测试计划并领导过小规模测试组，或正在接受培训以便担当该职位的人。然而，阅读过第1章到第5章的测试新手也应该能通读这些内容。

给测试新手的建议

第6章讨论如何处理由产品的开发和营销团队提交的问题报告，以及为有利于问题处理而进行的问题报告数据库设计。对于那些总是经历挫折，很难追踪和描述严重错误，只能放弃、忽略、误解或搁置问题报告的读者来说，他们会对第6章特别感兴趣。这也正是他们积极去了解问题报告是如何处理，以及怎样才能做得更好的时机。如果你觉得第6章很枯燥，可以简单地看看

前面的几节，跳过“数据库的技术细节”之后的内容。

第8章是关于打印机测试的。有些程序要处理很多打印任务，我们有员工专门设计打印机测试，本章是根据我们要求其掌握的详细程度编写的。如果内容对你过细，或你的工作不涉及太多打印，仅了解全章的整体思路即可。认真体会设备无关错误、打印机（或调制解调器、终端或视频卡等）类别相关的错误、驱动器相关错误、设备相关错误之间的差异。如果你能领会到我们为何按顺序测试这些错误类型，你就掌握了该章的精髓。

给教师的建议

作为测试人员的面试官，我们很高兴看到我们面试的人做过的很多工作（如编写的程序、写过的测试计划或测试报告）。商业保密守则禁止大多数测试人员向外界展示他们以前的工作，所以好的工作范例在毕业后的几年中都非常有用。以下是学生为拿学分可能做的一些项目，可保留在他/她的档案中。

- 为商业软件中使用的高度结构化的数据输入界面建立边界表（见第7章和第12章）。很多支票打印程序、地址簿、联系人管理程序都是实用且便宜的数据库应用程序范例。对单个的边界条件究竟需要多少测试才够？对几个边界条件的组合又大概需要多少测试才合适？首先找出必须测试的特别组合。对剩下的部分，编写一个采用随机数产生器（见第7章）的简短程序来生成对边界组合的测试用例。与手工选择边界条件的每个组合相比，这种方法是更好，还是更糟？
- 明白软件是如何与其他设备打交道的学生，可以修改第8章的内容来作为调制解调器、鼠标、视频卡、声卡或其他设备的测试程序。项目应包含能在特定的、现有商业软件中运行的测试范例。
- 按照软件手册对现有商业软件进行测试（见第10章）。项目报告就是注解了的手册，以及对软件与手册之间每个差异的缺陷报告，要明显地反映出在软件中而不是手册中存在一个错误。
- 使用自动测试工具对现有商业软件进行测试（见第11章）。不同类型的测试用例的创建时间是多长？与手工测试所需的时间相比较情况如何？在测试用例不失效的情况下，程序能够做多大程度的变更？基于这些情况，学生如何使用测试工具——在何种环境下、做何种测试？

第三部分——管理测试项目和小组

最后的第12章到第15章是针对高级测试人员和测试经理的。第12章和第13章分别从测试计划范围（见第12章）或项目范围（见第13章）的角度来组织上述材料。第14章考虑的是不当测试的法律后果，第15章讨论对测试小组的管理。

对本书布局的一些说明

本书的结构是按层次组织的。我们作了一些努力，尽量使这种结构清晰明显。例如，不同

层次的标题采用不同字体表示：

主标题

二级标题

三级标题

四级标题

我们也采用不同字体和特殊字符来标注一些信息：

- Courier字体：表示从计算机键盘输入的文本，或者，在用于避免与周围文字发生混淆的内容时，表示计算机屏幕显示的文本或计算机打印的数据库表格内容。
- Courier大写字体：表示数据库字段名或程序变量名。
- 楷体字体：表示术语的第一次重要使用，周围段落定义了该术语，或者表示强调的要点概述。
- <Key>：表示计算机键盘上按键的名字，如<Enter>和<Ctrl>。

最后，当引用本书其他章节时，使用节号，如“12.1.1节”。

致谢

本书的写作开始于1983年。这些年来，许多人就本书给予了我们帮助，对原稿和第1版（Kaner, 1988）提出了批评意见。

我们特别感谢下列人士（按姓名字母顺序）：Elaine Andersson教授、Boris Beizer博士、Jim Brooks、Randy Delucchi，Mel Doweary，David Farmer、Larry Jones教授、Sharon Hafner、Mahmood Hahn、Ginny kaner、Sam kaner博士、John Lavelle、Larry Malcus博士、Ted Matsumara、Don Maxwell博士、Bruce Miller、Derick Miller、Rachel Miller、Peter Morse、Jane Stepak和Emmanuel Uren。

当然，作者对书中的所有错误负责。

作者简介

Cem Kaner

技术及软件开发管理顾问，并在当地大学及几家软件公司中讲授软件测试课程。他还是律师，通常为个人开发者、小型开发服务公司及客户工作。他创建并主持着洛斯阿尔托斯软件测试研讨会（Los Altos Workshops on Software Testing）。Kaner在1976年开始使用计算机，当时他是一名人类实验心理学的研究生。1983年，他前往硅谷，作过程序员、人为因素分析师、用户界面设计人员、软件销售人员、团队开发咨询公司合伙人、技术撰稿人、软件测试技术小组负责人、软件测试经理、技术发布经理、软件开发经理，以及文档编制和软件测试主管。他还曾作为代理地方检察官以及作为加利福利亚地区消费者事务部门的调查员/调解人提供公益服务。他积极参与到影响软件质量法规的立法工作中，并且是《Bad Software: What to Do When Software Fails》的资深作者（Wiley，1998）。Kaner拥有数学学士、哲学学士、法学博士以及心理学博士等多个学位，而且他通过了美国质量协会的质量工程认证。读者可以通过kaner@kaner.com与他联系，在www.kaner.com查阅其有关软件测试的著作，在www.badsoftware.com上查阅其有关软件消费者保护的著作。

Jack Falk

软件质量管理及软件工程管理顾问。曾担任质量保证和工程服务主管；管理过多媒体软件产品的开发、软件开发运作及工程支持服务；还管理过手持设备操作系统软件、软件开发工具包（SDK）、娱乐、图形及财务应用软件的测试工作。另外，他还管理过若干重要零售商的配送和运作业务（在集团经理或主管级别）。Jack通过了美国质量协会的软件质量工程认证，在旧金山城市学院获得商业AA学位，并完成了大量的商业及软件开发课程。他是2级证券分析师，并持有不动产评估师的执照。他是圣克拉拉地区（Santa Clara Valley）软件质量协会副主席，还是洛斯阿尔托斯软件测试研讨会的积极参与者。读者可以通过jfalk@asqnet.org与其联系。

Hung Q. Nguyen

*Soft*Gear技术公司（一家硅谷公司，其宗旨是帮助软件开发组织在资源缺乏、时间紧迫的夹缝中尽量交付高质量的产品。）的创始人、总裁和CEO。这家公司创建于1994年，在完成若干外包测试工程及测试项目、软件测试支持产品以及一系列实用的软件测试培训课程之后，成功达到其预期目标。Hung还开发了培训材料，并在大学和众多知名的美国及国际软件公司中向大众讲授软件测试课程。他曾在计算机软件及硬件行业供职，担任过工程、质量保证、测试、产品开发以及信息技术的管理职位，并作为测试人员和程序员做出过重大贡献。Hung拥有考格斯威尔工艺学院的质量保证科学学士学位，是美国质量协会（ASQ）认证的质量工程师、美国质量协会高级会员及旧金山分会认证主任。Hung与妻子Heather及两个孩子（Wendy和Denny）居住在加利福利亚的福斯特城（Foster City）。读者可以通过hungn@softgeartech.com与其联系，或者在www.softgeartech.com上了解*Soft*Gear技术公司的更多信息及Hung的工作。

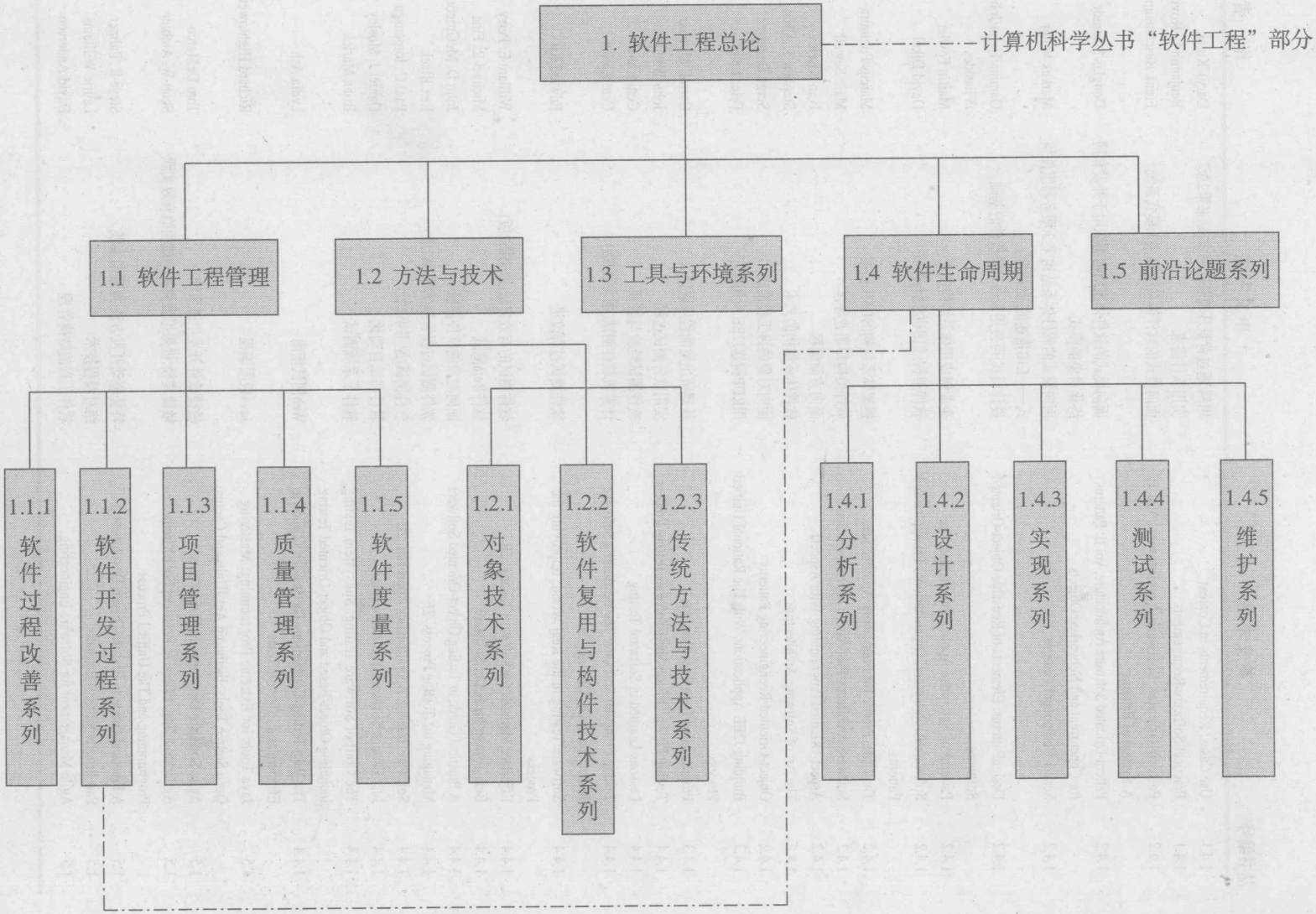
软件工程技术丛书书目

丛书编号	英文书名	中文书名	作　者
1	Object-Oriented and Classical Software Engineering, 5E	面向对象与传统软件工程(原书第5版)	Stephen R. Schach
1	Object-Oriented Software Engineering	面向对象软件工程	Timothy C. Lethbridge
1	Software Engineering: A Practitioner's Approach, 5E	软件工程:实践者的研究方法(原书第5版)	Roger S. Pressman
1	Software Engineering, 6E	软件工程(原书第6版)	Ian Sommerville
1	Software Engineering with Java	软件工程:Java语言实现	Stephen R. Schach
1	Project-Based Software Engineering:An Object-Oriented Approach	基于项目的软件工程:面向对象研究方法	Evelyn Stiller
1	Software Engineering Economics	软件工程经济学	Barry W. Boehm
1	Software Cost Estimation With Cocoma II	用Cocoma II模型进行软件成本估算	Barry W. Boehm
1	Object-Oriented Software Construction, 2E	面向对象的软件结构(原书第2版)	Bertrand Meyer
1	Software for Use: A Practical Guide to The Models and Methods of Usage-Centered Design	面向使用的软件设计	Larry L. Constantine
1.1.1	Software Process Improvement: Practical Guidelines for Business Success	软件过程改进	Sami Zahran
1.1.1	Making Process Improvement Work	软件过程改进简明实践	Neil S. Potter
1.1.2	The Road to the Unified Software Development Process	统一软件开发过程之路	Ivar Jacobson
1.1.2	The Unified Software Development Process	统一软件开发过程	Jacobson/Booch/Rumbaugh
1.1.2	The Rational Unified Process: An Introduction , 2E	Rational统一过程引论(原书第2版)	Philippe Kruchten
1.1.2	UML and The Unified Process: Practical Object-Oriented Analysis & Design	UML和统一过程:实用面向对象的分析和设计	Jim Arlow
1.1.2	The Unified Process Inception Phase	统一过程初始阶段	Scott Ambler
1.1.2	The Unified Process Elaboration Phase	统一过程细化阶段	Scott Ambler
1.1.2	The Unified Process Construction Phase	统一过程构造阶段	Scott Ambler
1.1.2	The Unified Process Transition & Production Phase	统一过程移交和生产阶段	Scott Ambler
1.1.3	Managing Global Software Projects	全球化软件项目管理	Gopalaswamy Ramesh
1.1.3	Software Project Management: A Unified Framework	软件项目管理:一个统一的框架	Walker Royce
1.1.3	How to Run Successful Projects III: The Silver Bullet	成功的软件项目管理:银弹方案(原书第3版)	Fergus O'Connell
1.1.3	Successful Software Development, 2E	成功的软件开发(原书第2版)	Scott E. Donaldson
1.1.3	Six Sigma Software Development	六西格码软件开发	Christine B. Tayntor
1.1.3	IT Project Management: On Track from Start to Finish	实用IT项目管理:从开始到结束的历程	Joseph Phillips
1.1.3	Successful IT Project Delivery: Learning the lessons of Project Failure	IT项目成功交付的秘诀	David Yardley
1.1.3	Software Project Management, 3E	软件项目管理(原书第3版)	Bob Hughes
1.1.3	Architecture-Centric Software Project Management	软件项目管理实用指南:以体系结构为中心	Daniel J. Paulish
1.1.3	Mentoring Object Technology Projects	对象技术项目管理	Richard T. Dué
1.1.3	Virtual Project Management	虚拟项目管理	Paul E. Mcmahon
1.1.3	AntiPatterns and Patterns in Software Configuration Management	软件配置管理中的模式与反模式	William J. Brown
1.1.4	Handbook of Software Quality Assurance, 3E	软件质量保证(原书第3版)	Gordon G. Schulmeyer
1.1.4	Software Reliability Engineering	软件可靠性工程	John Musa
1.1.4	Implementing ISO 9001:2000 The Journey from Conformance to Performance	2000版ISO 9001标准实施指南:从符合性到业绩改进	Tom Taormina
1.1.4	CMMI Distilled: A Practical Introduction to Integrated Process Improvement	CMMI精粹:集成化过程改进实用导论	Dennis M. Ahern

丛书编号	英文书名	中文书名	作　者
1.1.4	CMM Implementation Guide	CMM实施与软件过程改进	Kim Caputo
1.1.4	Implementing the Capability Maturity Model	CMM实施指南	James R.Persse
1.1.4	Object-Oriented Defect Management of Software	面向对象的软件缺限管理	Houman Younessi
1.1.4	Metrics and Models in Software Quality Engineering	软件质量工程:度量与模型	Stephen H. Kan
1.1.4	Performance Solutions	软件性能工程	Connie U. Smith
1.1.4	Peer Reviews in Software: A Practical Guide	软件同级评审	Karl E. Wiegers
1.1.5	Practical Software Measurement	实用软件度量	John McGarry
1.1.5	Software Metrics: A Rigorous and Practical Approach, 2E	软件度量(原书第2版)	Norman E. Fenton
1.1.5	Software Assessments, Benchmarks, and Best Practices	软件评估、基准测试与最佳实践	Capers Jones
1.2.1	The Object Primer: The Application Developer's Guide to Object Orientation and the UML, 2E	面向对象软件开发教程(原书第2版)	Scott W. Ambler
1.2.1	UML and C++: A Practical Guide to Object-Oriented Development, 2E	C++面向对象开发(原书第2版)	Richard C.Lee
1.2.1	Object-Oriented Methods: Principles & Practices, 3E	面向对象方法:原理与实践(原书第3版)	Ian Graham
1.2.1	Principles of Object-Oriented Software Development, 2E	面向对象软件开发原理(原书第2版)	Anton Eliëns
1.2.1	Object Solutions: Managing the Object-Oriented Project	面向对象项目的解决方案	Grady Booch
1.2.1	An Introduction To Object-Oriented Programming, 3E	面向对象编程导论(原书第3版)	Timothy Budd
1.2.1	The Unified Modeling Language User Guide	UML用户指南	Booch/Rumbaugh/Jacobson
1.2.1	The Unified Modeling Language Reference Manual	UML参考手册	Rumbaugh/Jacobson/Booch
1.2.1	Applying UML and patterns: An Introduction to Object-Oriented Analysis and Design, 1E	UML和模式应用(原书第1版)	Craig Larman
1.2.1	Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process , 2E	UML与模式应用(原书第2版)	Craig Larman
1.2.1	Object-Oriented Analysis and Design with Applications, 2E	面向对象分析与设计(原书第2版)	Grady Booch
1.2.1	Business Modeling with UML: Business Patterns at work	UML业务建摸	Hans-Erik Eriksson
1.2.2	Software Reuse: Architecture, Process and Organization for Business Success	软件复用: 结构、过程和组成	Ivar Jacobson
1.2.2	Software Reuse Techniques: Adding Reuse to the Systems Development Process	软件复用技术: 在系统开发过程中考虑复用	Carma McClure
1.2.2	Practical Software Reuse: Strategies for Introducing Reuse Concepts in Your Organization	软件复用实践	Donald J. Reifer
1.2.2	Large-Scale Component-Based Development	大规模基于构件的软件开发	Alan W. Brown
1.2.2	Component-based Product Line Engineering with UML	基于构件的产品线工程:UML方法	Colin Atkinson
1.2.2	Business Component Factory	商业构件工厂	Peter Herzum
1.4.1	Object-Oriented Analysis & Design	面向对象的分析与设计	Andrew Haigh
1.4.1	Analysis Patterns: Reusable Object Models	分析模式: 可复用的对象模型	Martin Fowler
1.4.1	Requirements Analysis and System Design: Developing Information Systems with UML	需求分析与系统设计	Leeszek A. Maciaszek
1.4.1	Systems Analysis and Design in a Changing World	系统分析与设计	John W. Satzinger
1.4.1	Advanced Use Case Modeling, Vol I: Software Systems	高级用例建模 卷1: 软件系统	Frank Armour
1.4.1	Requirements Engineering: A Good Practice Guide	需求工程	Ian Sommerville
1.4.1	Software Requirements and Estimation	软件需求与估算	Swapna Kishore
1.4.1	Effective Requirements Practices	有效需求实践	Ralph R.Young
1.4.1	Applying Use Cases: A Practical Guide, 2E	用例分析技术(原书第2版)	Geri Schneider
1.4.1	Managing Software Requirements	软件需求管理:统一方法	Dean Leffingwell
1.4.1	Writing Effective Use Cases	编写有效用例	Alistair Cockburn
1.4.1	Problem Frames Analyzing and Structuring Software Development Problems	Problem Frames Analyzing and Structuring Software Development Problems	Michael Jackson

丛书编号	英文书名	中文书名	作　者
1.4.1	Use Cases: Requirements in Context	用例:通过背景环境获取需求(原书第2版)	Dary1 Kulak
1.4.1	Practical Software Requirements	实用软件需求	Benjamin L. Kovitz
1.4.2	Pattern-Oriented Software Architecture, Vol I: A System of Patterns	面向模式的软件体系结构 卷1:模式系统	Frank Buschmann
1.4.2	Pattern-Oriented Software Architecture, Vol II: Patterns for Concurrent and Networked Objects	面向模式的软件体系结构 卷2:用于并发和网络化对象的模式	Douglas Schmidt
1.4.2	Server Component Patterns	面向模式的软件体系结构 卷3:服务器组件模式——EJB描述的组件结构	Markus Volter
1.4.2	DesignPatterns: Elements of Reusable Object-Oriented Software	设计模式:可复用面向对象软件的基础	Gamma/Helm/Johnson /Vlissides
1.4.2	Patterns of Enterprise Application Architecture	企业级应用体系结构模式	Martin Fowler
1.4.2	Software Architecture: Organizational Principles and Patterns	软件架构: 组织原则与模式	David Dikel
1.4.2	The UML Profile for Framework Architectures	框架体系结构的UML档案	Marcus Fontoura
1.4.2	Software Architect's Profession: An Introduction	软件架构师职业导读	Marc Sewell
1.4.2	Aspect-Oriented Programming With AspectJ	面向方面编程	Ivan Kiselev
1.4.2	The Art of Software Architecture	软件体系结构的艺术	Stephen T. Albin
1.4.2	Object-Oriented Reengineering Patterns	面向对象的再工程模式	Serge Demeyer
1.4.3	Building J2EE Applications With The Rational Unified Process	用RUP构建J2EE 应用程序	Peter Eeles
1.4.3	Programming from Specifications	从规范出发的程序设计	Carroll Morgan
1.4.4	Testing IT: An Off-the-Shelf SoftwareTesting Process	实用软件测试过程	John Watkins
1.4.4	Lessons Learned in Software Testing	软件测试经验与教训	Cem Kaner
1.4.4	Testing Computer Software: The Bestselling Software Testing Book Of All Time, 2E	计算机软件测试(原书第2版)	Cem Kaner
1.4.4	Software Testing in the Real World: Improving the Process	软件测试过程改进	Edward Kit
1.4.4	Effective Methods for Software Testing, 2E	软件测试的有效方法(原书第2版)	William E. Perry
1.4.4	Beta Testing for Better Software	软件Beta测试	Michael R. Fine
1.4.4	A Practical Guide to Testing Object-Oriented Software	面向对象的软件测试	John D. McGregor
1.4.4	Managing the Testing Process, 2E	软件测试过程管理(原书第2版)	Rex Black
1.4.4	Software Testing: A Craftsman's Approach, 2E	软件测试(原书第2版)	Paul C. Jorgensen
1.4.4	Just Enough Software Test Automation	软件测试自动化	Daniel J. Mosley
1.4.4	The Craft of Software Testing: Subsystem Testing, Including Object-Based and Object-Oriented Testing	软件子系统测试	Brian Marick
1.4.4	The Web Testing Companion: The Insider's Guide to Efficient and Effective Tests	Web测试指南	Lydia Ash
1.5	Java Tools for Extreme Programming: Mastering Open Source Tools, Including Ant,JUnit, and Cactus	Java极限编程	Richard Hightower
1.5	Agile Software Development Ecosystems	敏捷软件开发生态系统	Tom DeMarco
1.5	Agile Modeling: Effective Practices For eXtreme Programming and The Unified Process	敏捷建模:极限编程和统一过程的有效实践	Scott W. Ambler
1.5	A Practical Guide to Feature-Driven Development	特征驱动开发方法:原理与实践	Steve R. Palmer
1.5	Pair Programming Illuminated	结对编程技术	Laurie Williams
1.5	Agile Management for Software Engineering	软件工程的敏捷管理	David Anderson

软件工程技术丛书结构图



目　录

译者序

前言

关于本书结构和布局的说明

作者简介

第一部分　基础知识

第1章　一个样例测试系列 ……3

1.1　第一个测试周期 ……3

1.1.1　第1步：从一个显而易见的简单测试开始 ……3

1.1.2　第一次测试产生的问题报告 ……4

1.1.3　第2步：对还需要测试什么做一些记录 ……4

1.1.4　寻找边界条件 ……6

1.1.5　第3步：检查有效用例并观察发生了什么 ……7

1.1.6　第4步：做一些“快速的”测试 ……7

1.1.7　第5步：总结对程序及其问题的认识 ……9

1.1.8　第一个测试周期的总结 ……12

1.2　第二个测试周期 ……12

1.2.1　第1步：在进行任何测试之前应仔细评审对问题报告的反馈，以确定哪些需求必须满足，哪些无须满足 ……13

1.2.2　第2步：评审对不进行改正的问题的意见，它们可能暗示着进行进一步的测试 ……13

1.2.3　第3步：找出上次的记录，补充新记录，然后开始测试 ……14

1.3　后续测试周期中可能会发生的事情 ……15

第2章　测试的目标和局限 ……17

2.1　不可能完全测试一个程序 ……18

2.1.1　不可能测试到程序对任何可能输入的响应 ……18

2.1.2　不可能测试到程序每一条可能的执行路径 ……20

2.1.3　无法找出所有的设计错误 ……22

2.1.4　不能采用逻辑来证明程序的正确性 ……22

2.2　测试人员的目标是验证程序吗 ……22

2.2.1　无法验证程序运行正确 ……22

2.2.2　程序不能正确地运行 ……23

2.2.3　既然程序不能正确地工作，那么测试是不是个失败呢 ……23

2.2.4　测试人员不应该试图验证一个程序运行正确 ……23

2.3　那么，为什么要进行测试呢 ……24

2.3.1　测试一个程序的目的是为了发现它的问题 ……24

2.3.2　发现问题的目的是为了改正问题 ……25

第3章　测试的类型及其在软件开发过程中的地位 ……26

3.1　软件开发阶段综述 ……29

3.2　规划阶段 ……30

3.2.1　目标阐述 ……30

3.2.2　需求分析 ……31

3.2.3　功能定义 ……31

3.3　规划阶段进行的测试 ……31

3.3.1　产品对照评价 ……32

3.3.2　重点问题小组 ……32

3.3.3　任务分析 ……33

3.4　设计阶段 ……33

3.4.1　外部设计 ……33

3.4.2　内部设计 ……34

3.4.3　原型开发 ……35