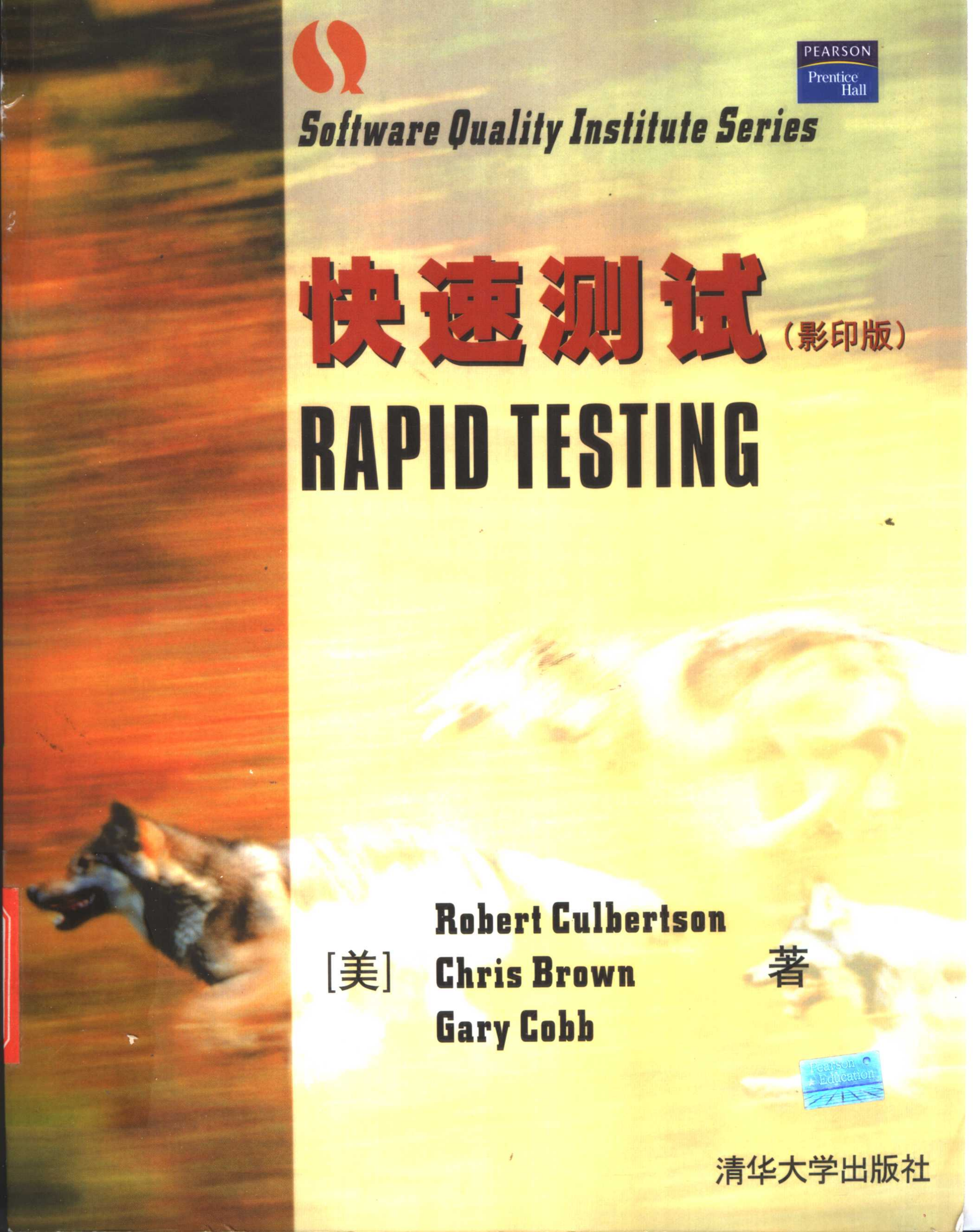

PEARSON
Prentice Hall

Software Quality Institute Series

快速测试（影印版）

RAPID TESTING

[美] Robert Culbertson
Chris Brown
Gary Cobb 著

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

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，情稍候再拨”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，尤其适用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 Berktay, 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

Chapter 4

Chapter 5

PART I

The Rapid Testing Process