Effective Methods for Software Testing
Third Edition

William E. Perry

Wiley Publishing, Inc.
Effective Methods for Software Testing
Third Edition

William E. Perry
This book is dedicated to my wife Cynthia, who for many years has been “testing” my ability to live in accordance with our marriage vows. She taught me that testing is a lifelong process, that testing is necessary to ensure that you are meeting your objectives, and that testing can be fun if it is performed correctly. Thank you, Cynthia. What you have taught me is incorporated into many of the concepts in this book.
About the Author

William E. Perry holds degrees from Clarkson University, University of Rochester, and Rochester Institute of Technology. Bill also holds the following professional certifications: CPA (Certified Public Accountant), CIA (Certified Internal Auditor), CISA (Certified Information Systems Auditor), CSQA (Certified Software Quality Analyst), and CSTE (Certified Software Tester). He has been an examiner for the Malcolm Baldrige National Quality Award, and served on standards committees for NIST (National Institute of Standards and Technology), IEEE (Institute of Electrical and Electronics Engineers), AICPA (American Institute of Certified Public Accountants) and ISACA (Information Systems Audit and Control Association).

In 1980, Bill founded the Quality Assurance Institute (QAI), a professional association for testers. QAI offers professional certification for Quality Assurance, Software Testing, Software Project Leaders and Business Analyst Professional. More than 27,000 individuals have been certified since the inception of the program.

Bill has authored more than 50 books, many published by John Wiley & Sons. He recently founded the Internal Control Institute (ICI). ICI and St. Petersburg College recently formed the Internal Control Center of Excellence to share best internal control practices, hold conferences on emerging internal control practices, and to offer e-learning courses and a professional certification in internal control.
Credits

Executive Editor
Robert Elliott

Production Editor
Felicia Robinson

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Joseph B. Wikert

Project Coordinator
Michael Kruzil

Graphics and Production Specialists
Carrie Foster
Mary J. Gillot
Lauren Goddard
Denny Hager
Joyce Haughyey
Stephanie D. Jumper
Rashell Smith

Quality Control Technicians
John Greenough
Brian H. Walls

Proofreading and Indexing
Techbooks
Contents

Introduction xxv

Part I Assessing Testing Capabilities and Competencies 1

Chapter 1 Assessing Capabilities, Staff Competency, and User Satisfaction 3

The Three-Step Process to Becoming a World-Class Testing Organization 3

Step 1: Define a World-Class Software Testing Model 5

Customizing the World-Class Model for Your Organization 7

Step 2: Develop Baselines for Your Organization 8

Assessment 1: Assessing the Test Environment 8

Implementation Procedures 9

Verifying the Assessment 13

Assessment 2: Assessing the Capabilities of Your Existing Test Processes 13

Assessment 3: Assessing the Competency of Your Testers 14

Implementation Procedures 14

Verifying the Assessment 16

Step 3: Develop an Improvement Plan 16

Summary 18

Part II Building a Software Testing Environment 35

Chapter 2 Creating an Environment Supportive of Software Testing 37

Minimizing Risks 38

Risk Appetite for Software Quality 38

Risks Associated with Implementing Specifications 39

Faulty Software Design 39

Data Problems 39

xxv

1

3

3

5

7

8

8

9

13

13

14

14

16

16

18

35

37

38

38

39

39

39

ix
Contents

Defining the Unit Test Plan Standard 83
Converting Testing Strategy to Testing Tactics 83
Process Preparation Checklist 86
Summary 86

Chapter 4 Selecting and Installing Software Testing Tools 103
Integrating Tools into the Tester’s Work Processes 103
Tools Available for Testing Software 104
Selecting and Using Test Tools 108
Matching the Tool to Its Use 109
Selecting a Tool Appropriate to Its Life Cycle Phase 109
Matching the Tool to the Tester’s Skill Level 111
Selecting an Affordable Tool 114
Training Testers in Tool Usage 116
Appointing Tool Managers 117
Prerequisites to Creating a Tool Manager Position 118
Selecting a Tool Manager 118
Assigning the Tool Manager Duties 119
Limiting the Tool Manager’s Tenure 120
Summary 120

Chapter 5 Building Software Tester Competency 125
What Is a Common Body of Knowledge? 125
Who Is Responsible for the Software Tester’s Competency? 126
How Is Personal Competency Used in Job Performance? 126
Using the 2006 CSTE CBOK 127
Developing a Training Curriculum 128
Using the CBOK to Build an Effective Testing Team 129
Summary 131

Part III The Seven-Step Testing Process 151

Chapter 6 Overview of the Software Testing Process 153
Advantages of Following a Process 153
The Cost of Computer Testing 154
Quantifying the Cost of Removing Defects 155
Reducing the Cost of Testing 156
The Seven-Step Software Testing Process 156
Objectives of the Seven-Step Process 159
Customizing the Seven-Step Process 160
Managing the Seven-Step Process 161
Using the Tester’s Workbench with the Seven-Step Process 162
Workbench Skills 163
Summary 164

Chapter 7 Step 1: Organizing for Testing 165
Objective 165
Workbench 166
Input 167
Contents

Do Procedures
Task 1: Appoint the Test Manager 167
Task 2: Define the Scope of Testing 168
Task 3: Appoint the Test Team 168
  Internal Team Approach 169
  External Team Approach 170
  Non-IT Team Approach 170
  Combination Team Approach 170
Task 4: Verify the Development Documentation 171
  Development Phases 171
  Measuring Project Documentation Needs 174
  Determining What Documents Must Be Produced 175
  Determining the Completeness of Individual Documents 179
  Determining Documentation Timeliness 180
Task 5: Validate the Test Estimate and Project Status 181
  Reporting Process 181
  Validating the Test Estimate 182
  Testing the Validity of the Software Cost Estimate 185
  Calculating the Project Status Using a Point System 189
Check Procedures 200
Output 200
Summary 200

Chapter 8 Step 2: Developing the Test Plan 209
Overview 209
Objective 210
Concerns 210
Workbench 211
Input 212
Do Procedures 212
  Task 1: Profile the Software Project 212
    Conducting a Walkthrough of the Customer/User Area 212
    Developing a Profile of the Software Project 213
  Task 2: Understand the Project Risks 215
  Task 3: Select a Testing Technique 222
    Structural System Testing Techniques 223
    Functional System Testing Techniques 229
  Task 4: Plan Unit Testing and Analysis 235
    Functional Testing and Analysis 236
    Structural Testing and Analysis 238
    Error-Oriented Testing and Analysis 240
    Managerial Aspects of Unit Testing and Analysis 243
  Task 5: Build the Test Plan 244
    Setting Test Objectives 245
    Developing a Test Matrix 245
    Defining Test Administration 250
    Writing the Test Plan 251
Chapter 9  Step 3: Verification Testing 291
Overview 292
Objective 293
Concerns 294
Workbench 294
Input 296
The Requirements Phase 296
The Design Phase 296
The Programming Phase 297
Do Procedures 298
Task 1: Test During the Requirements Phase 298
Requirements Phase Test Factors 299
Preparing a Risk Matrix 302
Performing a Test Factor Analysis 310
Conducting a Requirements Walkthrough 312
Performing Requirements Tracing 314
Ensuring Requirements Are Testable 315
Task 2: Test During the Design Phase 316
Scoring Success Factors 316
Analyzing Test Factors 318
Conducting a Design Review 320
Inspecting Design Deliverables 322
Task 3: Test During the Programming Phase 323
Desk Debugging the Program 325
Performing Programming Phase Test Factor Analysis 326
Conducting a Peer Review 328
Check Procedures 330
Output 331
Guidelines 331
Summary 332

Chapter 10  Step 4: Validation Testing 409
Overview 409
Objective 410
Concerns 410
Workbench 410
Input 411
### xiv Contents

<table>
<thead>
<tr>
<th>Do Procedures</th>
<th>412</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1: Build the Test Data</td>
<td>412</td>
</tr>
<tr>
<td>Sources of Test Data/Test Scripts</td>
<td>412</td>
</tr>
<tr>
<td>Testing File Design</td>
<td>413</td>
</tr>
<tr>
<td>Defining Design Goals</td>
<td>414</td>
</tr>
<tr>
<td>Entering Test Data</td>
<td>414</td>
</tr>
<tr>
<td>Applying Test Files Against Programs That Update Master Records</td>
<td>414</td>
</tr>
<tr>
<td>Creating and Using Test Data</td>
<td>415</td>
</tr>
<tr>
<td>Payroll Application Example</td>
<td>416</td>
</tr>
<tr>
<td>Creating Test Data for Stress/Load Testing</td>
<td>430</td>
</tr>
<tr>
<td>Creating Test Scripts</td>
<td>430</td>
</tr>
<tr>
<td>Task 2: Execute Tests</td>
<td>434</td>
</tr>
<tr>
<td>Task 3: Record Test Results</td>
<td>436</td>
</tr>
<tr>
<td>Documenting the Deviation</td>
<td>437</td>
</tr>
<tr>
<td>Documenting the Effect</td>
<td>438</td>
</tr>
<tr>
<td>Documenting the Cause</td>
<td>438</td>
</tr>
<tr>
<td>Check Procedures</td>
<td>439</td>
</tr>
<tr>
<td>Output</td>
<td>439</td>
</tr>
<tr>
<td>Guidelines</td>
<td>439</td>
</tr>
<tr>
<td>Summary</td>
<td>440</td>
</tr>
</tbody>
</table>

### Chapter 11 Step 5: Analyzing and Reporting Test Results | 459 |

| Overview | 459 |
| Concerns | 460 |
| Workbench | 460 |
| Input | 461 |
| Test Plan and Project Plan | 461 |
| Expected Processing Results | 461 |
| Data Collected during Testing | 461 |
| Test Results Data | 462 |
| Test Transactions, Test Suites, and Test Events | 462 |
| Defects | 462 |
| Efficiency | 463 |
| Storing Data Collected During Testing | 463 |
| Do Procedures | 463 |
| Task 1: Report Software Status | 464 |
| Establishing a Measurement Team | 465 |
| Creating an Inventory of Existing Project Measurements | 465 |
| Developing a Consistent Set of Project Metrics | 466 |
| Defining Process Requirements | 466 |
| Developing and Implementing the Process | 466 |
| Monitoring the Process | 466 |
| Task 2: Report Interim Test Results | 470 |
| Function/Test Matrix | 470 |
| Functional Testing Status Report | 471 |
| Functions Working Timeline Report | 472 |
| Expected Versus Actual Defects Uncovered Timeline Report | 472 |
Chapter 12  Step 6: Acceptance and Operational Testing

Overview 491
Objective 492
Concerns 493
Workbench 494
Input Procedures 495
  Task 1: Acceptance Testing 496
    Defining the Acceptance Criteria 497
    Developing an Acceptance Plan 498
    Executing the Acceptance Plan 499
    Developing Test Cases (Use Cases) Based on How Software Will Be Used 500
  Task 2: Pre-Operational Testing 503
    Testing New Software Installation 509
    Testing the Changed Software Version 509
    Monitoring Production 512
    Documenting Problems 513
  Task 3: Post-Operational Testing 513
    Developing and Updating the Test Plan 514
    Developing and Updating the Test Data 515
    Testing the Control Change Process 517
    Conducting Testing 518
    Developing and Updating Training Material 518
Check Procedures 522
Output 522
  Is the Automated Application Acceptable? 522
  Automated Application Segment Failure Notification 523
  Is the Manual Segment Acceptable? 523
  Training Failure Notification Form 524
Guidelines 524
Summary 525
<table>
<thead>
<tr>
<th>Chapter 13</th>
<th>Step 7: Post-Implementation Analysis</th>
<th>571</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>571</td>
<td></td>
</tr>
<tr>
<td>Concerns</td>
<td>572</td>
<td></td>
</tr>
<tr>
<td>Workbench</td>
<td>572</td>
<td></td>
</tr>
<tr>
<td>Input</td>
<td>574</td>
<td></td>
</tr>
<tr>
<td>Do Procedures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 1: Establish Assessment Objectives</td>
<td>574</td>
<td></td>
</tr>
<tr>
<td>Task 2: Identify What to Measure</td>
<td>575</td>
<td></td>
</tr>
<tr>
<td>Task 3: Assign Measurement Responsibility</td>
<td>575</td>
<td></td>
</tr>
<tr>
<td>Task 4: Select Evaluation Approach</td>
<td>575</td>
<td></td>
</tr>
<tr>
<td>Task 5: Identify Needed Facts</td>
<td>576</td>
<td></td>
</tr>
<tr>
<td>Task 6: Collect Evaluation Data</td>
<td>577</td>
<td></td>
</tr>
<tr>
<td>Task 7: Assess the Effectiveness of Testing Using Testing Metrics</td>
<td>577</td>
<td></td>
</tr>
<tr>
<td>Check Procedures</td>
<td>580</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>580</td>
<td></td>
</tr>
<tr>
<td>Guidelines</td>
<td>581</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>581</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part IV</th>
<th>Incorporating Specialized Testing Responsibilities</th>
<th>583</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Chapter 14</th>
<th>Software Development Methodologies</th>
<th>585</th>
</tr>
</thead>
<tbody>
<tr>
<td>How Much Testing Is Enough?</td>
<td>585</td>
<td></td>
</tr>
<tr>
<td>Software Development Methodologies</td>
<td>586</td>
<td></td>
</tr>
<tr>
<td>Overview</td>
<td>586</td>
<td></td>
</tr>
<tr>
<td>Methodology Types</td>
<td>587</td>
<td></td>
</tr>
<tr>
<td>Software Development Life Cycle</td>
<td>588</td>
<td></td>
</tr>
<tr>
<td>Defining Requirements</td>
<td>592</td>
<td></td>
</tr>
<tr>
<td>Categories</td>
<td>592</td>
<td></td>
</tr>
<tr>
<td>Attributes</td>
<td>593</td>
<td></td>
</tr>
<tr>
<td>Methodology Maturity</td>
<td>596</td>
<td></td>
</tr>
<tr>
<td>Competencies Required</td>
<td>598</td>
<td></td>
</tr>
<tr>
<td>Staff Experience</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>Configuration-Management Controls</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>Basic CM Requirements</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>Planning</td>
<td>602</td>
<td></td>
</tr>
<tr>
<td>Data Distribution and Access</td>
<td>602</td>
<td></td>
</tr>
<tr>
<td>CM Administration</td>
<td>602</td>
<td></td>
</tr>
<tr>
<td>Configuration Identification</td>
<td>603</td>
<td></td>
</tr>
<tr>
<td>Configuration Control</td>
<td>605</td>
<td></td>
</tr>
<tr>
<td>Measuring the Impact of the Software Development Process</td>
<td>605</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>606</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 15</th>
<th>Testing Client/Server Systems</th>
<th>611</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>611</td>
<td></td>
</tr>
<tr>
<td>Concerns</td>
<td>612</td>
<td></td>
</tr>
<tr>
<td>Workbench</td>
<td>613</td>
<td></td>
</tr>
<tr>
<td>Input</td>
<td>614</td>
<td></td>
</tr>
</tbody>
</table>
Contents

Do Procedures 614
   Task 1: Assess Readiness 614
      Software Development Process Maturity Levels 615
      Conducting the Client/Server Readiness Assessment 621
      Preparing a Client/Server Readiness Footprint Chart 621
   Task 2: Assess Key Components 622
   Task 3: Assess Client Needs 622
Check Procedures 624
Output 624
Guidelines 624
Summary 624

Chapter 16 Rapid Application Development Testing 633
   Overview 633
   Objective 634
   Concerns 634
      Testing Iterations 634
      Testing Components 635
      Testing Performance 635
      Recording Test Information 635
   Workbench 635
   Input 636
Do Procedures 636
   Testing Within Iterative RAD 636
   Spiral Testing 638
   Task 1: Determine Appropriateness of RAD 639
   Task 2: Test Planning Iterations 640
   Task 3: Test Subsequent Planning Iterations 640
   Task 4: Test the Final Planning Iteration 642
Check Procedures 642
Output 643
Guidelines 643
Summary 643

Chapter 17 Testing Internal Controls 655
   Overview 655
   Internal Controls 657
      Control Objectives 657
      Preventive Controls 658
         Source-Data Authorization 658
         Data Input 659
         Source-Data Preparation 659
         Turnaround Documents 659
         Prenumbered Forms 659
         Input Validation 659
         File Auto-Updating 661
         Processing Controls 661
Detective Controls 662
Data Transmission 663
Control Register 663
Control Totals 664
Documenting and Testing 664
Output Checks 664
Corrective Controls 665
   Error Detection and Resubmission 665
   Audit Trails 665
Cost/Benefit Analysis 666
Assessing Internal Controls 666
   Task 1: Understand the System Being Tested 666
   Task 2: Identify Risks 668
   Task 3: Review Application Controls 668
   Task 4: Test Application Controls 668
      Testing Without Computer Processing 669
      Testing with Computer Processing 669
      Transaction Flow Testing 672
      Objectives of Internal Accounting Controls 673
      Results of Testing 677
   Task 5: Document Control Strengths and Weaknesses 677
Quality Control Checklist 678
Summary 678

Chapter 18 Testing COTS and Contracted Software 685
Overview 686
COTS Software Advantages, Disadvantages, and Risks 686
   COTS Versus Contracted Software 686
   COTS Advantages 687
   COTS Disadvantages 687
   Implementation Risks 688
   Testing COTS Software 689
   Testing Contracted Software 690
Objective 691
Concerns 691
Workbench 692
Input 693
Do Procedures 693
   Task 1: Test Business Fit 693
      Step 1: Testing Needs Specification 693
      Step 2: Testing CSFs 695
   Task 2: Test Operational Fit 696
      Step 1: Test Compatibility 697
      Step 2: Integrate the Software into Existing Work Flows 698
      Step 3: Demonstrate the Software in Action 700
   Task 3: Test People Fit 701
Chapter 19  Testing in a Multiplatform Environment  717
Overview  717
Objective  718
Concerns  718
Background on Testing in a Multiplatform Environment  718
Workbench  719
Input  720
Do Procedures  721
  Task 1: Define Platform Configuration Concerns  721
  Task 2: List Needed Platform Configurations  723
  Task 3: Assess Test Room Configurations  723
  Task 4: List Structural Components Affected by the Platform(s)  723
  Task 5: List Interfaces the Platform Affects  725
  Task 6: Execute the Tests  726
Check Procedures  726
Output  726
Guidelines  726
Summary  727

Chapter 20  Testing Software System Security  733
Overview  733
Objective  734
Concerns  734
Workbench  734
Input  735
Where Vulnerabilities Occur  735
  Functional Vulnerabilities  736
  Vulnerable Areas  737
  Accidental Versus Intentional Losses  738
Do Procedures  739
  Task 1: Establish a Security Baseline  739
    Why Baselines Are Necessary  740
    Creating Baselines  740
    Using Baselines  749
  Task 2: Build a Penetration-Point Matrix  751
    Controlling People by Controlling Activities  751
    Selecting Security Activities  752
    Controlling Business Transactions  755
## Contents

Characteristics of Security Penetration 756  
Building a Penetration-Point Matrix 757  
Task 3: Analyze the Results of Security Testing 760  
Evaluating the Adequacy of Security 761  
Check Procedures 762  
Output 762  
Guidelines 762  
Summary 762  

### Chapter 21 Testing a Data Warehouse 765

Overview 765  
Concerns 765  
Workbench 766  
Input 767  
Do Procedures 768  
Task 1: Measure the Magnitude of Data Warehouse Concerns 768  
Task 2: Identify Data Warehouse Activity Processes to Test 769  
Organizational Process 769  
Data Documentation Process 769  
System Development Process 770  
Access Control Process 771  
Data Integrity Process 771  
Operations Process 772  
Backup/Recovery Process 773  
Performing Task 2 774  
Task 3: Test the Adequacy of Data Warehouse Activity Processes 774  
Check Procedures 780  
Output 780  
Guidelines 780  
Summary 780  

### Chapter 22 Testing Web-Based Systems 799

Overview 799  
Concerns 800  
Workbench 800  
Input 801  
Do Procedures 802  
Task 1: Select Web-Based Risks to Include in the Test Plan 802  
Security Concerns 803  
Performance Concerns 803  
Correctness Concerns 804  
Compatibility Concerns 804  
Reliability Concerns 806  
Data Integrity Concerns 806  
Usability Concerns 806  
Recoverability Concerns 807
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 7: Develop and Execute an Implementation Plan</td>
<td>891</td>
</tr>
<tr>
<td>Planning</td>
<td>891</td>
</tr>
<tr>
<td>Implementing Ideas</td>
<td>891</td>
</tr>
<tr>
<td>Requisite Resources</td>
<td>893</td>
</tr>
<tr>
<td>Quality Control Checklist</td>
<td>894</td>
</tr>
<tr>
<td>Conclusion</td>
<td>894</td>
</tr>
<tr>
<td>Summary</td>
<td>895</td>
</tr>
</tbody>
</table>

## Index

929
Most books about software testing explain “what” to do. This book, on the other hand, takes more of a “how-to” approach. It provides the procedures, templates, checklists, and assessment questionnaires necessary to conduct effective and efficient software testing.

The book is divided into five parts, as follows:

- **Part One: Assessing Testing Capabilities and Competencies.** It is difficult to make any significant change until you know where you are. A baseline tells not only where you are, but lets you measure your progress as your testing strategies and techniques improve. Part One provides three baseline assessments: the capabilities of your software testing group, the competencies of your individual testers, and the effectiveness of your test processes.

- **Part Two: Building a Software Testing Environment.** Software testers are most effective when they work in an environment that encourages and supports well-established testing policies and procedures. The environment includes the procedures and tools for testing, as well as the support and encouragement of management. Part Two begins by describing how to build an environment conducive to testing, and then expands the discussion by describing how to develop a testing process, select testing tools, and build the competency of your testers.

- **Part Three: The Seven-Step Testing Process.** Part Three comprises the core material in the book. It defines a world-class software testing process, from its initiation through testing changes made to operational software systems. This material can be used two ways. First, it contains sufficient procedures and templates so that an organization can use the process as their own. Of course, most organizations inevitably will make some changes to accommodate local vocabulary, specific needs, and customs. This customization process, the seven-step process in this book becomes “owned” by the software testers.
Part Four: Incorporating Specialized Testing Responsibilities. The seven-step testing process is a generic process that almost all software testing organizations can use. However, the mission of software testers may incorporate specialized activities, such as testing security. Rather than incorporating these specialized testing activities directly into the seven-step process, they are presented as individual, specialized activities. As appropriate, they can be incorporated into the seven-step process.

Part Five: Building Agility into the Testing Process. Part Five, which draws on what you’ve learned earlier in the book, is designed to help you identify the strengths and weaknesses of your current software testing process, and then modify it to become more usable or agile.

Getting the Most Out of This Book

This book is not designed to be read like a novel, from beginning to end, nor is it filled with human interest stories about testers. The book focuses on how to conduct software testing. It is designed to help you improve your testing competencies and processes. The self-assessments in Part One will help you identify which parts of the book you need to read first.

The following guidelines will help you maximize the benefit from this book:

– Establish a baseline of current performance. Part One of this book (and Chapter 5) contains four self-assessments for establishing baselines. You need to know where you are so that you can develop a good plan for moving forward.

– Define the software testing organization you would like to have. It has been said that if you do not know where you’re going, all roads lead there. Too many software testing groups just add new testing programs, processes, and tools without knowing if they will integrate effectively.

– Develop a plan for moving from your baseline to your goal. Few organizations can quickly and effectively install an entirely new software testing process. Gradual change is normally much better than radical change. Therefore, identify the gaps between where you are and where you want to be. Determine which of those gaps if closed would provide the greatest benefit to your organization. That becomes the part of the plan you implement first. Over time you will move the entire testing process from your current baseline to your desired goal.

For additional information on software testing conferences and training programs, visit www.taiworldwide.org. For information on software testing certifications, visit www.softwarecertifications.org.

What’s New in the Third Edition

The core of this book is the step-by-step process for testing software. This edition has simplified that process from 11 steps to 7 steps.
A major addition to this edition is the self-assessment in Chapter 5, which testers can use to identify their strengths and weaknesses and then build a personal improvement plan. The self-assessment is based on the Common Body of Knowledge (CBOK) for the Certified Software Tester (CSTE).

Other significant additions include

- A new chapter on testing internal control
- An expanded chapter on testing security
- A new chapter on adapting testing to the developmental methodology used to build the software
- Two new chapters on how to incorporate agile methods into the testing process

**What’s on the CD**

This book includes a CD that contains the work papers and quality control checklists to help you implement the software testing process.

To use the CD, first you need to select a software testing activity that you want to implement in your organization—for example, test planning. Then, from the chapter on test planning, identify those work papers and checklists that you believe would be beneficial to your organization. You can extract those work papers and checklists from the CD and begin a customization process. For example, you can include the name of your organization, add or delete portions of the work papers, and change the terminology to be consistent with your organization.

After you have used the work papers for conducting a software test, you should bundle the work papers into a case study for new testers. If they use the book to learn the basics of software testing and then can cross reference what they have learned to examples of how the work papers are actually used in software testing, learning should be accelerated.