

**Testing Applications
on the Web:
Test Planning for Mobile and
Internet-Based Systems**

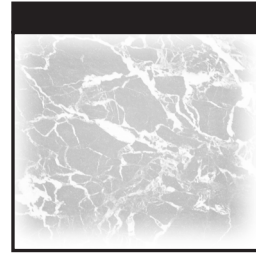
Second Edition

Hung Q. Nguyen
Bob Johnson
Michael Hackett



WILEY

Wiley Publishing, Inc.



**Testing Applications
on the Web:
Test Planning for Mobile and
Internet-Based Systems**

Second Edition

Hung Q. Nguyen
Bob Johnson
Michael Hackett



WILEY

Wiley Publishing, Inc.

Executive Publisher: Robert Ipsen
Executive Editor: Carol Long
Development Editor: Scott Amerman
Editorial Manager: Kathryn A. Malm
Production Editor: Felicia Robinson
Text Design & Composition: Wiley Composition Services

Copyright © 2003 by Hung Q. Nguyen, Bob Johnson, and Michael Hackett. All rights reserved.

Published by Wiley Publishing, Inc., Indianapolis, Indiana
Published simultaneously in Canada

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 Section 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, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8700. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4447, E-mail: permcoordinator@wiley.com.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Trademarks: Wiley, the Wiley Publishing logo and related trade dress are trademarks or registered trademarks of Wiley Publishing, Inc., 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.

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 Cataloging-in-Publication Data:

ISBN: 0-471-20100-6

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To Heather, Wendy, Denny, Leilani, Jesse and Anne, whose love and friendship give me the endless source of energy and happiness.

Hung Q. Nguyen

To Victoria, for all the advice, help, support, and love she has given me.

Bob Johnson

*To Ron, from whom I have stolen much time to make this book happen.
Thank you for your love and support.*

Michael Hackett



Contents

Preface	xxi
Foreword	xxiii
Acknowledgments	xxv
About the Authors	xxvii
Part One Introduction	1
Chapter 1 Welcome to Web Testing	3
Why Read This Chapter?	3
Introduction	4
The Evolution of Software Testing	4
The Gray-Box Testing Approach	7
Real-World Software Testing	9
Themes of This Book	10
What's New in the Second Edition	12
New Contents and Significant Updates	12
What Remains from the First Edition	13
Chapter 2 Web Testing versus Traditional Testing	15
Why Read This Chapter?	15
Introduction	16
The Application Model	16
Hardware and Software Differences	20
The Differences between Web and Traditional	
Client-Server Systems	22
Client-Side Applications	22
Event Handling	23
Application Instance and Windows Handling	26
UI Controls	28

Web Systems	28
Hardware Mix	30
Software Mix	30
Server-Based Applications	31
Distributed Server Configurations	32
The Network	33
Bug Inheritance	33
Back-End Data Accessing	34
Thin-Client versus Thick-Client Processing	35
Interoperability Issues	36
Testing Considerations	37
Bibliography	38
Part Two Methodology and Technology	39
Chapter 3 Software Testing Basics	41
Why Read This Chapter?	41
Introduction	42
Basic Planning and Documentation	42
Common Terminology and Concepts	43
Test Conditions	43
Static Operating Environments	43
Dynamic Operating Environments	44
Test Types	46
Acceptance Testing	46
Feature-Level Testing	50
Phases of Development	58
Test-Case Development	60
Equivalence Class Partitioning and	
Boundary Condition Analysis	60
State Transition	63
Use Cases	66
Example Test Cases from Use Cases	68
Test Cases Built from Use Cases	71
Templates for Use-Case Diagram, Text, and Test Case	75
Condition Combination	75
The Combinatorial Method	78
Bibliography	80
Chapter 4 Networking Basics	81
Why Read This Chapter?	81
Introduction	82
The Basics	82
The Networks	82
The Internet	83
Local Area Networks (LANs)	84
Wide Area Networks (WANs)	85
Connecting Networks	86
Connectivity Services	86

Direct Connection	86
Other Network Connectivity Devices	88
TCP/IP Protocols	89
The TCP/IP Architecture	90
Testing Scenarios	93
Connection Type Testing	94
Connectivity Device Testing	97
Other Useful Information	99
IP Addresses and DNS	99
IP Address	100
Network Classes	100
Domain Name System (DNS)	101
Subnet	103
Subnet Masks	105
Custom Subnets	106
A Testing Example	106
Host Name and IP Resolution Tests	106
Testing Considerations	108
Bibliography	110
Chapter 5	111
Web Application Components	111
Why Read This Chapter?	111
Introduction	112
Overview	112
Distributed Application Architecture	113
Traditional Client-Server Systems	113
Thin- versus Thick-Client Systems	113
Web-Based Client-Server Systems	114
Software Components	116
Operating Systems	117
Application Service Components	117
Third-Party Components	119
Integrated Application Components	119
Dynamic Link Library (DLL)	119
Potential DLL-Related Errors	122
Scripts	123
Web Application Component Architecture	123
Server-Side Components	123
Core Application Service Components	124
Markup Language Pages	125
XML with SOAP	125
Web-to-Database Connectivity	125
Other Application Service Components	128
Client-Side Components	130
Web Browsers	130
Add-on/Plug-in Components	131
Testing Discussion	133
Test-Case Design Analysis	134
Test Partitioning	138

Testing Considerations	141
DLL Testing Issues	142
Script Testing Issues	143
Characteristics of a Script	143
Use of Scripts in Web Applications	144
Testing Scripts in Web Applications	145
Coding-Related Problems	145
Script Configuration Testing	147
Bibliography	147
Chapter 6 Mobile Web Application Platform	149
Why Read This Chapter?	149
Introduction	150
What Is a Mobile Web Application?	150
Various Types of Mobile Web Client	151
Palm-Sized PDA Devices	151
Data Synchronizing	152
Web Connectivity	152
Various Types of Palm-Sized PDA Devices	153
Handheld PCs	154
WAP-Based Phones	155
i-Mode Devices	157
Smart Phones or Mobile Phone/PDA Combos	157
Mobile Web Application Platform	
Test Planning Issues	159
Microbrowsers	159
Web Clipping Application: How Does It Work?	161
Handheld Device Hardware Restrictions	163
Software-Related Issues	164
Wireless Network Issues	166
Wireless Network Standards	166
Wireless Modem	170
Wireless LAN and Bluetooth	170
Other Software Development Platforms and Support Infrastructures	171
The Device Technology Converging Game: Who Is the Winner?	172
Bibliography and Additional Resources	172
Bibliography	172
Additional Resources	173
Chapter 7 Test Planning Fundamentals	177
Why Read This Chapter?	177
Introduction	178
Test Plans	178
Test-Plan Documentation	180
Test-Plan Templates	182
Test-Plan Section Definitions	182

LogiGear One-Page Test Plan	184
Developing a One-Page Test Plan	185
Step 1: Test Task Definition	185
Step 2: Task Completion Time	185
Step 3: Placing the Test Task into Context	186
Step 4: Table Completion	186
Step 5: Resource Estimation	186
Using the LogiGear One-Page Test Plan	187
Testing Considerations	188
Issue Reports	188
Weekly Status Reports	190
Automated Testing	191
Milestone Criteria and Milestone Test	192
Bibliography	192
Chapter 8 Sample Application	193
Why Read This Chapter?	193
Introduction	194
Application Description	194
Technical Overview	195
System Requirements	196
Functionality of the Sample Application	196
Installing the Sample Application	197
Getting Started	197
Division Databases	197
Importing Report Data	197
System Setup	198
Project Setup	198
E-Mail Notification	198
Submitting Defect Reports	198
Generating Metrics	199
Documentation	200
Bibliography	201
Chapter 9 Sample Test Plan	203
Why Read This Chapter?	203
Introduction	204
Gathering Information	204
Step 1: Testing-Task Definitions for the Sample Application	205
Step 2: Task Completion Time	205
Step 3: Placing Test Tasks into the Project Plan	209
Step 4: Calculate Hours and Resource Estimates	210
Sample One-Page Test Plan	210
Bibliography	212

Part Three Testing Practice	213
Chapter 10 User Interface Tests	215
Why Read This Chapter?	215
Introduction	216
User Interface Design Testing	216
Profiling the Target User	217
Computer Experience	217
Web Experience	218
Domain Knowledge	218
Application-Specific Experience	218
Considering the Design	220
Design Approach	221
User Interaction (Data Input)	225
Data Presentation (Data Output)	240
User Interface Implementation Testing	243
Miscellaneous User Interface Elements	243
Display Compatibility Matrix	246
Usability and Accessibility Testing	247
Accessibility Testing	248
Testing Considerations	249
Bibliography and Additional Resources	251
Bibliography	251
Recommended Reading	252
Useful Links	252
Chapter 11 Functional Tests	253
Why Read This Chapter?	253
Introduction	254
An Example of Cataloging Features in Preparation for Functional Tests	254
Testing the Sample Application	254
Testing Methods	257
Functional Acceptance Simple Tests	257
Task-Oriented Functional Tests	258
Forced-Error Tests	259
Boundary Condition Tests and Equivalent Class Analysis	263
Exploratory Testing	264
Software Attacks	265
Which Method Is It?	265
Bibliography	267
Chapter 12 Server-Side Testing	269
Why Read This Chapter?	269
Introduction	270
Common Server-Side Testing Issues	271
Connectivity Issues	271
Time-Out Issues	271
Maintaining State	272

Resource Issues	274
Backup and Restore Issues	275
Fail-over Issues	276
Multithreading Issues	277
Server Side Testing Tips	281
Using Log Files	281
Using Monitoring Tools	284
Creating Test Interfaces or Test Drivers	289
The Testing Environment	291
Working with Live Systems	292
Resetting the Server	292
Using Scripts in Server-Side Testing	293
Bibliography	294
Additional Resources	294
Testing Tools for Run-Time Testing	295
Chapter 13 Using Scripts to Test	297
Why Read This Chapter?	297
Introduction	298
Batch or Shell Commands	298
Batch Files and Shell Scripts	301
Scripting Languages	302
Why Not Just Use a Compiled Program Language?	302
What Should You Script?	303
Application of Scripting to Testing Tasks	303
System Administration: Automating Tasks	303
Discovering Information about the System	304
Testing the Server Directly: Making Server-Side Requests	305
Working with the Application Independent of the UI	306
Examining Data: Log Files and Reports	307
Using Scripts to Understand Test Results	308
Using Scripts to Improve Productivity	309
A Script to Test Many Files	309
A Set of Scripts That Run Many Times	310
Executing Tests That Cannot Be Run Manually	311
Scripting Project Good Practice	311
Scripting Good Practice	312
Resource Lists	313
General Resources for Learning More about Scripting	313
Windows Script Host (WSH)	313
Batch and Shell	314
Perl	314
Tel	315
AWK	315
Learn SQL	315
Where to Find Tools and Download Scripts	316
Bibliography and Useful Reading	316

Chapter 14 Database Tests	317
Why Read This Chapter?	317
Introduction	318
Relational Database Servers	320
Structured Query Language	320
Database Producers and Standards	321
Database Extensions	321
Example of SQL	322
Client/SQL Interfacing	325
Microsoft Approach to CLI	325
Java Approach to CLI	328
Testing Methods	328
Common Types of Errors to Look For	329
Database Stored Procedures and Triggers	333
White-Box Methods	333
Code Walk-through	333
Redundancy Coding Error Example	334
Inefficiency Coding Error Example	334
Executing the SQL Statements One at a Time	336
Executing the Stored Procedures One at a Time	336
Testing Triggers	341
External Interfacing	342
Black-Box Methods	342
Designing Test Cases	342
Testing for Transaction Logic	343
Testing for Concurrency Issues	344
Preparation for Database Testing	345
Setup/Installation Issues	346
Testing with a Clean Database	349
Database Testing Considerations	349
Bibliography and Additional Resources	350
Bibliography	350
Additional Resources	351
Chapter 15 Help Tests	353
Why Read This Chapter?	353
Introduction	354
Help System Analysis	354
Types of Help Systems	354
Application Help Systems	354
Reference Help Systems	355
Tutorial Help Systems	355
Sales and Marketing Help Systems	355
Evaluating the Target User	355
Evaluating the Design Approach	356
Evaluating the Technologies	356
Standard HTML (W3 Standard)	356
Java Applets	357

Netscape NetHelp	358
ActiveX Controls	358
Help Elements	359
Approaching Help Testing	361
Two-Tiered Testing	361
Stand-alone Testing	361
Interaction between the Application and the Help System	361
Types of Help Errors	361
Testing Considerations	365
Bibliography	366
Chapter 16 Installation Tests	367
Why Read This Chapter?	367
Introduction	368
The Roles of Installation/Uninstallation Programs	369
Installer	369
Uninstaller	371
Common Features and Options	372
User Setup Options	372
Installation Sources and Destinations	373
Server Distribution Configurations	373
Server-Side Installation Example	374
Media Types	378
Branching Options	379
Common Server-Side-Specific Installation Issues	384
Installer/Uninstaller Testing Utilities	387
Comparison-Based Testing Tools	387
InControl4 and InControl5	387
Norton Utilities' Registry Tracker and File Compare	387
Testing Considerations	388
Bibliography and Additional Resources	394
Bibliography	394
Additional Resources	394
Chapter 17 Configuration and Compatibility Tests	395
Why Read This Chapter?	395
Introduction	396
The Test Cases	397
Approaching Configuration and Compatibility Testing	398
Considering Target Users	400
When to Run Compatibility and Configuration Testing	400
Potential Outsourcing	401
Comparing Configuration Testing with Compatibility Testing	401
Configuration/Compatibility Testing Issues	403
COTS Products versus Hosted Systems	403
Distributed Server Configurations	404

Client-Side Issues	405
Web Browsers	408
Testing Considerations	411
Bibliography	414
Additional Resources	414
Chapter 18 Web Security Testing	415
Why Read This Chapter?	415
Introduction	416
What Is Computer Security?	417
Security Goals	417
From Which Threats Are We Protecting Ourselves?	418
Common Sources of Security Threats	418
What Is the Potential Damage?	419
Anatomy of an Attack	420
Information Gathering	420
Network Scanning	422
Attacking	423
Attacking Intents	423
Security Solution Basics	424
Strategies, People, and Processes	425
Education	425
Corporate Security Policies	426
Corporate Responses	426
Authentication and Authorization	427
Passwords	427
Authentication between Software Applications or Components	428
Cryptography	428
Other Web Security Technologies	430
Perimeter-Based Security: Firewalls, DMZs, and Intrusion Detection Systems	432
Firewalls	432
Setting Up a DMZ	434
Intrusion Detection Systems (IDS)	435
Common Vulnerabilities and Attacks	435
Software Bugs, Poor Design, and Programming Practice	436
Buffer Overflows	436
Malicious Input Data	439
Command-Line (Shell) Execution	439
Backdoors	440
JavaScript	440
CGI Programs	440
Java	440
ActiveX	441
Cookies	441
Spoofing	442

Malicious Programs	442
Virus and Worm	442
Trojan Horses	442
Misuse Access Privilege Attacks	442
Password Cracking	443
Denial-of-Service Attacks	443
Physical Attacks	444
Exploiting the Trust Computational Base	444
Information Leaks	444
Social Engineering	444
Keystroke Capturing	445
Garbage Rummaging	445
Packet Sniffing	445
Scanning and Probing	445
Network Mapping	445
Network Attacks	445
Testing Goals and Responsibilities	446
Functionality Side Effect: An Error-Handling Bug Example	446
Testing for Security	449
Testing the Requirements and Design	449
Requirements Are Key	449
Trusted Computational Base (TCB)	450
Access Control	450
Which Resources Need to Be Protected?	451
Client Privacy Issues: What Information Needs to Be Private?	451
Testing the Application Code	452
Backdoors	452
Exception Handling and Failure Notification	452
ID and Password Testing	453
Testing for Information Leaks	453
Random Numbers versus Unique Numbers	454
Testing the Use of GET and POST	454
Parameter-Tampering Attacks	455
SQL Injection Attacks	456
Cookie Attacks	456
Testing for Buffer Overflows	458
Testing for Bad Data	459
Reliance on Client-Side Scripting	460
When Input Becomes Output	460
Testing Third-Party Code	461
Known Vulnerabilities	461
Race Conditions	462
Testing the Deployment	462
Installation Defaults	462
Default Passwords	462
Internationalization	462
Program Forensics	463
Working with Customer Support Folks	463

Penetration Testing	463
Testing with User Protection via Browser Settings	465
Testing with Firewalls	468
The Challenges Testers Face	471
Other Testing Considerations	473
Bibliography and Additional Resources	476
Bibliography	476
Additional Resources	477
Useful Net Resources	477
Tools	478
Chapter 19 Performance Testing	479
Why Read This Chapter?	479
Introduction	480
Performance Testing Concepts	481
Determining Acceptable Response Time	
or Acceptable User Experience	481
Response Time Definition	482
Performance and Load Stress Testing Definitions	483
Searching for Answers	484
A Simple Example	485
Performance Testing Key Factors	487
Workload	489
System Environment and Available Resources	489
Response Time	490
Key Factors Affecting Response Time or Performance	492
Three Phases of Performance Testing	493
Setting Goals and Expectations	
and Defining Deliverables	494
Gathering Requirements	496
What Are You Up Against?	496
What If Written Requirements Don't Exist?	496
Defining the Workload	497
Sizing the Workload	498
Server-Based Profile	498
User-Based Profile	501
Problems Concerning Workloads	504
Selecting Performance Metrics	505
Throughput Calculation Example	506
Which Tests to Run and When to Start	508
Tool Options and Generating Loads	512
Tool Options	512
Analyzing and Reporting Collected Data	513
Generating Loads	513
Writing the Test Plan	515
Identifying Baseline Configuration	
and Performance Requirements	515
Determining the Workload	515
Determining When to Begin Testing	515

Determine Whether the Testing Process Will Be Hardware-Intensive or Software-Intensive	516
Developing Test Cases	516
Testing Phase	516
Generating Test Data	517
Setting Up the Test Bed	517
Setting Up the Test Suite Parameters	518
Performance Test Run Example	518
Analysis Phase	520
Other Testing Considerations	523
Bibliography	525
Chapter 20 Testing Mobile Web Applications	527
Why Read This Chapter?	527
Introduction	528
Testing Mobile versus Desktop Web Applications	528
Various Types of Tests	536
Add-on Installation Tests	536
Data Synchronization-Related Tests	536
UI Implementation and Limited Usability Tests	537
UI Guideline References	538
Browser-Specific Tests	539
Platform-Specific Tests	539
Platform or Logo Compliance Tests	540
Configuration and Compatibility Tests	540
Connectivity Tests	541
Devices with Peripheral Network Connections	541
Latency	541
Transmission Errors	542
Transitions from Coverage to No-Coverage Areas	542
Transitions between Data and Voice	542
Data or Message Race Condition	542
Performance Tests	543
Security Tests	544
Testing Web Applications Using an Emulation Environment	544
Testing Web Applications Using the Physical Environment	545
Survey of Mobile Testing Support Tools	546
Device and Browser Emulators	546
Palm Computing	547
OpenWave	547
Nokia	548
YoSpace	548
Microsoft	548
Web-Based Mobile Phone Emulators and WML Validators	548
Desktop WAP Browsers	549

Other Testing Considerations	549
Bibliography and Additional Resources	550
Bibliography	550
Additional Resources	550
Chapter 21 Web Testing Tools	553
Why Read This Chapter?	553
Introduction	554
Types of Tools	554
Rule-Based Analyzers	554
Sample List of Link Checkers and HTML Validators	554
Sample List of Rule-Based Analyzers for C/C++, Java, Visual Basic, and Other Programming and Scripting Languages	556
Load/Performance Testing Tools	557
Web Load and Performance Testing Tools	557
GUI Capture (Recording/Scripting) and Playback Tools	559
Sample List of Automated GUI Functional and Regression Testing Tools	559
Runtime Error Detectors	561
Sample List of Runtime Error-Detection Tools	561
Sample List of Web Security Testing Tools	562
Java-Specific Testing Tools	564
Other Types of Useful Tools	564
Database Testing Tools	564
Defect Management Tool Vendors	565
QACity.Com Comprehensive List of DEFECT TRACKING Tool Vendors	565
Additional Resources	566
On the Internet	566
Development and Testing Tool Mail-Order Catalogs	566
Chapter 22 Finding Additional Information	567
Why Read This Chapter?	567
Introduction	568
Textbooks	568
Web Resources	569
Useful Links	569
Useful Magazines and Newsletters	574
Miscellaneous Papers on the Web from Carnegie Mellon University's Software Engineering Institute	574
Professional Societies	576

Appendix A	LogiGear Test Plan Template	579
Appendix B	Weekly Status Report Template	595
Appendix C	Error Analysis Checklist: Web Error Examples	601
Appendix D	UI Test-Case Design Guideline: Common Keyboard Navigation and Shortcut Matrix	613
Appendix E	UI Test-Case Design Guideline: Mouse Action Matrix	615
Appendix F	Web Test-Case Design Guideline: Input Boundary and Validation Matrix I	617
Appendix G	Display Compatibility Test Matrix	621
Appendix H	Browser OS Configuration Matrix	623
Index		625



Preface

Testing Applications on the Web introduces the essential technologies, testing concepts, and techniques that are associated with browser-based applications. It offers advice pertaining to the testing of business-to-business applications, business-to-end-user applications, Web portals, and other Internet-based applications. The primary audience is software testers, software quality engineers, quality assurance staff, test managers, project managers, IT managers, business and system analysts, and anyone who has the responsibility of planning and managing Web-application test projects.

Testing Applications on the Web begins with an introduction to the client-server and Web system architectures. It offers an in-depth exploration of Web application technologies such as network protocols, component-based architectures, and multiple server types from the testing perspective. It then covers testing practices in the context of various test types from user interface tests to performance, load, and stress tests, and security tests. Chapters 1 and 2 present an overview of Web testing. Chapters 3 through 6 cover methodology and technology basics, including a review of software testing basics, a discussion on networking, an introduction to component-based testing, and an overview of the mobile device platform. Chapters 7 through 9 discuss testing planning fundamentals, a sample application to be used as an application under test (AUT) throughout the book, and a sample test plan. Chapters 10 through 20 discuss test types that can be applied to Web testing. Finally, Chapters 21 and 22 offer a survey of Web testing tools and suggest where to go for additional information.

Testing Applications on the Web answers testing questions such as, “How do networking hardware and software affect applications under test?” “What are Web application components, and how do they affect my testing strategies?”

“What is the role of a back-end database, and how do I test for database-related errors?” “How do I test server-side software?” “What are performance, stress, and load tests, and how do I plan for and execute them?” “What do I need to know about security testing, and what are my testing responsibilities?” “What do I need to consider in testing mobile Web applications?”

With a combination of general testing methodologies and the information contained in this book, you will have the foundation required to achieve these testing goals—maximizing productivity and minimizing quality risks in a Web application environment.

Testing Applications on the Web assumes that you already have a basic understanding of software testing methodologies, including test planning, test-case design, and bug report writing. Web applications are complex systems that involve numerous components: servers, browsers, third-party software and hardware, protocols, connectivity, and much more. This book enables you to apply your existing testing skills to the testing of Web applications.

NOTE This book is not an introduction to software testing. If you are looking for fundamental software testing practices, you will be better served by reading *Testing Computer Software, Second Edition*, by Kaner, Cem, Jack Falk, and Hung Q. Nguyen (Wiley, 1999). For additional information on Web testing and other testing techniques and resources, visit www.QAcity.com.

We have enjoyed writing this book and teaching the Web application testing techniques that we use every day to test Web-based systems. We hope that you will find here the information you need to plan for and execute a successful testing strategy that enables you to deliver high-quality applications in an increasingly distributed-computing, market-driven, and time-constrained environment in this era of new technology.



Foreword

Writing about Web testing is challenging because the field involves the interdependence of so many different technologies and systems. It's not enough to write about the client. Certainly, the client software is the part of the application that is the most visible to the customer, and it's the easiest to write about (authors can just repackage the same old stuff published about applications in general. Hung, Michael, and Bob do provide client-side guidance, but their goal is to provide information that is specific to Web applications. (For more generic material, you can read *Testing Computer Software*, Second Edition, Wiley, 1999.)

But client-side software is just the tip of the iceberg. The application displays itself to the end user as the client, but it does most of its work in conjunction with other software on the server-side, much of it written and maintained by third parties. For example, the application probably stores and retrieves data via third-party databases. If it sells products or services, it probably clears customer orders with the customer's credit card company. It might also check its distributor for available inventory and its shippers for the cost of shipping the software to the customer. The Web application communicates with these third parties through network connections written by third parties. Even the user interface is only partially under the application developer's control—the customer supplies the presentation layer: the browser, the music and video player, and perhaps various other multimedia plug-ins.

The Web application runs on a broader collection of hardware and software platforms than any other type of application in history. Attributes of these platforms can change at any time, entirely outside of the knowledge or control of the Web application developer.

In *Testing Applications on the Web*, Nguyen, Hackett, and Johnson take this complexity seriously. In their view, a competent Web application tester must learn the technical details of the systems with which the application under test interacts. To facilitate this, they survey many of those systems, explaining how applications interact with them and providing testing tips.

As a by-product of helping testers appreciate the complexity of the Web testing problem, the first edition of *Testing Applications on the Web* became the first book on gray-box testing. In so-called black-box testing, we treat the software under test as a black box. We specify the inputs, we look at the outputs, but we can't see inside the box to see how it works. The black-box tester operates at the customer's level, basing tests on knowledge of how the system *should* work. In contrast, the white-box tester knows the internals of the software, and designs tests with direct reference to the program's source code. The gray-box tester doesn't have access to the source code, but he or she knows much more about the underlying architecture and the nature of the interfaces between the application under test and the other software and the operating systems.

The second edition continues the gray-box analysis by deepening the discussions in the first edition. It also adds several new chapters to address business-critical testing issues from server-side, performance- and application-level security testing to the latest mobile Web application testing. A final strength of the book is the power of the real-world example. Hung Quoc Nguyen is the president of the company that published TRACKGEAR, a Web-based bug tracking system, enabling the authors can give us the inside story of its development and testing.

This combination of a thorough and original presentation of a style of analysis, mixed with detailed insider knowledge is a real treat to read. It teaches us about thinking through the issues involved when the software under test interacts in complex ways with many other programs, and it gives the book a value that will last well beyond the specifics of the technologies described therein.

Cem Kaner, J.D., Ph. D.
Professor of Computer Sciences
Florida Institute of Technology



Acknowledgments

While it is our names that appear on the cover, over the years, many people have helped with the development of this book. We want to particularly thank Brian Lawrence, for his dedication in providing thorough reviews and critical feedback. We all thank Cem Kaner for his guidance, friendship, and generosity, and for being there when we needed him. We thank, too, Jesse Watkins-Gibbs for his work on examples and sample code, as well as for his technical expertise and his commitment to getting our book done.

We would also like to thank our professional friends who took time out from their demanding jobs and lives to review and add comment on the book: Yannick Bertolus, George Hamblin, Elisabeth Hendrickson, Nematolah Kashanian, Pat McGee, Alberto Savoia, and Garrin Wong. We would like to thank our copyeditor Janice Borzendowski. We also want to thank the following people for their contributions (listed in alphabetical order): James L. Carr, William Coleman, Norm Hardy, Pam Hardy, Thomas Heinz, Chris Hibbert, Heather Ho, Brian Jones, Denny Nguyen, Kevin Nguyen, Wendy Nguyen, Steve Schuster, Kurt Thams, Anne Tran, Dean Tribble, and Joe Vallejo. Finally, we would like to thank our colleagues, students, and staff at LogiGear Corporation, for their discussions and evaluations of the Web testing training material, which made its way into this book. And thanks to our agent Claudette Moore of Moore Literacy Agency.

Certainly, any remaining errors in the book are ours.



About the Authors

Hung Q. Nguyen is Founder, President, and CEO of LogiGear Corporation. Nguyen has held leadership roles in business management, product development, business development, engineering, quality assurance, software testing, and information technology. Hung is an international speaker and a regular contributor to industry publications. He is the original architect of TRACK-GEAR, a Web-based defect management system. Hung also teaches software testing for the University of California at Berkeley and Santa Cruz Extension, and LogiGear University.

Hung is the author of *Testing Applications on the Web*, First Edition (Wiley 2000); and with Cem Kaner and Jack Falk, he wrote the best-selling book *Testing Computer Software* (ITP/Wiley 1993/1999). He holds a Bachelor of Science in Quality Assurance from Cogswell Polytechnical College, and is an ASQ-Certified Quality Engineer and a member of the Advisory Council for the Department of Applied Computing and Information Systems at UC Berkeley Extension.

You can reach Hung at hungn@logigear.com; or, to obtain more information about LogiGear Corporation and Hung's work, visit www.logigear.com.

Bob Johnson has been a software developer, tester, and manager of both development and testing organizations. With over 20 years of experience in software engineering, Bob has acquired key strengths in building applications on a variety of platforms. Bob's career in software development ranges from Web programming to consulting on legal aspects of e-commerce to the requirement and review process. Whether working in test automation, Web security, or back-end server testing, Bob is at the forefront of emerging technologies.

In addition to participating in the Los Altos Workshops on Software Testing (LAWST), Bob has written articles for *IEEE Software*, *Journal of Electronic Commerce*, and *Software Testing and Quality Engineering*. He can be reached at rjohnson@testingbook.com.

Michael Hackett is Vice President and a founding partner of LogiGear Corporation. He has over a decade of experience in software engineering and the testing of shrink-wrap and Internet-based applications. Michael has helped well-known companies release applications ranging from business productivity to educational multimedia titles, in English as well as a multitude of other languages. Michael has taught software testing for the University of California at Berkeley Extension, the Software Productivity Center in Vancouver, the Hong Kong Productivity Centre, and LogiGear University. Michael holds a Bachelor of Science in Engineering from Carnegie-Mellon University. He can be reached at michaelh@logigear.com.