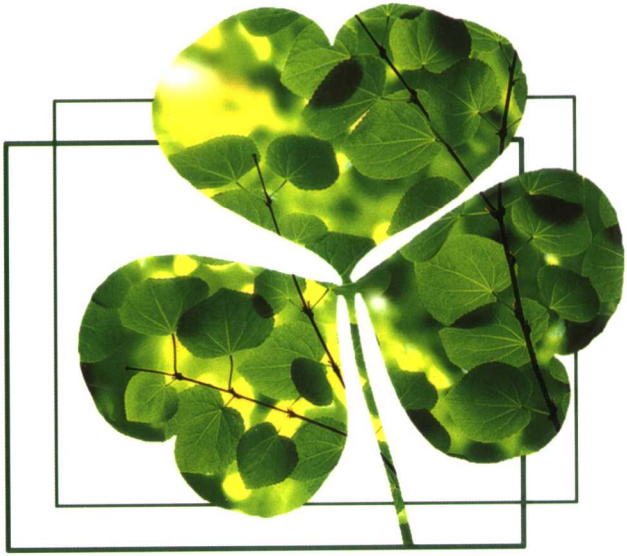
Broadview®
www.broadview.com.cn



测试实践丛书

软件测试实践

周伟明 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

TP311.5/234
2008

测试实践丛书

软件测试实践

周伟明 著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书以软件开发过程中涉及的测试知识为基础，主要讲解了测试用例设计方法、用例与代码评审、单元测试、集成测试、系统测试和一些专门的测试，如内存测试、性能测试、安全性测试等。本书还对测试方面的理论进行了一些探索，以测试空间和设计空间理论作为依据，贯穿于全书之中。特色之处是测试驱动设计、测试空间中的安全性准则、测试用例的设计准则等均以测试空间理论为基础，提出了新的思路。

本书可作为在职测试人员实践上的指南和理论上的辅导书，也可帮助程序员理解测试人员的工作内容和工作方式从而更好地配合测试人员完成项目测试，保证项目质量。

未经许可，不得以任何方式复制或抄袭本书的任何部分。

版权所有，侵权必究。

图书在版编目（CIP）数据

软件测试实践 / 周伟明著. —北京：电子工业出版社，2008.5
（测试实践丛书）

ISBN 978-7-121-05907-0

I. 软… II.周… III.软件—测试 IV.TP311.5

中国版本图书馆 CIP 数据核字（2008）第 013431 号

技术编辑：赖勇浩

责任编辑：杨绣国

印　　刷：北京市天竺颖华印刷厂

装　　订：三河市金马印装有限公司

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：787×980　1/16　印张：22　字数：370 千字

印　　次：2008 年 5 月第 1 次印刷

印　　数：5 000 册　　定价：45.00 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 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

丛书总序

初次听说电子工业出版社准备策划出版一套《测试实践丛书》，我就感到一种欣慰，这个选题对中国软件产业发展具有十分重要的现实意义。

当今世界软件产业之所以称之为产业，不仅是因为其产品的产值具有了超过传统产业的规模，而更重要的是因为其产品的工程化和工业化的生产与服务体系为整个人类社会提供了必要的质量保证。

随着用户对软件产品质量要求的不断提高以及软件工程技术的日益成熟，软件测试在软件生产与服务过程中成为一个越来越重要的环节，在软件企业、IT 服务企业、客户 IT 部门等机构中扮演着更为重要的角色。对于一个软件企业，“你不去发现缺陷，那一定是你的客户去发现缺陷”，提高软件产品质量已经成为增强企业竞争力的重要任务。

测试经理、测试工程师已经成为热门的职业，国际知名的软件企业和 IT 服务公司，从内设独立的测试部门、质量管理部门，到委托或承接第三方测试、测试外包、建立完整的缺陷测试管理与服务体系，组织形态日渐成熟，产业分工日趋细化，独立的测试行业已经初步形成。

近十几年来，在国家发展软件与集成电路产业政策的引导下，我国软件产业发展很快，软件企业成长迅速，但在整体上和发达国家软件企业差距还很大。其中的差距之一就体现在软件测试和质量控制上。虽然目前对测试的重视程度已经普遍提高，但是几乎所有的国内 IT 公司都存在测试时间不充分、测试软硬件资源不充分、缺乏合格的测试人力资源等问题。

发展中国的软件测试产业，无论是面向国际还是国内需求都应具有巨大的潜力和广阔的商业前景。其中，最关键的就是人才培养。培养一个优秀的测试工程师不容易，培养一个优秀的测试项目经理更困难。一个优秀的测试从业人员，不仅需要掌握测试理论、方法、

技术、工具，还需要深刻理解过程管理，更需要具有很高的素质（理念、思路、沟通、表达等）。一个优秀的测试企业需要把人（People）、过程（Process）和技术（Technology）三要素有机地结合好。发展中国的软件测试产业还需要一批既懂技术又懂管理的企业家，他们才是中国软件测试产业发展的主力军。

《测试实践丛书》致力于从实践的视角融会贯通测试的理论、技术和管理，通过案例分析真正让人们理解 People, Process, Technology 三角关系在测试行业的成功运用。丛书的作者们把他们多年来理性的思考和宝贵的实战经验奉献给读者，相信会给大家带来思考和启发。

最后，衷心希望这套丛书能够为培养一批有志于发展中国软件测试产业的测试技术人才和管理人才做出重要的贡献。

陈钟

陈钟老师简介：

2004 年被评为“影响中国软件开发的 20 人”之一。现任北京大学教授、博士生导师，北京大学软件与微电子学院院长、北京大学网络与信息安全实验室主任、北京大学工程学位评审委员会副主任。

社会兼职有中国软件行业协会常务理事、教育部计算机科学与技术教学指导委员会委员等。

此为试读，需要完整PDF请访问：www.ertongbook.com

作者序

无论是软件开发的过去、现在，还是未来，质量问题都一直是困扰所有软件开发人员的“疙瘩”。软件发展早期，软件规模较小，所以相对地软件 Bug 较少，涌现了许多擅长编写和调试软件的英雄式的程序员。但随着软件规模的增大，英雄难过质量关，出现了许多因为质量问题将上百万行源代码推倒重来的事例。与此同时，软件工程和测试技术应运而生并快速发展起来，在它们的辅助下软件质量得到了很大的改善，从而也使得软件开发的风险和成本都大幅度降低。

一个学科的发展通常要经过艺术、技术、原理三个阶段：软件测试最早便是以艺术出现的（最早见于 Myers, G. J.的《软件测试的艺术》一书）；近十年软件测试技术快速发展，但目前软件测试仍处于技术阶段；还没有到达原理（理论上的成熟）阶段。

在十余年的职业软件开发生涯中，我一直对测试情有独钟，不仅参与软件测试，而且经常在开发之余思考关于测试方面的问题，比如如何以低成本的有效方法进一步提高软件质量等。如此这般日积月累之后，却也积累了许多关于软件测试的心得，唯恐日后遗忘，又觉分享与众人才是正途，就产生了整理成书的想法，最后在博文视点的支持下终于付梓，了却一大心愿。

本书主要以我的实践经验为基础，以软件开发过程中涉及的测试技术为主，辅以作者平时对软件测试原理方面的探索。内容不仅涵盖测试用例设计方法、评审、单元测试、集成测试、系统测试等知识，还讲述了一些专门的测试如内存测试、性能测试、安全性测试等。第 1 章和第 2 章中提及的测试空间和设计空间等内容是作者对软件测试的原理性质的探索结果，虽然理论上可能仍有值得商榷之处，但的确是事事亲历，字字用心，唯愿与大家分享。

本书不适合作为初学者的第一本书，但可以作为一本纲领性质的指导书。本书部分内容要求读者有一定的编程基础知识，如其中有部分内容就需要读者有 C/C++ 编程方面的基础知识。

第 4 章的部分内容、第 5 章、第 6 章、第 8 章及附录 A 的内容都需要读者有 C/C++编程方面的知识才能阅读，其他章节的知识可以供无编程知识的测试人员阅读和参考。本书部分内容也可供软件设计人员和研究人员作参考。

本书的部分内容来自我的实践经验，其中多属于探索性内容，因此错误、遗漏和不足之处在所难免。真诚地希望读者能提出批评、建议和问题，可以将问题或建议回复到作者的博客 *http://blog.csdn.net/drzhouweiming* 里。另外本书出版后，相关堪误表也会及时在作者的博客里发布、更新。

致谢

本书花费了我一年多的专职时间写作而成，在写作过程中得到了许多朋友的帮助。其中最大的贡献来自赖勇浩先生，他花了近半年的业余时间负责编辑、加工书稿内容，并提出了许多有价值的修改意见，使得本书的质量得到较大的提升。当然还有许多提出过意见的朋友，因篇幅关系不能一一罗列，敬请谅解。

追本溯源，之所以能够写作本书，主要得益于 10 年前在 Santa Cruz 的岁月。特别是当时的项目经理 Neil Readshaw 对软件质量方面的严格要求，这使得质量意识深深植根于我的脑海，才能在后来一直关注和思考软件质量和测试方面的问题，并能有所心得，在此真诚地感谢 Dascom Inc 公司和 Neil Readshaw 先生。

另外，如果没有家人的支持，我也不可能有时间专门来写作本书，因此本书的很大一部分功劳要归功于他们，在此深深地感谢我的家人！

在写作过程中，书中的部分内容已发布到了作者的博客上，还有一部分内容发表在《程序员》杂志上，得到了很多读者反馈的宝贵意见和建议，在此对这些读者表示感谢！

由于从开始写作到现在已经过去了近两年的时间，有很多人对作者和本书提供了宝贵的意见和建议，但可能由于记忆的疏漏而无法记起您的名字，在此深表歉意，但作者一直记得哪些知识点、事例是来自大家的帮助，在这里谢谢大家！

周伟明

2007 年 12 月于上海

目录

此为试读,需要完整PDF请访问: www.ertongbook.com

第 1 章 软件测试概述

1

- 不经意丢下的一个烟头可以毁掉一座森林，这话用在软件上最恰当不过了，程序员编程时不经意的一个字节错误同样能使整个软件系统崩溃。
- 假设电脑只有电子的大小（约为 10^{-30}kg），假设将整个可观测到的宇宙物质（约为 10^{53}kg）全部做成电脑来运行测试，那么宇宙在毁灭了 $2.5\times10^{4\ 909}$ 次后才有可能将某些软件的所有测试执行完。
- 曾有智者说过，“一件小事乘以 13 亿之后就变成大事了”，用这句话来形容软件缺陷造成的破坏力真是太恰当了。一些软件的安全漏洞等看似不起眼的缺陷造成的破坏远远超过那些航天飞机失事、导弹误中自己人等事故，甚至远超过了一场现代战争造成的破坏。

1.1 测试的发展

在传统制造业等行业中，测试早就被当作控制产品质量的手段，不论是玩具还是飞机等都需要经过各种各样的测试才能投入使用。不过本书主要讨论的是软件测试，软件测试的发展经历以下几个阶段。

黑暗时代：计算机诞生后，早期还是使用机器语言编程，编写的程序也较简单。那时基本上是调试程序，只要程序能正常运行就可以了，谈不上进行什么测试，所以这个时代只能称为测试的黑暗时代。

石器时代：20 世纪 50 年代到 60 年代期间，高级语言开始出现了。IBM 公司的程序师约翰· 巴科斯（J. Backus）在 1954 年发明了 FORTRAN 语言。60 年代中期，美国达特默斯学院的托马斯· 卡茨（T.Kurtz）和约翰· 凯梅尼 （J. Kemeny）发明了 BASIC 语言。高级语言的出现使程序开始复杂起来，但这时期仍然是硬件占据主导地位，软件只是占次要地位。软件质量主要靠开发人员来保证，测试在这个时期还没有得到突飞猛进的发展，这个时代可以称为测试的石器时代。

青铜时代：20 世纪 70 年代，贝尔实验室的 Dennis M Ritchie 和 Ken Thompson 等人开发了 UNIX 操作系统和 C 语言。UNIX 和 C 语言的出现可以说是软件业的一次革命，C 语言迅速成为工业标准，结构化编程思想也在这个时代诞生了。有了 C 语言这种工具，系统软件编写不再是一件困难的事情。随着计算机硬件的高速发展，CPU 速度越来越快，内存和外存越来越大，软件规模也越做越大，于是对软件的可靠性等提出了更高要求，这个时期的测试工作主要是发现软件的缺陷。

铁器时代：20 世纪 80 年代，PC 机的高速发展使得 IT 业的规模迅速扩大，软件业已经进入了一个高速发展的年代。在这个时代，C++等面向对象的语言出现了。自此出现的大型软件的规模要以百万行为单位来计算，软件规模的增大对软件测试的要求更高了。自此，测试理论和方法开始快速发展，但软件测试主要还是以手工测试为主，软件质量的保证仍然依赖软件开发人员。

火药时代：90 年代后，由于多媒体技术的成熟，以及 Internet 的快速发展和大规模推广，PC 机开始在世界范围内普及，软件业进入了一个爆炸性增长的年代。这时的大型软件要以千万行为单位来计算，测试方面的要求也越来越高，不仅要发现缺陷，更重要的是缺陷的预防和预测，质量管理提到了新的高度。在这个时期，软件自动化测试工具开始大量出现，测试自动化成为测试的重要手段，测试缺陷分析技术也得到蓬勃发展。

工业时代：软件测试进入了工业时代吗？从目前软件的发展来看，虽然软件的规模越来越大，但不等于进入了工业时代，就像建造了长城、金字塔这样宏伟的工程不表示社会进入了工业时代一样。社会进入工业时代的标志是大量的机器工具的使用使得人均生产率大幅提高，但现实中我们可以看到，大型软件的人均生产率却是越来越低，从技术上看并无大的突破性技术出现，软件工业时代的来临还有很长的路要走。

1.2 测试的目的

软件测试的目的是什么？这个话题在许多书中都有提及。初学者也许会认为测试是为了找出软件中的所有缺陷，但这个想法在实践中只能是一个美好的愿望，在现实情况中是不可能实现的。为了说明不可能找出所有的缺陷，请看下面关于测试能否证明软件没有缺陷的分析。

1.2.1 测试能证明软件没有缺陷吗

一些初学者可能以为只要肯投入足够的人力和时间，对软件进行完全的测试，就能够找出所有的缺陷，这种看法是否正确呢？我们可以先来看一下下面的实例：

图 1-1 是 Windows 2000 操作系统中的一个命令行窗口，在命令行窗口中可以输入命令，命令包含的字符可以是键盘上任意可输入的字符。

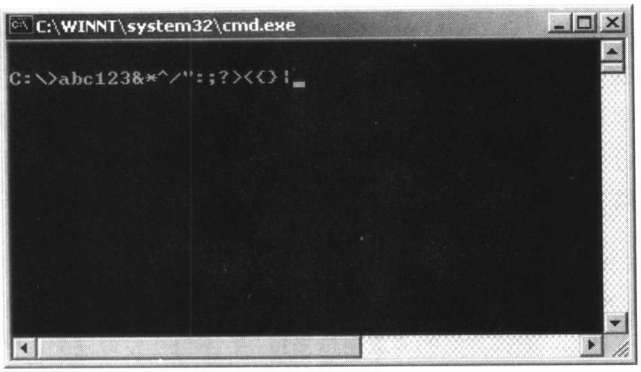


图 1-1 Windows 命令行窗口图

我们来看一下测试这个命令行程序需要多少测试用例，你可以任意输入 1234、abcd1234、abc*&^1234\{等各种长度的各种组合，最长可以输入 2046 个字符长度的字符串。

只考虑英文输入的情况下，键盘上的可显字符就有 94 个之多，如果考虑其他语言，则输入每个字符的可能性最多可以有 256 种，因此所有可能输入的组合有 $256^{2\,046}=2^{16\,368}$ 种，这个数字已经远远超过了宇宙常量普朗克常数的大小。

假设摩尔定律永远有效，如果按照摩尔定律每 18 个月电脑的处理速度翻一番，那么至少要在 2 万 4 千年以后，电脑才有能力对这个命令行窗口程序进行完全测试。摩尔定律会永久有效吗？现实中，摩尔定律才运行了二十几年，电脑速度就已快到极限了，所以要想给程序进行完全测试，在大多数现实情况下是根本不可能的。

也许有人会说现在科技还不够发达，如果科技有突破的话，说不定计算机速度将来能达到非常高，软件完全测试在将来也许有可能呢！我们不妨再做个有趣的计算试试。

目前的超级计算机速度也才达到百万亿次的级别，即 10^{14} 级别的运算速度。假设将来计算机速度达到了宇宙常量级别的速度，即运算速度达到了每秒运行 10^{63} 次的级别。再假设科技足够发达，一台电脑只有一个电子大小。假设把目前可观测到的整个宇宙的物质全部变成电脑，目前可观测到的宇宙质量大约有 10^{53}kg，电子的质量约为 10^{-30}kg，因此总共有 10^{83} 台运行速度为 10^{63} 的电脑，假设一条指令就可以校验一次上述命令行的输入，那么所有的电脑运行 1 秒钟可以校验 10^{146} 种输入，而 $2^{16\,368}\approx10^{4\,927}$，总共要运行 $10^{4\,781}$ 秒的时间才可以将整个程序测试完。宇宙的寿命大约为 4×10^{17} 秒，如果要将这个程序测完，宇宙都毁灭了 $2.5\times10^{4\,763}$ 次了。所以不管将来科技如何发达，要想将软件完全测试永远都是不可能的事情。

既然无法对软件进行完全的测试，测试当然不能证明软件没有缺陷。

在 Cem Kaner、Jack Falk 著的《计算机软件测试（第二版）》中，指出了以下几种情况都是不可能完全测试的：

（1）不可能测试到程序对任何可能输入的响应；

（2）不可能测试到程序每一条可能的执行路径；

（3）无法找到所有的设计错误；

（4）不能采用逻辑来证明程序正确性。

本书第 3 章用例设计方法中也会讲到另外两点：

（5）不可能测试到所有可能的场景；

（6）不可能对所有隐性元素进行测试。

既然不能证明软件没有缺陷，那么反正测试过的软件可能还会有大量的缺陷在里面，是不是就不需要测试软件了呢？不测试软件，那么遗留的缺陷就更多，软件将不可用，所以测试是必须的。测试能做什么呢？首先，我们有必要了解清楚测试的目的是什么。

1.2.2 测试的目的

测试既然不是用来证明软件没有缺陷的，那测试的目的到底是什么呢？测试主要有以下 8 个方面的目的：

1. 验证软件需求和功能是否得到完整实现

测试首先必须用来验证软件的需求和功能是否得到完整实现。系统测试是根据软件产品需求规格来进行的，所以在进行系统测试时可以实现所有遗漏的需求。但在开发时还是需要通过需求跟踪来保证需求在开发的各阶段都得到了完整实现。

其次是验证软件在正常和非正常情况下的功能和特性。测试不仅要验证软件在正常情况下的功能和特性是否可以使用和达到期望值，更多的是验证在非正常情况下功能和特性能否达到期望的要求。比如一个计算器程序，正常情况下当用户输入正确的数学表达式时要能够计算出正确的结果，但是当用户输入错误的表达式时，软件要给用户提示说用户输入错误，而不能出现运行异常（如崩溃）。

2. 验证软件是否可以发布使用

软件是否可以发布使用需要经过测试来验证，未经测试的软件是不能发布的。即使是内

部使用的软件也同样需要测试，软件的发布需要经过验收测试。

3. 发现软件系统的缺陷、错误及不足

软件系统的缺陷、错误及不足需要经过测试来发现。目前发现软件系统的缺陷、错误及不足的主要手段有评审、检视、走读、单元测试、集成测试、系统测试等。

4. 获取软件产品的质量信息

软件产品的质量信息也必须通过测试才能获取，没有经过测试的软件，软件质量的好坏是无从知道的，最多只能根据开发人员的水平进行推测。经过测试后，就可以得到开发各阶段发现的缺陷数，进而可以较为准确地推测出软件潜在的缺陷数。

5. 预防下一版本可能出现的问题

测试不仅可以用来发现当前版本的问题，还可以根据目前发现的问题进行分析，找出当前版本出现的问题有哪些类型，产生这些类型问题的根源是什么。比如当前版本出现了很多指针方面的错误，那么就可以针对指针操作加强员工培训。

6. 预防用户使用软件时可能出现的问题

把没有经过测试的软件提供给用户使用，将会使用户在使用过程中遭受大量挫折，大大降低了愉快的用户体验。测试可以有效地发现大部分影响使用的错误，经修正后软件预防了用户使用软件时可能出现的问题。

7. 提前发现开发过程中的问题和风险

测试还能提前发现开发过程中的问题和风险，这是为什么呢？因为写系统测试用例时可以发现需求中的问题和遗漏，写集成测试用例时可以发现高层设计中的问题，写单元测试用例则可以发现详细设计和编码中的问题。通过测试，可以在早期阶段就发现这些错误，大大降低开发的风险。

8. 提供可以用以分析的测试结果数据

测试还能提供用以分析的测试结果数据、测试问题记录表等数据。在测试完后进行分析，可以了解主要有哪些类型的缺陷，进而分析产生这些类型缺陷的原因。还可以分析开发各阶段发现的问题数，把他们与以前的经验数据进行对比分析，从而知道在开发阶段中哪个阶段是薄弱环节，进而重点针对薄弱环节进行加强。

此为试读，需要完整PDF请访问：www.ertongbook.com