

软件测试

(英文版·第2版)

Ron Patton

Software Testing

Second Edition

SAMS

(美) Ron Patton 著

经典原版书库

软件测试

(英文版·第2版)

Software Testing

(Second Edition)

江苏工业学院图书馆
藏书章

(美) Ron Patton 著



机械工业出版社
China Machine Press

English reprint edition copyright © 2006 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Software Testing, Second Edition* (ISBN 0-672-32798-8) by Ron Patton, Copyright © 2006.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Sams.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版由Pearson Education Asia Ltd.授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

本书封面贴有Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2005-4835

图书在版编目(CIP)数据

软件测试(英文版·第2版)/(美)巴顿(Patton, R.)著. -北京:机械工业出版社, 2006. 1

(经典原版书库)

书名原文: *Software Testing, Second Edition*

ISBN 7-111-17770-3

I. 软… II. 巴… III. 软件-测试-英文 IV. TP311.5

中国版本图书馆CIP数据核字(2005)第127467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:迟振春

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2006年1月第1版第1次印刷

718mm×1020mm 1/16·25.75印张

印数:0 001-3 000册

定价:38.00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

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权

权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件：hzsj@hzbook.com

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

About the Author

Ron Patton lives in the Seattle area with his wife, Valerie. His software test experience is wide and varied, from mission-critical systems to painting programs for kids. Ron graduated from Penn State in 1984 with a B.S. degree in computer science. He began his career at Texas Instruments as a quality assurance engineer, testing embedded systems and user interface software for industrial automation equipment. In 1992 he joined Microsoft as a software test lead in the Systems Group for Multimedia Viewer, the authoring tool and multimedia display engine used by Encarta, Cinemania, and Bookshelf. He moved on to become the software test manager of the Kids Product Unit, shipping CD-ROM titles such as Creative Writer, Fine Artist, 3D Movie Maker, and the Magic School Bus series. Before he left Microsoft, he was the software test manager of the Hardware Group, responsible for the software shipped with the mouse, keyboard, gaming, telephony, and ActiMates product lines. He currently is a freelance project management and software quality consultant.

Ron's most memorable project was ActiMates Barney, for which he test managed both the hardware and software efforts. "Microsoft actually paid my team and me to shake, bake, freeze, thaw, pull, drop, tumble, dunk, and shock dozens of prototype Barney dolls until we reduced them to piles of electronic rubble and purple fuzz," he recalls. "You can't get much more test satisfaction than that."

If you have comments or suggestions for this book, or if you find a bug in it that you want to report, you can send Ron an email at test@valart.com.

Dedication

To my best friend and wife, Valerie, who's hoping that after I finish this second edition we can go on vacation to a tropical island.

Acknowledgments

Many thanks go to Sams Publishing and the editors and staff who helped me publish this second edition. A big thank you goes to Danny Faught who provided great input as an expert reviewer.

To my parents, Walter and Eleanore, for allowing me to quit my accordion lessons and buying me a TRS-80 Model I computer back in 1977. To my sister, Sandra, for keeping my parents busy with her baton competitions so I could hide in my room and learn to program. To Ruth Volland, my computer science teacher at Mohawk High School, for dragging me to all those science fairs and giving me extra time on the school's ASR 33 teletypes. To Mark Ferrell, who taught me electronics and kept me out of trouble as a teenager. To Alan Backus and Galen Freemon of TI for allowing me the freedom to explore software test automation. To all my past co-workers and employees for teaching me more than I could have ever learned myself about software testing. And, to my wonderful wife, Valerie, for saying, "Go ahead, send it in, see what happens" when, in 1991, I posed the question of sending my résumé to a little company in far-away Seattle called Microsoft. Each of you made a contribution to this book. Thank you!

We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

As a publisher for Sams Publishing, I welcome your comments. You can email or write me directly to let me know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that I cannot help you with technical problems related to the topic of this book. We do have a User Services group, however, where I will forward specific technical questions related to the book.

When you write, please be sure to include this book's title and author as well as your name, email address, and phone number. I will carefully review your comments and share them with the author and editors who worked on the book.

Email: feedback@sampublishing.com

Mail: Paul Boger
Publisher
Sams Publishing
800 East 96th Street
Indianapolis, IN 46240 USA

For more information about this book or another Sams Publishing title, visit our website at www.sampublishing.com. Type the ISBN (0672327988) or the title of a book in the Search field to find the page you're looking for.

Table of Contents

Introduction	1
About the Second Edition	1
Who Should Use This Book?	2
What This Book Will Do for You	2
Software Necessary to Use This Book	3
How This Book Is Organized	3
Part I: The Big Picture	3
Part II: Testing Fundamentals	4
Part III: Applying Your Testing Skills	4
Part IV: Supplementing Your Testing	5
Part V: Working with Test Documentation	5
Part VI: The Future	5
Appendix	6
Conventions Used in This Book	6
 Part I The Big Picture	
 1 Software Testing Background	 9
Infamous Software Error Case Studies	9
Disney's <i>Lion King</i> , 1994–1995	10
Intel Pentium Floating-Point Division Bug, 1994	10
NASA Mars Polar Lander, 1999	11
Patriot Missile Defense System, 1991	12
The Y2K (Year 2000) Bug, circa 1974	12
Dangerous Viewing Ahead, 2004	13
What Is a Bug?	13
Terms for Software Failures	13
Software Bug: A Formal Definition	14
Why Do Bugs Occur?	16
The Cost of Bugs	18
What Exactly Does a Software Tester Do?	19
What Makes a Good Software Tester?	20
Summary	21
Quiz	22

2	The Software Development Process	23
	Product Components	23
	What Effort Goes Into a Software Product?	24
	What Parts Make Up a Software Product?	28
	Software Project Staff	29
	Software Development Lifecycle Models	30
	Big-Bang Model	31
	Code-and-Fix Model	32
	Waterfall Model	33
	Spiral Model	34
	Summary	36
	Quiz	36
3	The Realities of Software Testing	37
	Testing Axioms	38
	It's Impossible to Test a Program Completely	38
	Software Testing Is a Risk-Based Exercise	39
	Testing Can't Show That Bugs Don't Exist	40
	The More Bugs You Find, the More Bugs There Are	41
	The Pesticide Paradox	41
	Not All the Bugs You Find Will Be Fixed	42
	When a Bug's a Bug Is Difficult to Say	43
	Product Specifications Are Never Final	44
	Software Testers Aren't the Most Popular Members of a Project Team	45
	Software Testing Is a Disciplined Technical Profession	45
	Software Testing Terms and Definitions	46
	Precision and Accuracy	46
	Verification and Validation	47
	Quality and Reliability	48
	Testing and Quality Assurance (QA)	48
	Summary	49
	Quiz	49

Part II Testing Fundamentals

4	Examining the Specification	53
	Getting Started	53
	Black-Box and White-Box Testing	55
	Static and Dynamic Testing	56
	Static Black-Box Testing: Testing the Specification	56

Performing a High-Level Review of the Specification	57
Pretend to Be the Customer	57
Research Existing Standards and Guidelines	58
Review and Test Similar Software	59
Low-Level Specification Test Techniques	60
Specification Attributes Checklist	60
Specification Terminology Checklist	61
Summary	61
Quiz	62
5 Testing the Software with Blinders On	63
Dynamic Black-Box Testing: Testing the Software While Blindfolded	64
Test-to-Pass and Test-to-Fail	66
Equivalence Partitioning	67
Data Testing	70
Boundary Conditions	70
Sub-Boundary Conditions	75
Default, Empty, Blank, Null, Zero, and None	77
Invalid, Wrong, Incorrect, and Garbage Data	78
State Testing	79
Testing the Software's Logic Flow	80
Testing States to Fail	84
Other Black-Box Test Techniques	87
Behave Like a Dumb User	87
Look for Bugs Where You've Already Found Them	88
Think like a Hacker	88
Follow Experience, Intuition, and Hunches	88
Summary	89
Quiz	89
6 Examining the Code	91
Static White-Box Testing: Examining the Design and Code	91
Formal Reviews	92
Peer Reviews	94
Walkthroughs	95
Inspections	95
Coding Standards and Guidelines	96
Examples of Programming Standards and Guidelines	96
Obtaining Standards	98

Generic Code Review Checklist	99
Data Reference Errors	99
Data Declaration Errors	100
Computation Errors	101
Comparison Errors	101
Control Flow Errors	102
Subroutine Parameter Errors	102
Input/Output Errors	102
Other Checks	103
Summary	103
Quiz	104
 7 Testing the Software with X-Ray Glasses	105
Dynamic White-Box Testing	106
Dynamic White-Box Testing Versus Debugging	107
Testing the Pieces	108
Unit and Integration Testing	109
An Example of Module Testing	111
Data Coverage	113
Data Flow	114
Sub-Boundaries	115
Formulas and Equations	115
Error Forcing	116
Code Coverage	117
Program Statement and Line Coverage	118
Branch Coverage	119
Condition Coverage	120
Summary	121
Quiz	122
 Part III Applying Your Testing Skills	
 8 Configuration Testing	125
An Overview of Configuration Testing	126
Isolating Configuration Bugs	129
Sizing Up the Job	131
Approaching the Task	132
Decide the Types of Hardware You'll Need	133
Decide What Hardware Brands, Models, and Device Drivers Are Available	133

Decide Which Hardware Features, Modes, and Options Are Possible	134
Pare Down the Identified Hardware Configurations to a Manageable Set	134
Identify Your Software's Unique Features That Work with the Hardware Configurations	135
Design the Test Cases to Run on Each Configuration	136
Execute the Tests on Each Configuration	137
Rerun the Tests Until the Results Satisfy Your Team	137
Obtaining the Hardware	137
Identifying Hardware Standards	139
Configuration Testing Other Hardware	139
Summary	140
Quiz	140
9 Compatibility Testing	141
Compatibility Testing Overview	142
Platform and Application Versions	143
Backward and Forward Compatibility	144
The Impact of Testing Multiple Versions	144
Standards and Guidelines	146
High-Level Standards and Guidelines	147
Low-Level Standards and Guidelines	148
Data Sharing Compatibility	148
Summary	150
Quiz	151
10 Foreign-Language Testing	153
Making the Words and Pictures Make Sense	154
Translation Issues	154
Text Expansion	155
ASCII, DBCS, and Unicode	155
Hot Keys and Shortcuts	156
Extended Characters	157
Computations on Characters	158
Reading Left to Right and Right to Left	158
Text in Graphics	159
Keep the Text out of the Code	159

Localization Issues	160
Content	160
Data Formats	162
Configuration and Compatibility Issues	163
Foreign Platform Configurations	163
Data Compatibility	165
How Much Should You Test?	166
Summary	167
Quiz	168
11 Usability Testing	169
User Interface Testing	170
What Makes a Good UI?	170
Follows Standards and Guidelines	171
Intuitive	173
Consistent	173
Flexible	175
Comfortable	176
Correct	176
Useful	178
Testing for the Disabled: Accessibility Testing	178
Legal Requirements	179
Accessibility Features in Software	180
Summary	182
Quiz	182
12 Testing the Documentation	183
Types of Software Documentation	183
The Importance of Documentation Testing	187
What to Look for When Reviewing Documentation	188
The Realities of Documentation Testing	189
Summary	190
Quiz	190
13 Testing for Software Security	193
<i>WarGames</i> —the Movie	194
Understanding the Motivation	195
Threat Modeling	197
Is Software Security a Feature? Is Security Vulnerability a Bug?	200
Understanding the Buffer Overrun	201

Using Safe String Functions	203
Computer Forensics	205
Summary	208
Quiz	209
14 Website Testing	211
Web Page Fundamentals	212
Black-Box Testing	213
Text	215
Hyperlinks	216
Graphics	217
Forms	217
Objects and Other Simple Miscellaneous Functionality	218
Gray-Box Testing	218
White-Box Testing	220
Configuration and Compatibility Testing	222
Usability Testing	224
Introducing Automation	226
Summary	227
Quiz	227
Part IV Supplementing Your Testing	
15 Automated Testing and Test Tools	231
The Benefits of Automation and Tools	231
Test Tools	233
Viewers and Monitors	234
Drivers	235
Stubs	236
Stress and Load Tools	237
Interference Injectors and Noise Generators	238
Analysis Tools	239
Software Test Automation	239
Macro Recording and Playback	240
Programmed Macros	242
Fully Programmable Automated Testing Tools	243
Random Testing: Monkeys and Gorillas	245
Dumb Monkeys	246
Semi-Smart Monkeys	248
Smart Monkeys	248

xviii Software Testing

Realities of Using Test Tools and Automation	250
Summary	251
Quiz	252
16 Bug Bashes and Beta Testing	253
Having Other People Test Your Software	253
Test Sharing	255
Beta Testing	256
Outsourcing Your Testing	258
Summary	259
Quiz	259

Part V Working with Test Documentation

17 Planning Your Test Effort	263
The Goal of Test Planning	264
Test Planning Topics	265
High-Level Expectations	265
People, Places, and Things	266
Definitions	267
Inter-Group Responsibilities	268
What Will and Won't Be Tested	270
Test Phases	270
Test Strategy	271
Resource Requirements	271
Tester Assignments	272
Test Schedule	272
Test Cases	274
Bug Reporting	274
Metrics and Statistics	274
Risks and Issues	275
Summary	275
Quiz	275
18 Writing and Tracking Test Cases	277
The Goals of Test Case Planning	277
Test Case Planning Overview	279
Test Design	281
Test Cases	283
Test Procedures	285