

 WILEY

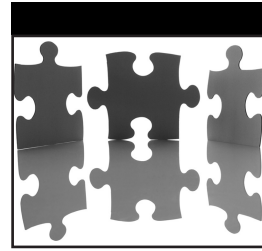
Rex Black

Managing the Testing Process

**Practical Tools and Techniques for Managing
Hardware and Software Testing**



THIRD EDITION



Managing the Testing Process

Practical Tools and Techniques for Managing Software and Hardware Testing

Third Edition

Rex Black



WILEY

Wiley Publishing, Inc.

Managing the Testing Process: Practical Tools and Techniques for Managing Software and Hardware Testing, Third Edition

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2009 by Rex Black. All rights reserved.

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-40415-7

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2009929457

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc. is not associated with any product or vendor mentioned in this book.



About the Author

With a quarter-century of software and systems engineering experience, **Rex Black** is President of RBCS (www.rbcs-us.com), a leader in software, hardware, and systems testing. For more than a dozen years, RBCS has delivered services in consulting, outsourcing, and training for software and hardware testing. Employing the industry's most experienced and recognized consultants, RBCS conducts product testing, builds and improves testing groups, and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation, and more.

As the leader of RBCS, Rex is the most prolific author practicing in the field of software testing today. His popular first book, *Managing the Testing Process*, now in its third edition, has sold more than 30,000 copies around the world, including Japanese, Chinese, and Indian releases. His five other books on testing, *Critical Testing Processes*, *Foundations of Software Testing*, *Pragmatic Software Testing*, *Advanced Software Testing: Volume I*, and *Advanced Software Testing: Volume II*, have also sold tens of thousands of copies, including Hebrew, Indian, Chinese, Japanese, and Russian editions. He has contributed to 10 other books as well. He has written more than 25 articles, presented hundreds of papers, workshops, and seminars, and given about 30 keynote speeches at conferences and events around the world. Rex is a former president of both the International Software Testing Qualifications Board and the American Software Testing Qualifications Board.

When he is not working with clients around the world, developing or presenting a training seminar, or in his office, Rex spends time at home or around the world with his wife and business partner, Laurel Becker; his daughters Emma Grace and Charlotte Catherine; and his faithful canine friends Hank and Cosmo.



Credits

Executive Editor

Robert Elliott

Development Editor

Kelly Talbot

Technical Editor

Judy McKay

Production Editor

Daniel Scribner

Copy Editor

Candace English

Editorial Director

Robyn Siesky

Editorial Manager

Mary Beth Wakefield

Production Manager

Tim Tate

**Vice President and Executive
Group Publisher**

Richard Swadley

**Vice President and Executive
Publisher**

Barry Pruett

Associate Publisher

Jim Minatel

Project Coordinator, Cover

Lynsey Stanford

Proofreader

Nancy C. Hanger / Windhaven

Indexer

Robert Swanson

Cover Image

David Arky / Corbis

Cover Designer

Ryan Sneed



Acknowledgments

This book is a third edition, and that happens only when the first edition and second edition are successful. So, first off, I'd like to thank those of you who bought the first edition or second edition. I hope it's proven a useful reference for you. A special thanks goes to RBCS clients who used these books on their projects, attendees who provided evaluations of RBCS training courses, and readers who wrote reviews or sent me emails about the book. I have addressed your suggestions for improvement in this third edition.

A book gets into people's hands only through a lot of behind-the-scenes hard work by a publisher's team. A special thanks to the fine people at Wiley who helped bring this book to fruition, especially Kelly Talbot and Robert Elliott. RBCS associate Judy McKay provided valuable technical reviewing help. I'd also like to thank Ben Ryan, who shepherded *Managing the Testing Process* along through the first two editions, starting in 1998. I'd also like to thank my friends at Microsoft Press who helped me with the first edition: Erin O'Connor, John Pierce, Mary Renaud, Ben Ryan (again), and Wendy Zucker.

In the course of learning how to manage test projects, I have worked with many talented professionals as a tester, test manager, and consultant. The list of people who helped me is literally too long to include here, but my gratitude to all of my colleagues and clients is immense.

The material in this book appears in one-day, two-day, and three-day test management courses that RBCS associates and I have presented all around the world. I thank all the attendees of those seminars for their help making this material better in the third edition.

Of course, my appreciation goes out to all my past and current colleagues, subcontractors, employees, clients, and employers. I especially want to thank the clients who graciously agreed to the use of data and documentation from their projects to illustrate many of the tools and techniques I discuss.

Four people I want to name specifically in this regard are Judy McKay, Andrew Brooks, Jodi Mullins, and Steven Gaudreau. Judy is a director of quality assurance at large network equipment company. Andrew Brooks is vice president, CA Network and Voice Management Quality Assurance. Jodi Mullins is senior software engineer, CA Network and Voice Management Test Automation. Steven Gaudreau is software engineer, CA Network and Voice Management Test Automation. Each shared specific case studies, authored by them, about topics central to a chapter of each book. I really appreciate their valuable, practitioner insights.

Please attribute all errors, omissions, mistakes, opinions, and bad jokes in this book solely to me.

In the realm of “without whom,” of course, I thank my parents, Rex, Sr. and Carolynn, for their love and support over the years. My greatest appreciation goes to my wife and business partner, Laurel Becker. *Managing the Testing Process* has taken me away from a lot of things in my life, three times now, but I especially appreciate my wife’s support in terms of her own time given up for me.

I’ve changed a few of my ideas since I wrote the first and second editions, but the biggest changes in my life have involved the arrival of my daughters. Along with having a burst of wisdom that led me to marry Laurel, I have to say that Emma Grace and Charlotte Catherine are the greatest things to happen in my life. All parents have dreams for their children’s success, and I hope that my two beautiful and inspirational daughters have the same luck and success in their careers that I have had. Whatever Emma and Charlotte choose to do, this book is dedicated to them, with the utmost of a father’s love.



Contents at a Glance

Introduction		xxiii
Chapter 1	Defining What's on Your Plate: The Foundation of a Test Project	1
Chapter 2	Plotting and Presenting Your Course: The Test Plan	49
Chapter 3	Test-System Architecture, Cases, and Coverage	79
Chapter 4	An Exciting Career in Entomology Awaits You: A Bug-Tracking Database	145
Chapter 5	Managing Test Cases: The Test-Tracking Spreadsheet	199
Chapter 6	Tips and Tools for Crunch Mode: Managing the Dynamic	257
Chapter 7	Stocking and Managing a Test Lab	293
Chapter 8	Staffing and Managing a Test Team	319
Chapter 9	The Triumph of Politics: Organizational Challenges for Test Managers	377
Chapter 10	Involving Other Players: Distributed Testing, outsourcing, and related topics	421
Chapter 11	Economics of Testing: Fiscal Context	475
Chapter 12	Testing Implications of Project and Process: Situational Context	497

Appendix A Hardware-Testing Fundamentals: An Introduction for Software-Testing Professionals	553
Appendix B Omninet: The Internet Everywhere Marketing Requirements Document	567
Appendix C Omninet: The Internet Everywhere System Requirements Document	575
Appendix D Bibliography, Related Readings, and Other Resources	591
Glossary	601
Index	613



Contents

Introduction	xxiii
Chapter 1 Defining What's on Your Plate: The Foundation of a Test Project	1
What You Might Test: The Extended Test Effort	2
From Microscope to Telescope: Test Granularity	2
Structural (White-Box) Tests	2
Behavioral (Black-Box) Tests	3
Live Tests	4
The Complementary and Continuous Nature of Test Granularity	4
A Stampede or a March? Test Phases	4
Unit Testing	5
Component or Subsystem Testing	5
Integration or Product Testing	6
String Testing	7
System Testing	7
Acceptance or User-Acceptance Testing	7
Pilot Testing	8
Why Do I Prefer a Phased Test Approach?	8
Test-Phase Sequencing	9
First Cut	10
What You Should Test: Considering Quality	10
Three Blind Men and an Elephant: Can You Define Quality?	11
The Perils of Divergent Experiences of Quality	12
What to Worry About: How to Analyze Quality Risks	14
Properties and Benefits of Analytical Risk-Based Testing	15
Kick-Starting Quality-Risk Analysis with Checklists	17

Identify and Assess: Process Options for Quality-Risk Analysis	25
Quality-Risk-Analysis Techniques and Templates	29
Tips and Challenges of Quality-Risk Analysis	35
What You Can Test: Schedule, Resources, and Budget	36
Shoehorning: Fitting a Test Schedule into the Project	37
Estimating Resources and Creating a Budget	43
Negotiating a Livable Test Project	45
Case Study	46
Exercise	47

Chapter 2 Plotting and Presenting Your Course: The Test Plan 49

Why I Write Test Plans	49
How Many Test Plans?	50
Using Drafts to Stimulate Discussion	51
Using a Test Plan Template	52
Overview	52
Bounds	53
Scope	53
Definitions	53
Setting	54
Quality Risks	55
Proposed Schedule of Milestones	56
Transitions	56
Entry Criteria	58
Continuation Criteria	59
Exit Criteria	60
Test Development	60
Test Configurations and Environments	61
Test Execution	62
Resources	62
Test-Case and Bug Tracking	64
Bug Isolation and Classification	64
Test Release Management	65
Test Cycles	68
Test Hours	69
Risks and Contingencies	69
Change History	71
Referenced Documents	71
Frequently Asked Questions	71
The IEEE 829 Template: Compare and Contrast	71
Selling the Plan	75

Clarity, Pertinence, and Action	76
Case Study	76
Exercises	77
Chapter 3	79
Test System Architecture, Cases, and Coverage	79
Test System Architecture and Engineering	79
The Action Components: Definitions	82
It's Not Saint Paul's, But. . .Principles for Test System Architecture	84
Not an Objet d'Art: Test System Quality	85
No Test System Is an Island: Testers and the Test System	89
Miscellaneous Best Practices and Principles for Quality Test Systems	90
The Bricks and Mortar of the System: Test Cases	91
Creating Test Conditions	91
A Basic Testing Template	92
A Stress-Test Case for DataRocket	96
Another Useful Test-Case Template	98
IEEE 829 Test System Templates	99
On Test Design and the Smart Use of Templates	104
How Detailed? The Effects of Precision	105
Avoiding the Dreaded Test Escape: Coverage and Regression-Test Gaps	109
Bad Coverage Decisions with the Best Intentions	110
Are You Testing What Development Is Building?	111
Relating Quality Risks to Test Cases	112
Configuration Coverage	113
Bug Coverage	116
Regression-Test Gaps	118
Is Automation a Complete Solution?	120
Four Ways to Spread Tests Across Cycles	122
What if I Can't Repeat All the Tests? Alternative Regression-Risk-Mitigation Strategies	131
"There's a Lesson to Be Learned Here. . .": Test Case Incremental Improvement	133
Responding to Failures	133
Adopting Best Practices	133
Using Reactive Testing	134
You Can't Do It All: Deciding What Not to Do	134
Case Study	135
Bonus Case Study	136

Bonus Case Study	138
Example of Test Specification Levels of Detail	139
Level 1	139
Level 2	139
Level 3	140
Level 4	141
Level 5	142
Exercises	142
Chapter 4 An Exciting Career in Entomology Awaits You: A Bug-Tracking Database	145
Why Bother? The Case for a Formal Bug-Tracking System	146
So, What Seems to Be the Problem? The Failure Description	148
More Like Hemingway than Faulkner	150
Ten Steps to Better Bug Reports	152
Flexible Reporting: Beginning to Construct a Database	154
The Vital Few and the Trivial Many: Ranking Importance	156
Putting the Tracking in Bug Tracking: Adding Dynamic Information	158
Using States to Manage Bug Life Cycles	158
Emphasizing Ownership and Accountability	160
One Key Handoff: Isolation to Debugging	161
Guiding the Bug Life Cycle: The Bug-Triage Process	163
Putting the Dynamic Fields in Place	165
Finishing Touches: Capturing Bug Data for Analysis	167
What the Bug Relates To: Subsystem, Configuration, and Quality Risks	167
Where the Bug Came From: Resolution and Root Cause	169
Functional	171
System	172
Process	172
Data	172
Code	172
Documentation	172
Standards	173
Other	173
Duplicate	173
NAP	173
Bad Unit	173
RCN	173
Unknown	173
How Long Was the Bug Around? Close Date and the Injection, Detection, and Removal Phases	174
The Finalized Bug-Tracking Database	175
The IEEE 829 Standard	176

Extracting Metrics from the Bug-Tracking Database	179
How Defect Removal Proceeds: The Opened/Closed Chart	179
Why Bugs Happen: The Root-Cause Chart	184
How Development Responds: The Closure-Period Chart	184
What Was Broken: The Subsystem Chart	186
An After-the-Fact Metric: Defect-Detection Percentage	188
A Note on Metrics and Charts	189
Managing Bug Tracking	190
Politics and Misuse of Bug Data	190
Don't Fail to Build Trust	190
Don't Be a Backseat Driver	191
Don't Make Individuals Look Bad	192
Sticky Wickets	192
Bug or Feature?	192
Irreproducible Bug	193
Deferring Trivia or Creating Test Escapes?	193
Case Study	194
Exercises	195
Chapter 5	199
Managing Test Cases: The Test-Tracking Spreadsheet	199
Building a Minimalist Test-Tracking Spreadsheet	200
The Basic Spreadsheet	200
Using the Test-Tracking Spreadsheet on Test Projects	203
Making Enhancements	205
Assigning Identifiers and Testers to Test Suites and Cases	205
Adding Date and Hours Information: Plan versus Actual	207
Understanding How Long Tests Run	208
Increasing the Precision of a Test-Case State	208
Prioritizing Test Suites and Cases	212
Scrutinizing the Roll Up Columns	213
Other Ways to Summarize and Group Data	214
Extending the Spreadsheet by Including Test-Case Details	215
Tracking Coverage	216
Putting the Test-Tracking System in Motion	217
Little Trouble	217
Big Trouble	218
No Problem!	221
The IEEE 829 Test Log	221
Extracting Metrics from the Test-Tracking Spreadsheet	223
Can We Get Any Work Done? Charting Test Progress	224
Are We Getting as Much Work Done as Planned? Charting Planned Test Fulfillment	225
Are We Testing What We Promised? Charting Test and Bug Coverage	226

Test Status in a Nutshell: Building a Balanced Scorecard or Dashboard	228
Questioning Dashboards: Dissent and Disputes	229
IEEE 829 Test Reporting: Interim, Level, and Master	232
The Level Interim Status Report	232
The Level Test Report	233
The Master Test Report	235
Case Study One	236
Case Study Two	239
Introduction	239
The Evolution of a Tool	240
Requirements for Moving Forward	241
Design and Implementation	241
Managing Risk	245
Showing the Big Picture and the Details	245
Summary	248
Exercises	249
Chapter 6 Tips and Tools for Crunch Mode: Managing the Dynamic	257
Do Sweat the Details: Staying on Top of Everything	257
Moving Forward While Getting All the Facts: The Desire for Certainty, the Imperative of Progress	258
Dependencies, Schedules, and Reminders: The Importance of Follow-Up	258
It Won't Deliver Itself: Revisions and Release Processes	259
It Won't Install Itself, Either: Configuring the Test Environment	260
"The Hobgoblin of Small Minds" Is Your Friend: Auditing and Updating Test Results	261
Defining a Test-Execution Process	263
Test-Result Misinterpretation: Minimizing False Positives and False Negatives	263
"I Wish You a Merry Dragon-Boat Festival. . .": When Crunch Time, Holidays, and Cultures Collide	266
A Spider's Web of Connections: Managing Test Hardware- and Software-Configuration Logistics	267
The Pieces and How They Connect: An Entity-Relationship Diagram	268
From Diagram to Schemas: Implementing the Logistics Database	271
Budgeting and Planning: Using the Logistics Database Ahead of Time	271
The Work, Who Does It, and Where It Happens: The People Side	272

The Assets, How You Use Them, and Where They Live:	
The Hardware and Infrastructure Side	276
What’s Running Where? Tracking Software Configurations	281
Expect the Unexpected: A Change-Management Database	284
So What? Using (and Misusing) Change-Management Data	285
Simple Is Good: The Change-Management Database	286
Case Study	288
Exercises	290
Chapter 7 Stocking and Managing a Test Lab	293
Do You Need a Test Lab?	294
Selecting and Planning a Lab Area	295
The Test-Lab Inventory	299
A Sample Inventory Template	300
Software	300
Hardware	300
Consumables	301
Furnishings	302
Tools	302
Reference Materials	302
Using Risk Analysis to Pick the Right Inventory	303
Further Thoughts on Stocking Your Lab	304
Security and Tracking Concerns	305
Managing Equipment and Configurations	306
Keeping the Test Environment Clean	309
Human Factors	310
A Safe Lab Is a Productive Lab	310
Damage to Lab Equipment	312
Productivity in the Lab	313
Case Study	314
Exercises	317
Chapter 8 Staffing and Managing a Test Team	319
The Right Person for the Job: What Kind of People Make	
Good Test Engineers	320
Professional Pessimism	320
Balanced Curiosity	321
Focus: No Space Cadets	322
Avoid the Aspiring Hero	323
Shun the Sloth	324
Reject Casper Milquetoast	324
Defining the Test Team: How Many Whos Do What?	325
Size	325
Skills	327

Education, Training, and Certification	331
Positions, Experience, and Goals	336
Specialists or Project Resources? Organizational Models	338
Hiring Testers	341
Defining the Job	341
Gathering and Screening Résumés	345
On-Site Interviews	347
Making the Hiring Decision	350
Avoiding—and Undoing—Hiring Mistakes	351
Bringing the New Tester on Board	352
Giving a Damn: Motivating Your Test Team	353
Be On Your Team’s Side	354
Support a Reasonable Workstyle	356
Foster Career Development for Each Tester	359
Don’t Offer Bonuses Based on Meeting a Schedule	360
Don’t Buy Bugs Like Sacks of Rice	360
Expecting Thanks for Saturday Night’s Pizza	360
Promoting an Us-versus-Them Mentality	361
So, What Are People Actually Doing?	361
Extending Your Talent: Using Temporary Experts and Implementers	362
The Roles Temporary Workers Play	362
Long-Term Temporary Workers	364
Hiring Contractors	368
Bringing on the Experts	372
Case Study	374
Exercises	375
Chapter 9 The Triumph of Politics: Organizational Challenges for Test Managers	377
Don Quixote, Champion of Quality: What’s Your Job, Anyhow?	377
Test Missions and Test Policies	378
Test Team and Manager Titles	381
Where You Fit: The Test Group in the Organization	383
What Else Fits? Adding Other Functions to Test	386
Working with Other Managers: Directions of Test Management	388
Managing Upward	389
Bringing Management to Your Reality: Communicating Clearly	390
“‘How about a Third Shift, and Weekends, and . . .’: The Effects of Lateness on Test	393
Managing Outward	395

Your Partners in Building Quality Systems: Development Peers	395
The Supporting Cast: Internal Service Providers	397
Help Desk, Customer Support, or Technical Support: Often Overlooked	397
Business Analysts, Sales, and Marketing: Vital Allies	398
Testing in the Dark: Should You Proceed without Documentation?	400
Pink Slips: Layoffs and Liquidation	403
Presenting the Results: The Right Message, Delivered Properly	404
Good Ways to Deliver Bad News	405
Institutionalizing a Test Dashboard	406
The Importance of Accuracy and Audience	407
“You Can Tell the Pioneers...”: The Effect of Early Adoption on Test	409
Exercises	412

Chapter 10 Involving Other Players: Distributed Testing, outsourcing, and related topics 421

Choosing Your Partners	422
Your Vendors	424
Testing Service Providers	428
Sales Offices	431
Users and User-Surrogates	432
Planning a Distributed Test Effort	433
Assessing Capabilities	434
Understanding the Cost	435
Collating, Coordinating, and Partitioning the Test Program	436
Organizing Logistics	437
Dealing with Mapping Issues	439
Managing a Distributed Test Effort	442
Monitoring Test Execution	442
Communicating Status and Changing Direction	443
Handling Political Considerations	444
Being Sensitive to Culture Clashes	445
Building and Maintaining Trust	446
How Outsourcing Affects Testing	448
Increased Need for Organization	450
Selecting the Right Test Team	452
Planning and Preparing for Testing in Outsourced Projects	455
Maintaining Focus During Test Execution	459
Conclusions about Outsourcing and Testing	460
Case Study 1	460

Case Study 2	461
Key Differences between Testing Service Providers and In-House Test Teams	461
Test Tasks Appropriate for Testing Service Providers	463
Test Tasks Appropriate for In-House Test Teams	465
Organizational Challenges	466
Processes for Effective Use of Testing Service Providers	469
Bonus Case Study: People Are Not Widgets! Judy Mckay	469
Conclusion	472
Exercises	473
Chapter 11 Economics of Testing: Fiscal Context	475
Is Quality Free? The Economic Justification for the Testing Investment	477
What Does Testing Really Cost?	477
A SpeedyWriter Case Study	478
Management Obstacles to Test Funding	481
Test-Manager Budgeting Faux Pas: Obstacles the Test Manager Creates	481
Regrettable Necessity: Obstacles the Testing Reality Creates	482
Communication Breakdowns: Management Blind Spots and the Difficulty of Education	484
Surmounting the Obstacles . . . Then Doing What We Can	485
Case Study	487
Exercises	488
Chapter 12 Testing Implications of Project and Process: Situational Context	497
Where Testing Fits into the Project Life Cycle	498
Common Life-Cycle Themes	498
The V Model and Sequential Projects	501
The Spiral Model	504
Evolutionary, Incremental, or Agile Models	506
Code and Fix	508
Testing Maintenance Releases	509
System, Subsystem, Commercial Off-the-Shelf Software, and Component Integration	512
Hardware/Software Systems	513
Process Improvement	514
“But We’re Different . . .”: The Commonality of Solutions	514
The Test Team Is Not an Island: External Effects on Your Productivity	517
Process Gas Pedals	518

Process Brake Pedals	521
Project Retrospectives	525
Improving Your Test Process	527
Managing the Testing Process: A Retrospective Conclusion	534
Case Study 1: Agile Testing Challenges	536
Dealing with the Volume and Speed of Change	537
Remaining Effective during Very Short Iterations	537
Receiving Code after Inconsistent (and Often Inadequate) Unit Testing	538
Managing the Increased Regression Risk	539
Making Do with Poor, Changing, and Missing Test Oracles	539
Dealing with a Shifting Test Basis	540
From Detailed Documentation to Many Meetings	541
Holding to Arbitrary Sprint Durations	542
Dealing with Blind Spots in the Sprint Silos	543
Managing Expectations	545
Case Study 2: Maturity and ROI	547
Exercises	549
 Appendix A Hardware Testing Fundamentals: An Introduction for Software Testing Professionals	 553
Test Management	553
Basic Functionality and Self Tests	554
Electrical Testing	554
Environmental Tests	555
Mechanical Life	557
Thermal Tests	558
Reliability	558
Packaging Tests	560
Acoustics	560
Safety	561
Radiation	561
Standards and Regulations	562
Components and Subsystems	563
Integrated Software	564
Supplier Quality Engineering	564
Pilot Testing	565
Case Study	565
 Appendix B Omninet: The Internet Everywhere Marketing Requirements Document	 567
Scope	569
Terms Acronyms and Abbreviations	569
Applicable Documents	570

Required release date	570
Description of requirements	570
General technical requirements	570
Welcome	570
Payment	570
Internet Browser	571
Performance	571
Localization	571
Content Control	571
Session Termination	571
Confidentiality	572
Administration	572
Software Updates	572
View Kiosks	572
View Users	573
Modify User	573
Terminate User	573
Appendix C Omninet: The Internet Everywhere System Requirements Document	575
Functionality System Requirements	576
Reliability System Requirements	580
Usability System Requirements	581
Efficiency System Requirements	582
Maintainability System Requirements	583
Portability System Requirements	584
Design Models	585
Omninet System Architecture	585
Payment Processing Decision Table	586
Kiosk Module Flow	586
Kiosk State-Transition Diagram	588
Kiosk State-Transition Table	588
Kiosk OS/Browser/Connection Speed Configuration	
Orthogonal Array	588
Appendix D Bibliography, Related Readings, and Other Resources	591
Bibliography and Related Readings	591
Online and Hard-Copy Publications	595
Contacting RBCS	595
Our Value	596
Our People	596
Our Clients	596
Consulting	597
Assessments	597
Jump-Starts	598

Project Test Services	598
Test Recruiting, Staff Augmentation, and Outsourcing	599
Training and Certification	599
The Bottom Line	599
Glossary	601
Index	613



Introduction

So, you are responsible for managing a computer hardware or software test project? Congratulations! Maybe you've just moved up from test engineering or moved over from another part of the development team, or maybe you've been doing test projects for a while. Whether you are a test manager, a development manager, a technical or project leader, or an individual contributor with some level of responsibility for your organization's test and quality assurance program, you're probably looking for some ideas on how to manage the unique beast that is a test project.

This book can help you. The first edition, published in 1999, and the second edition, published in 2002, have sold over 35,000 copies in the last decade. There are popular Indian, Chinese, and Japanese editions, too. Clients, colleagues, readers, training attendees, and others have read the book, writing reviews and sometimes sending helpful emails, giving me ideas on how to improve and expand it. So, thanks to all of you who read the first and second editions, and especially to those who have given me ideas on how to make this third edition even better.

This book contains what I wish I had known when I moved from programming and system administration to test management. It shows you how to develop some essential tools and apply them to your test project. It offers techniques that can help you get and use the resources you need to succeed. If you master the basic tools, apply the techniques to manage your resources, and give each area just the right amount of attention, you can survive managing a test project. You'll probably even do a good job, which might make you a test project manager for life, like me.

The Focus of This Book

I've written *Managing the Testing Process* for several reasons. First, many projects suffer from a gap between expectations and reality when it comes to delivery dates, budgets, and quality, especially between the individual contributors creating and testing the software, the senior project managers, and the users and the customers. Similarly, computer hardware development projects often miss key schedule and quality milestones. Effective testing and clear communication of results as an integrated part of a project risk management strategy can help.

Second, when I wrote the first edition, there was a gap in the literature on software and hardware testing. We had books targeting the low-level issues of how to design and implement test cases, as well as books telling sophisticated project managers how to move their products to an advanced level of quality using concepts and tools such as the Capability Maturity Model, software quality metrics, and so forth. However, I believe that test managers like us need a book that addresses the basic tools and techniques, the bricks and mortar, of test project management. While there are now a number of books addressing test management, I believe this book remains unique in terms of its accessibility and immediate applicability to the first-time test manager while also offering guidance in how to incrementally improve a foundational test management approach. It also offers a proven approach that works for projects that include substantial hardware development or integration components.

The tips and tools offered in this book will help you plan, build, and execute a structured test operation. As opposed to the all-too-common ad hoc or purely reactive test project, a structured test operation is planned, repeatable, and documented, but preserves creativity and flexibility in all the right places. What you learn here will allow you to develop models for understanding the meaning of the myriad data points generated by testing so that you can effectively manage what is often a confusing, chaotic, and change-ridden area of a software or hardware development project. This book also shows you how to build an effective and efficient test organization.

To that end, I've chosen to focus on topics unique to test management in the development and maintenance environments. Because they're well covered in other books, I do not address two related topics:

Basic project management tools such as work-breakdown structures, Gantt charts, status reporting, and people management skills. As you move into management, these tools will need to be part of your repertoire, so I encourage you to search out project management books — such as the ones listed in the bibliography in Appendix D — to

help you learn them. A number of excellent training courses and certifications currently exist for project management as well.

Computer hardware production testing. If your purview includes this type of testing, I recommend books by W. Edwards Deming, Kaoru Ishikawa, and J. M. Juran as excellent resources on statistical quality control, as well as Patrick O'Connor's book on reliability engineering; see the bibliography in Appendix D for details on books referenced here.

Software production, in the sense of copying unchanging final versions to distribution media, requires no testing. However, both hardware and software production often include minor revisions and maintenance releases. You can use the techniques described in this book to manage the smaller test projects involved in such releases.

The differences between testing software and hardware are well documented, which might make it appear, at first glance, that this book is headed in two directions. I have found, however, that the differences between these two areas of testing are less important from the perspective of test project management than they are from the perspective of test techniques. This makes sense: hardware tests software, and software tests hardware. Thus, you can use similar techniques to manage test efforts for both hardware and software development projects.

Canon or Cookbook?

When I first started working as a test engineer and test project manager, I was a testing ignoramus. While ignorance is resolvable through education, some of that education is in the school of hard knocks. Ignorance can lead to unawareness that the light you see at the end of the tunnel is actually an oncoming train. "How hard could it be?" I thought. "Testing is just a matter of figuring out what could go wrong, and trying it."

As I soon discovered, however, the flaws in that line of reasoning lie in three key points:

- The tasks involved in "figuring out what could go wrong, and trying it" — that is, in designing good test cases — are quite hard indeed. Many authors have written good books on test case engineering, particularly in the last two decades. Unfortunately, my university professors didn't teach about testing, even though Boris Beizer, Bill Hetzel, and Glenford Myers had all published on the topic prior to or during my college career. As software engineering enters its sixth decade, that has begun to change. However, even at prestigious universities the level of exposure to testing that most software-engineers-in-the-making receive remains too low.

- Testing does not go on in a vacuum. Rather, it is part of an overall project — and thus testing must respond to real project needs, not to the whims of hackers playing around to see what they can break. In short, test projects require test project management.
- The prevalence of the “how hard can testing be” mindset only serves to amplify the difficulties that testing professionals face. Once we’ve learned through painful experience exactly how hard testing can be, it sometimes feels as if we are doomed — like a cross between Sisyphus and Dilbert — to explain, over and over, on project after project, why this testing stuff takes so long and costs so much money.

Implicit in these points are several complicating factors. One of the most important is that the capability of an organization’s test processes can vary considerably: testing can be part of a repeatable, measured process, or an ad hoc afterthought to a chaotic project. In addition, the motivating factors — the reasons why management bothers to test — can differ in both focus and intensity. Managers motivated by fear of repeating a recent failed project see testing differently than managers who want to produce the best possible product, and both motivations differ from those of people who organize test efforts out of obligation but assign them little importance. Finally, testing is tightly connected to the rest of the project, so the test manager is often subject to a variety of outside influences. These influences are not always benign when scope and schedule changes ripple through the project.

These factors make it difficult to develop a *how to* guide for planning and executing a test project. As academics might say, test project management does not lend itself to the easy development of a canon. “Understand the following ideas and you can understand this field” is a difficult statement to apply to test management. And the development of a testing canon is certainly not an undertaking I’ll tackle in this book.

Do you need a canon to manage test projects properly? I think not. Instead, consider this analogy: I am a competent and versatile cook, an amateur chef. I will never appear in the ranks of world-renowned chefs, but I regularly serve passable dinners to my family. I have successfully prepared a number of multicourse Thanksgiving dinners, some in motel kitchenettes. I mastered producing an edible meal for a reasonable cost as a necessity while working my way through college. In doing so, I learned how to read recipes out of a cookbook, apply them to my immediate needs, juggle a few ingredients here and there, handle the timing issues that separate dinner from a sequence of snacks, and play it by ear.

An edible meal at a reasonable cost is a good analogy for what your management wants from your testing organization. This book, then, can serve as a test project manager’s cookbook, describing the basic tools you need and helping you assemble and blend the proper ingredients.

The Tools You Need

Several basic tools underlie my approach to test management:

A solid quality risk analysis. You can't test everything. Therefore, a key challenge to test management is deciding what to test. You need to find the important bugs early in the project. Therefore, a key challenge to test management is sequencing your tests. You sometimes need to drop tests due to schedule pressure. Therefore, a key challenge to test management is test triage in a way that still contains the important risks to system quality. You need to report test results in terms that are meaningful to non-testers. Therefore, a key challenge to test management is tracking and reporting residual levels of risk as test execution continues. Risk based testing, described in this book, will help you do that.

A thorough test plan. A detailed test plan is a crystal ball, allowing you to foresee and prevent potential crises. Such a plan addresses the issues of scope, quality risk management, test strategy, staffing, resources, hardware logistics, configuration management, scheduling, phases, major milestones and phase transitions, and budgeting.

A well-engineered system. Good test systems ferret out, with wicked effectiveness, the bugs that can hurt the product in the market or reduce its acceptance by in-house users. Good test systems mitigate risks to system quality. Good test systems build confidence when the tests finally pass and the bugs get resolved. Good test systems also produce credible, useful, timely information. Good test systems possess internal and external consistency, are easy to learn and use, and build on a set of well-behaved and compatible tools. I use the phrase *good test system architecture* to characterize such a system. The word *architecture* fosters a global, structured outlook on test development within the test team. It also conveys to management that creating a good test system involves developing an artifact of elegant construction, with a certain degree of permanence.

A state-based bug tracking database. In the course of testing, you and your intrepid test team will find lots of bugs, a.k.a. issues, defects, errors, problems, faults, and other less-printable descriptions. Trying to keep all these bugs in your head or in a single document courts immediate disaster because you won't be able to communicate effectively within the test team, with programmers, with other development team peers, or with the project management team — and thus won't be able to contribute to increased product quality. You need a way to track each bug through a series of states on its way to closure. I'll show you how

to set up and use an effective and simple database that accomplishes this purpose. This database can also summarize the bugs in informative charts that tell management about projected test completion, product stability, system turnaround times, troublesome subsystems, and root causes.

A comprehensive test-tracking spreadsheet. In addition to keeping track of bugs, you need to follow the status of each test case. Does the operating system crash when you use a particular piece of hardware? Does saving a file in a certain format take too long? Which release of the software or hardware failed an important test? A simple set of worksheets in a single spreadsheet can track the results of every single test case, giving you the detail you need to answer these kinds of questions. The detailed worksheets also roll up into summary worksheets that show you the big picture. What percentage of the test cases passed? How many test cases are blocked? How long do the test suites really take to run?

A simple change management database. How many times have you wondered, “How did our schedule get so far out of whack?” Little discrepancies such as slips in hardware or software delivery dates, missing features that block test cases, unavailable test resources, and other seemingly minor changes can hurt. When testing runs late, the whole project slips. You can’t prevent test-delaying incidents, but you can keep track of them, which will allow you to bring delays to the attention of your management early and explain the problems effectively. This book presents a simple, efficient database that keeps the crisis of the moment from becoming your next nightmare.

A solid business case for testing. What is the amount of money that testing saves your company? Too few test managers know the answers to this question. However, organizations make tough decisions about the amount of time and effort to invest in any activity based on a cost benefit analysis. I’ll show you how to analyze the testing return on investment, based on solid, well established quality management techniques.

This book shows you how to develop and apply these basic tools to your test project, and how to get and use the resources you need to succeed. I’ve implemented them in the ubiquitous PC-based Microsoft Office suite: Excel, Word, Access, and Project. You can easily use other office-automation applications, as I haven’t used any advanced features.