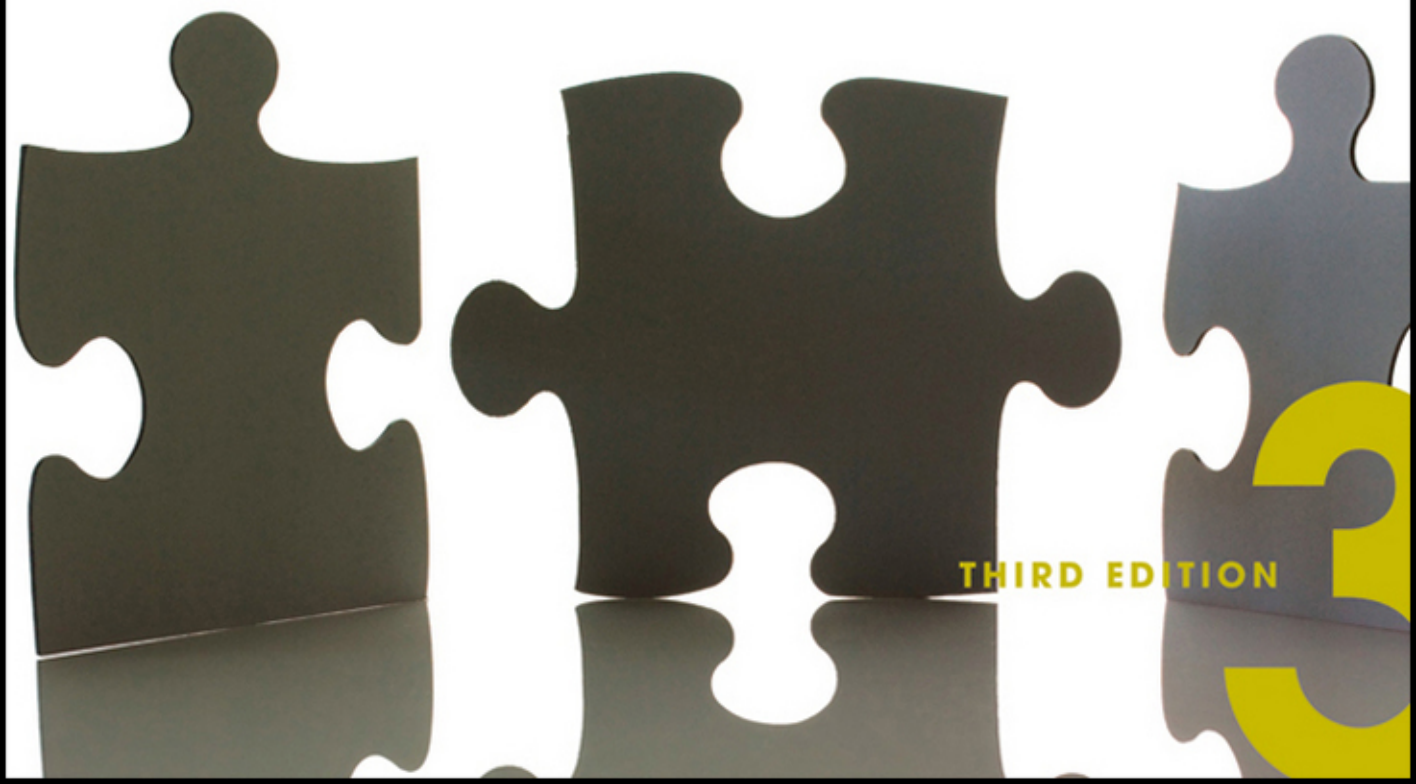
WILEY

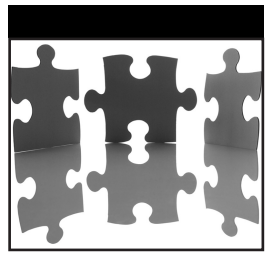
Rex Black

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 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

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

Library of Congress Control Number: 2009929457

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

Contents

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.