



PEARSON
Prentice
Hall

Software Quality Institute Series

快速测试 (影印版)

RAPID TESTING

Robert Culbertson

[美]

Chris Brown

著

Gary Cobb

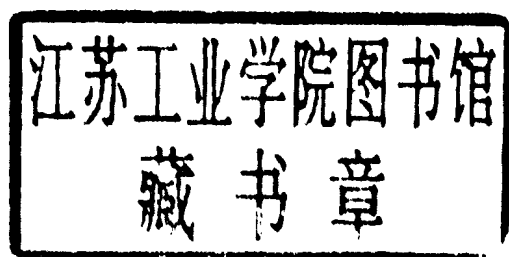


清华大学出版社

快速测试

Rapid Testing (影印版)

Robert Culbertson
Chris Brown 著
Gary Cobb



清华大学出版社
北 京

English reprint edition copyright © 2004 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: *Rapid Testing*, 1st Edition by Robert Culbertson, Chris Brown and Gary Cobb, Copyright © 2003

All Rights Reserved.

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

This edition is authorized for sale and distribution only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong, Macao SAR and Taiwan).

本书影印版由 Pearson Education (培生教育出版集团) 授权给清华大学出版社出版发行。

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

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。

北京市版权局著作权合同登记号 图字 01-2003-8778 号

版权所有, 翻印必究。

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。

图书在版编目(CIP)数据

快速测试 = *Rapid Testing* / (美) 卡伯特森 (Culbertson, R.), (美) 布朗 (Brown, C.), (美) 科布 (Cobb, G.) 著. —影印本. —北京: 清华大学出版社, 2004.4

ISBN 7-302-08152-2

I. 快… II. ①卡… ②布… ③科… III. 软件—测试—方法—英文 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2004) 第 013350 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社总机: 010-6277 0175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-6277 6969

责任编辑: 郭福生 常晓波

封面设计: 立日新

印 刷 者: 世界知识印刷厂

装 订 者: 北京市鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 180×230 印张: 25.75

版 次: 2004 年 4 月第 1 版 2004 年 4 月第 1 次印刷

书 号: ISBN 7-302-08152-2/TP · 5890

印 数: 1~3000

定 价: 42.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010) 62770175-3103 或 (010) 62795704

序

相信很多朋友都有过类似的经历：在银行的自动取款机取钱，输入要取的金额后，取款机半天也没反应；或者是打电话查询自己的电话费，一直都是“系统忙，请稍候再拨”的提示；更为普遍的是许多政府网站点击不了几下就报 ASP 页面错误等。这些软件系统的开发商如果能早一点发现这些问题多好啊（虚拟语气）。

其实，上面所列问题都是应该在系统开发和试运行阶段就应该暴露出来并得到解决。暴露这些问题就责无旁贷地落在测试人员的头上。尽管我们都知道软件质量问题可能来源于需求错误、设计不合理或编码错误等其他原因。但是别忘了，测试是将产品移交给客户之前开发方工作的最后一个主要环节，所以提高测试效率对一个软件项目的成功就显得格外重要。

本书正是这样一本全面介绍和分析测试过程和技术的专业书籍。作者介绍了测试过程的基本概念，然后又分专题讨论了测试中常用的技术与技巧，最后给出了一整套与测试工作相关的样例文档。与许多同类的测试书籍比较，作者更为全面深入地分析了测试工作。例如 JAR（Joint Application Requirement）方法、度量方法、评审技术等。尤其是相关的度量方法，更是进一步分析了软件项目管理中两个前沿专题：软件估计与度量，作者将这些方法应用到软件测试的工作分析过程中，对于提高测试工作效率和管理水平显得尤为重要。此外，作者还不时在书中穿插自己在实际项目中的体会，说明测试方法与技巧的具体应用，读者可以轻松领会这些方法的微妙之处。

本书可以作为软件行业从业人员以及软件学院学生了解测试工作的参考读物，尤其适合作为软件测试人员和管理人员的案头常备读物，不时翻阅，相信当会开卷有益。

曹 济
北京 SPIN

Preface

This book presents a practical approach to software testing, placing emphasis on a test process that is geared to today's aggressive pace of software development. It is designed for use by practicing test engineers and test managers, and includes tips, techniques, and examples that can be used to improve the efficiency and speed of software testing. The book should also be suitable for people who are beginning a career in software testing. It includes an extensive set of references that are intended to support established test professionals as well as those just starting out.

The speed and efficiency of software testing depends how well the test process integrates with the overall development life cycle and upon the efficiency of the test techniques used. This book shows how to improve the speed and efficiency of your testing effort by focusing on three areas:

- Beginning the testing life cycle simultaneously with the project's requirements phase, so that bugs can be caught as early as possible and so test planning and test case development can start as soon as possible.
- Presenting efficient static testing techniques such as inspections and walkthroughs that can be used to test work products that are generated during the development life cycle.
- Presenting efficient dynamic testing techniques that can be used to find bugs during integration, system, and acceptance test phases.

Key Features

The following features of this book are intended to help you improve your software testing efficiency:

- An emphasis is placed on tailoring the test processes to meet aggressive time-to-market goals without sacrificing product quality.

- Software testing is placed in the context of the overall software development life cycle. The development life cycle is viewed from the vantage point of the test engineer. Development models such as evolutionary prototyping and spiral development models are considered, as well as the traditional waterfall.
- Static testing techniques are presented that can be used to get the test team involved early in the development life cycle. Static testing will find bugs early in the life of the product and will allow you to define test plans and test cases as soon as possible.
- Includes examples of key work products of the test process.

How the Book is Organized

This book consists of three parts that are organized as follows:

Part I. The Rapid Testing Process—This part defines the basic concepts and terms related to software testing. It describes rapid testing processes that are closely integrated with the overall software development life cycle. The traditional waterfall development model is considered, as well as life cycles based on incremental delivery and evolutionary prototyping. Each phase of the software development process is examined from the viewpoint of the software test engineer, and methods of bug detection and prevention are described as means of improving test efficiency.

Part II. Rapid Testing Techniques and Tips—This part describes in detail the tips and techniques that can be used to develop a rapid test process. Methods are presented for eliciting and analyzing requirements, estimating and scheduling the test effort, conducting inspections and reviews, designing black box tests, and reporting test results. A variety of dynamic testing methodologies are discussed, including functional analysis, equivalence partitioning, boundary value analysis, testing for memory leaks, use-case testing, and performance tests.

Part III. Rapid Testing Examples—Part III contains a set of examples for the process and techniques discussed in the first two parts of the book. The examples are based on the Test Management Toolkit (TMT), which is a pedagogical application that allows test managers and engineers to manage test plans, bug reports, test results, and other information related to software testing. It is a web-based application that permits several

users, even when geographically dispersed, to support one or more test projects simultaneously.

There are examples of four key work products related to the test process:

- Requirements Definition
- Test Plan
- Test Procedure Specification
- Test Summary Report

About the Authors

Robert Culbertson has over 25 years of experience in engineering, software development, software testing, and project management. In work at Cisco Systems, Texas Instruments, IBM, the University of Texas, and DSC Communications, he has experienced first hand the subjects presented in *Rapid Testing*. Robert has B.S.E.E. and M.S.E.E. degrees from the University of Texas at Austin, and a Ph.D. in electrical and electronic engineering from the University of Birmingham in England.

Gary Cobb has pursued a dual career of teaching and working in industry in the Austin area over the last 25 years. He has taught in the University of Texas at Austin Departments of Mathematics, Computer Science, and Electrical and Computer Engineering. He has also taught multimedia courses for the Department of Computer Science at Southwest Texas State University, where he was the Director of the Multimedia Laboratory. His industrial experience includes full-time employment at Texas Instruments Inc., Lockheed Martin, and Dell Computer Corporation. He teaches short courses for the University of Texas' Software Quality Institute, which also sponsored his development of the Software Criteria for the Greater Austin Quality Award. Gary has a Ph.D. in mathematics from the University of Texas in Austin.

Chris Brown has over 20 years experience in the software and computer industry. He has worked in various testing roles at Advanced Micro Devices, Cisco Systems, Compaq Computer Corporation, and IBM. At AMD he was responsible for systems level silicon validation and compatibility testing of all available hardware and software configurations. At Compaq he was responsible for scripting and testing of prototype and pilot computer

systems for the Portable Products Computing Division. Chris was the team lead for OS/2 Database Manager, Communications Manager, LAN Transport and LAN Manager defect support while at IBM. He was President/CEO of Computer Security Corporation, a company that sold virus protection software to the United States Air Force, Prudential, Boeing, and other major companies and labs. He also served as a Regional Manager and Vice President at Dataserv Computer Maintenance/BellSouth, where he was responsible for field maintenance of over 40,000 computers. Chris has a Bachelor of Science degree in Electrical Engineering and an MBA.

Acknowledgments

The authors are indebted to Al Dale, founder of the Software Quality Institute (SQI), for his support of this book and the entire SQI book series. We also wish to thank Paul Petralia for his encouragement and support throughout the writing and production process. Thanks also go to the capable members of the SQI and Prentice Hall teams for their constant support and professional help. Jessica Balch and the production team at Pine Tree Composition did an excellent job of bringing the book to life, and it was a pleasure working with them.

Robert Culbertson wishes to acknowledge his family—Terri, Shelley, and Michael—who encouraged Dad and who gave up countless hours of family time in support of this project. Robert thanks his mom, Lorene, and brother, Lawrence, for their unfailing support. Robert is also grateful to all the managers, engineers, and teachers who have served as mentors and colleagues throughout the years. In particular, thanks go to Bob Marinconz, Mark Sherwood, Mark Schnucker, Karen Boyd, Mark Scafidi, Loy Dossman, John Whitworth, Arthur Holly, Orhan Berkay, Tom Muir, and David Blackstock.

Gary Cobb thanks his wife, Marilyn, for her patience and support during the long hours dedicated to writing his portion of this book. He also thanks their children, Glen Cobb, Molly Pierson, Stephen Cobb, and Meredith McLarty for their support. For the people who have had significant impact on Gary's career, he wants to thank his parents, J. D. and Mary Louise Cobb and the following teachers and professionals: Dr. W. T. Guy, Dr. H. S. Wall, Dr. Herbert Woodson, Dr. Roger Bate, L. C. Craig, Joanne Click, Ed Martin, Jim Nelson, and Mike Crosby.

Contents

Preface	<i>xv</i>
----------------	-----------

Part I	
The Rapid Testing Process	1

Chapter 1	
Introduction to Rapid Testing	3

Basic Definitions for Software Testing	4
What Is Rapid Testing?	6
People	7
Integrated Test Process	8
Static Testing	8
Dynamic Testing	8
Developing a Rapid Testing Strategy	9
The Software Development Process	9
A Waterfall Test Process	12
Requirements Analysis	14
Test Planning	15
Test Design, Implementation, and Debugging	16
System Test	16
Acceptance Test	17
Maintenance	17
Tying Testing and Development Together	18
What's Next	21
References	21

Chapter 2

Requirements Analysis & Testing

23

- The Requirements Process 23
- Eliciting Requirements 27
 - Requirements Definition 30
 - Requirements Specification 33
- Requirements Traceability Matrix 35
- Requirements Testing 36
 - What to Look for When Testing Requirements 37
 - The Use of Prototypes 39
 - Testing in an Evolutionary Prototyping Life Cycle 40
- What's Next 42
- References 43

Chapter 3

Test Planning

45

- Test Strategy 47
- Define the Scope of Testing 48
 - Define the Testing Approach 51
 - Define Testing Criteria and Quality Checkpoints 53
 - Define Automation Strategy 54
- Defining the Test System 58
 - Test Architecture 58
 - Test Tools 60
 - Test Environment 61
 - Test Configurations 64
- Estimating the Test Effort 65
 - Identify Tasks 67
 - Determine Effort 69
 - Determine Duration and Construct Schedule 73
 - Assess Schedule Risk 75
- Preparing and Reviewing the Test Plan Documents 76
 - Test Plan Format 76
 - Test Plan Reviews 85

What's Next 86

References 86

Chapter 4

Test Design and Development

89

Test Design 90

 Define Test Objectives 93

 Define Input Specifications 94

 Define Test Configurations 94

 Test Design Document 95

Developing Test Cases 96

 Develop Detailed Test Procedures 98

 Define Expected Results 100

 Setups and Cleanups—Testing from a Known State 101

 Test Case Template 102

 Test Case Configuration Management 104

Verifying and Debugging Tests 105

Test Case Automation 106

What's Next 107

References 108

Chapter 5

System Testing

109

Finding and Tracking Bugs 111

 Define Bug States 111

 Bug Tracking Essentials 114

 Write Bug Reports 118

 Bug Reviews 119

Running Tests 122

 Entry into System Testing 122

 Test Cycles 122

 Log Test Results 124

Reporting Test Results	126
Test Status Report	126
Bug Status Report	127
Test Summary Report	128
Exit Criteria and Release Readiness	130
What's Next	131

Chapter 6

Putting It All Together: People and Process 133

The People Side of Testing	134
Traits of a Successful Test Engineer	134
Traps to Avoid	136
Interviewing Tips	137
Improving the Test Process	140
The Software Engineering Institute's Capability Maturity Model (CMM)	141
Relationship of the CMM to Rapid Testing	144
An Approach to Process Improvement	146
What's Next	147
References	148

Part II

Rapid Testing Techniques and Tips 149

Chapter 7

Introduction to Testing Techniques and Tips 151

The Scope of Testing Techniques	151
Development Life Cycle	152
Benefits of Rapid Testing	155
Definition of Static Testing	156
Definition of Dynamic Testing	157
Life Cycle of a Bug	158

Formal Test Phases	160
Responsibilities of Test Team Members	161
What's Next	161

Chapter 8

Joint Application Requirements (JAR): A Method for Eliciting Requirements Using Rapid Testing 165

JAR Methodology	166
Roles of the Test Engineer in a JAR	173
Summary	175

Chapter 9

Static Testing Techniques and Tips 177

Cyclomatic Complexity and Its Relationship to Test Executions	178
Example of Graphing the Design of a Module	179
Formal Reviews	182
Using Checklists	185
Audits	186
Inspections/Walkthroughs/Peer Reviews	188
Inspection Team Roles and Responsibilities	189
Reporting on the Inspection Process	192
Metrics for the Inspection Process	192
Using e-mail or an e-App to Streamline Inspections	193
Formal Verification	194
Specification-Based Languages	195
Automated Theorem Proofs	195
Testing Automation/Tools	196
Requirements Traceability	196
Physical Units Checker	197

Symbolic Execution	197
Cross-Reference Listings	198
Pretty Printers	199
Version Comparators	199
Algorithm Testing	200
Test Support Facilitator	203
Shared Issues Databases	204
Summary	205
References	205

Chapter 10

Dynamic Testing Techniques and Tips

207

Functional Testing and Analysis	209
Equivalence Partitioning	210
Boundary Value Analysis	211
Negative Testing	211
Risk-Based Testing	213
Path Coverage Testing	215
Use-Case Testing	221
Bebugging/Mutation	222
Tracing/Tracebacks/Snap Dumps/Post-Mortem Dumps	223
Breakpointing/Patching	224
Data Flow Testing	226
Testing for Memory Leaks	227
Human-Computer Interface Testing	229
Load Performance Testing	230
Platform Configuration Testing	234
Summary	236
References	237

Chapter 11**Developing and Using Testing Metrics: Modeling and Forecasting Bugs**

239

- Definition of Metric and Measurement 241
- Using Standard Metrics to Make Improvements 250
- Metrics for Testing 253
 - Bug Density (Bugs/KSLOC) 254
- Project-Based Bug Model 255
- Software Error Estimation Program (SWEEP) 257
- Summary 261
- References 262

Chapter 12**Testing Effort Estimation Techniques and Tips**

263

- Using Mathematics to Estimate Software Developments 267
- Function Point Technology 287
- Summary 290
- References 291

Part III**Rapid Testing Examples**

293

Chapter 13**Example Requirements**

295

- Test Management Toolkit Release 1.0 Requirements Definition 297

Chapter 14**Example Test Plan**

315

- Test Management Toolkit Release 1.0 Test Plan 318

Chapter 15**Test Design and Development Examples 331**

Test Management Toolkit Release 1.0 Test Procedure Specification 336

Chapter 16**Example System Test Summary Report 375**

Test Management Toolkit Release 1.0 Test Summary Report 377

Index 385

PART

I

The Rapid Testing Process