

EFFECTIVE METHODS
FOR SOFTWARE TESTING

Effective Methods for Software Testing

William Perry



A Wiley-QED Publication

John Wiley & Sons, Inc.

New York • Chichester • Brisbane • Toronto • Singapore

Publisher: Katherine Schowalter
Editor: Robert Elliott
Managing Editor: Maureen B. Drexel
Text Design & Composition: Publishers' Design and Production Services, Inc.

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where John Wiley & Sons, Inc. is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

This text is printed on acid-free paper.

Copyright © 1995 by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional service. If legal advice or other expert assistance is required, the services of a competent professional person should be sought.

Reproduction or translation of any part of this work beyond that permitted by section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

Library of Congress Cataloging-in-Publication Data

Perry, William E.
Effective methods for software testing / William Perry.
p. cm.
Includes index.
ISBN 0-471-06097-6 (cloth : alk. paper)
1. Computer software—Testing. I. Title.
QA 76.76.T48P46 1995
005.1'4—dc20

95-16910
CIP

Printed in the United States of America
10 9 8 7 6 5 4

Effective Methods for Software Testing

Preface

Testing is an unnecessary and unproductive activity if its sole purpose is to validate that the specifications were implemented as written. If the developmental processes for software work correctly, they would implement those specifications as written. Thus, testing as performed in most organizations is a process designed to compensate for an ineffective software development process.

It is unrealistic to develop software and not test it. The perfect development process does not exist, and the probability of it existing in the foreseeable future is close to nil. As long as the developmental and maintenance process continues to introduce defects into software, testing will be a very important component of the developmental process. Few users would have confidence in using software if its functions were not validated.

If the assumption that testing will occur is true, then the process should be designed to provide as much value as possible. This can be accomplished by using tests to reduce business risk, not only testing that specifications are implemented as documented, but that those documented specifications are what the business needs, and that the software and its associated manual procedures provide adequate assurance that the associated business risks are reduced to an acceptable level. The objective of this book is to provide a test strategy and tactics based on those assumptions.

As a system analyst and computer programmer, I never found computer testing to be an enjoyable part of my job function. My disenchantment was due partially to the detailed and repetitive nature of testing. However, my real concern over testing dealt with the fact that when I had completed the test process the effectiveness of my work was in doubt. There was no methodology to provide the needed assurance that my system was devoid of defects.

In the last few years, there has been extensive research on the methods of testing as well as testing tools and techniques. The concept of testing has grown from an after-programming evaluation process to a concept that is an integral part of each phase of the system development life cycle. Testing is no longer an adjunct to the system development life cycle (SDLC), but rather, is a key part of it.

The approach outlined in this book will increase the time and effort most organizations expend on testing. The payback comes from detecting problems earlier in the system development life cycle in order to avoid designing and coding the system incorrectly and then correcting those defects. Experience by those organizations using life cycle testing indicates that while the test cost increases, the net cost to develop the system decreases significantly. These concepts work.

Contents

	Preface	xv
Part 1	Developing a Test Approach	1
1	Addressing the Software System Business Risk	3
	What Is Software Testing?	3
	Who Is Associated with Testing?	4
	The Multiple Roles of Testing	4
	What Is a Defect?	5
	<i>Defects versus Failures</i>	5
	<i>Process Problems and Defect Rates</i>	6
	The Business Perspective for Testing	6
	How Good Is Your Existing Testing Process?	7
	Interpretation of Assessment	7
	How to Use This Book to Improve Your Test Process	7
	<i>Based on the Self-Assessment</i>	7
	<i>Based on Software Test Responsibilities</i>	9
2	Defining a Software System Testing Strategy	13
	Computer System Strategic Risks	13
	The Economics of Testing	14
	Common Computer Problems	15
	<i>Software Problems</i>	16
	<i>Data Problems</i>	16
	Economics of System Development Life Cycle (SDLC) Testing	16
	Testing—An Organizational Issue	18

Establishing a Testing Policy	18
<i>Methods</i>	19
The Structured Approach to Testing	20
<i>Requirements</i>	22
<i>Design</i>	22
<i>Program (Build / Construction)</i>	22
<i>Test</i>	23
<i>Installation</i>	23
<i>Maintenance</i>	23
Test Strategy	23
<i>Test Factors</i>	25
<i>Developing a Test Strategy</i>	27
Testing Methodology	29
Status of Software Testing	29

3 Developing Software System Testing Tactics **30**

The Workbench Concept	30
Testing Definitions and Concepts	31
The Eight Steps to Develop Testing Tactics	33
1. <i>Acquire and Study Test Strategy</i>	33
2. <i>Determine the Type of Development Project</i>	34
3. <i>Determine the Type of Software System</i>	34
4. <i>Determine the Project Scope</i>	35
5. <i>Identify Tactical Risks</i>	36
6. <i>Determine When Testing Should Occur</i>	37
7. <i>Build a System Test Plan</i>	40
8. <i>Build the Unit Test Plan</i>	41
How Much Time Should Be Spent on Tactics?	41

Part 2 Testing Software Using a Life Cycle Methodology **53**

4 Life Cycle Testing Approach **55**

The Cost of Computer Testing	55
<i>Quantifying the Cost of Removing Defects</i>	56
<i>Reducing the Cost of Testing</i>	58
Life Cycle Testing Concept	58
Composition of the Test Team	58
<i>Internal IS Test Team Approach</i>	59
<i>External IS Test Team Approach</i>	60
<i>Non-IS Test Team Approach</i>	60
<i>Combination Test Team Approach</i>	60
Testing Concerns	61
Responsibility for Establishing a Test Plan	65

5 Requirements Phase Testing **66**

Deliverables	66
Test Concerns	67
<i>Requirements Comply with Methodology (Methodology Test Factor)</i> ...	67
<i>Functional Specifications Defined (Correctness Test Factor)</i>	67

	<i>Usability Specifications Determined (Ease of Use Test Factor)</i>	67
	<i>Maintenance Specifications Determined (Maintainable Test Factor)</i> . . .	67
	<i>Portability Needs Determined (Portable Test Factor)</i>	68
	<i>System Interface Defined (Coupling Test Factor)</i>	68
	<i>Performance Criteria Established (Performance Test Factor)</i>	68
	<i>Operational Needs Defined (Ease of Operations Test Factor)</i>	68
	<i>Tolerances Established (Reliability Test Factor)</i>	68
	<i>Authorization Rules Defined (Authorization Test Factor)</i>	68
	<i>File Integrity Requirements Defined (File Integrity Test Factor)</i>	69
	<i>Reconstruction Requirements Defined (Audit Trail Test Factor)</i>	69
	<i>Impact of Failure Defined (Continuity of Processing Test Factor)</i>	69
	<i>Desired Service Level Defined (Service Level Test Factor)</i>	69
	<i>Access Defined (Security Test Factor)</i>	69
	Test Responsibilities	69
	Recommended Test Tools	70
	<i>Walk-Through Test Tool</i>	70
	<i>Risk Matrix Test Tool</i>	72
	Requirements Phase Test Process	81
	<i>Test Programs</i>	82
6	Design Phase Testing	99
	Deliverables	99
	Test Concerns	100
	<i>Data Integrity Controls Designed</i>	100
	<i>Authorization Rules Designed</i>	100
	<i>File Integrity Controls Designed</i>	100
	<i>Audit Trail Designed</i>	100
	<i>Contingency Plan Designed</i>	101
	<i>Method to Achieve Service Level Designed</i>	101
	<i>Access Procedures Defined</i>	101
	<i>Design Complies with Methodology</i>	101
	<i>Design Conforms to Requirements</i>	101
	<i>Design Facilitates Use</i>	101
	<i>Design Is Maintainable</i>	101
	<i>Design Is Portable</i>	102
	<i>Interface Design Complete</i>	102
	<i>Design Achieves Criteria</i>	102
	<i>Needs Communicated to Operations</i>	102
	Test Responsibilities	102
	Recommended Test Tools	102
	<i>Scoring Test Tool</i>	103
	<i>Design Review Test Tool</i>	105
	Test Process	106
7	Program Phase Testing	151
	Deliverables	151
	Test Concerns	152
	<i>Data Integrity Controls Implemented</i>	152
	<i>Authorization Rules Implemented</i>	152
	<i>File Integrity Controls Implemented</i>	152

	<i>Audit Trail Implemented</i>	153
	<i>Contingency Plan Written</i>	153
	<i>System to Achieve Service Level Designed</i>	153
	<i>Security Procedures Implemented</i>	153
	<i>Program Complies with Methodology</i>	153
	<i>Program Conforms to Design (Correctness)</i>	153
	<i>Program Conforms to Design (Ease of Use)</i>	153
	<i>Program Is Maintainable</i>	154
	<i>Program Conforms to Design (Portability)</i>	154
	<i>Program Conforms to Design (Coupling)</i>	154
	<i>Operating Procedures Developed</i>	154
	<i>Program Achieves Criteria (Performance)</i>	154
	Test Responsibilities	154
	Recommended Test Tools	155
	<i>Desk Debugging</i>	155
	<i>Program Peer Review</i>	156
	Test Process	157
8	Evaluating Test Results	177
	Deliverables	177
	Types of Testing	178
	<i>Manual, Regression, and Functional Testing (Reliability)</i>	178
	<i>Compliance Testing (Authorization)</i>	178
	<i>Functional Testing (File Integrity)</i>	178
	<i>Functional Testing (Audit Trail)</i>	178
	<i>Recovery Testing (Continuity of Testing)</i>	179
	<i>Stress Testing (Service Level)</i>	179
	<i>Compliance Testing (Security)</i>	179
	<i>Testing Complies with Methodology</i>	179
	<i>Functional Testing (Correctness)</i>	179
	<i>Manual Support and Testing (Ease of Use)</i>	179
	<i>Inspections (Maintainability)</i>	179
	<i>Disaster Testing (Portability)</i>	180
	<i>Functional and Regression Testing (Coupling)</i>	180
	<i>Compliance Testing (Performance)</i>	180
	<i>Operations Testing (Ease of Operations)</i>	180
	Responsibilities	180
	Recommended Tools	181
	<i>Test Data Test Tool</i>	181
	<i>Volume Test Test Tool</i>	185
	Test Process	195
9	Installation Phase Testing	213
	Deliverables	213
	Test Concerns	214
	<i>Accuracy and Completeness of Installation Verified (Reliability)</i>	214
	<i>Data Changes during Installation Prohibited (Authorization)</i>	215
	<i>Integrity of Production Files Verified</i>	215
	<i>Installation Audit Trail Recorded</i>	215

	<i>Integrity of Previous System Assured (Continuity of Processing)</i>	215
	<i>Fail-safe Installation Plan Implemented (Service Level)</i>	215
	<i>Access Controlled during Installation (Security)</i>	215
	<i>Installation Complies with Methodology</i>	216
	<i>Proper Programs and Data Placed into Production</i>	216
	<i>Usability Instructions Disseminated</i>	216
	<i>Documentation Complete (Maintainability)</i>	216
	<i>Documentation Complete (Portability)</i>	216
	<i>Interface Coordinated (Coupling)</i>	216
	<i>Integration Performance Monitored</i>	216
	<i>Operating Procedures Implemented</i>	217
	Responsibilities	217
	Recommended Test Tools	217
	<i>Checklist Test Tool</i>	218
	<i>Acceptance Test Criteria Test Tool</i>	219
	Test Process	220
10	Acceptance Testing	238
	What Is Software Acceptance?	238
	Software Acceptance Process	239
	1. <i>Define the Buyer's Role</i>	239
	2. <i>Define the Acceptance Criteria</i>	240
	3. <i>Develop an Acceptance Plan</i>	242
	4. <i>Execute the Acceptance Plan (Conduct Acceptance Tests</i> <i>and Reviews)</i>	242
	5. <i>Develop the Acceptance Decision</i>	243
Part 3	Testing Changes to Software Systems (Software Maintenance)	245
11	Testing Methodology for Software Maintenance	247
	Objectives of Testing and Training	247
	Control Concerns	248
	Steps Involved in Testing and Training	248
	1A. <i>Develop Test Plan</i>	249
	1B. <i>Develop Training Plan</i>	250
	2A. <i>Prepare Test Data</i>	251
	2B. <i>Prepare Training Material</i>	252
	3A. <i>Conduct Acceptance Training</i>	252
	3B. <i>Conduct Training</i>	253
	4A. <i>Automated Application Acceptable?</i>	253
	4B. <i>Manual Segment Acceptable?</i>	254
	5A. <i>Making Test Adjustments</i>	255
	5B. <i>Making Training Adjustments</i>	255
	Software Change Feedback	255
	Software Change Review Checklist	257
12	Testing the Correctness of Installing a Software Change	278
	Objectives of Installing the Change	278
	Change Control Concerns	279

Steps Involved in Testing the Change Installation	280
1. <i>Develop the Restart/Recovery Plan</i>	280
2. <i>Enter the Change into Production</i>	281
3. <i>Delete Any Unneeded Versions</i>	282
4. <i>Operate the Application</i>	282
5. <i>Monitor Production</i>	282
6. <i>Document Problems</i>	283
Software Change Feedback	283
Software Change Review Checklist	285

Part 4 Testing Methods, Tools, and Techniques **303**

13 How Testers Can Help Mangle the Software Development **305**

Testing the Validity of a Software Cost Estimate	305
<i>Test Objective</i>	305
<i>Why Software Cost Estimating Is Error Prone</i>	306
<i>Strategies for Software Cost Estimating</i>	306
<i>Types of Parametric Models</i>	307
<i>General Pattern Followed by the Models</i>	307
<i>Testing the Validity of the Software Cost Estimate</i>	309
Discussion of the Influencing Factors	310
<i>Project-Specific Factors</i>	310
<i>Organization-Dependent Factors</i>	312
<i>Verify the Correctness of the Cost-Estimating Model Estimate</i>	314
Testing the Progress of the Software System	314
<i>Overview of the Point Accumulation Tracking System</i>	315
<i>Typical Methods of Measuring Performance</i>	315
<i>How to Use the Point System</i>	317
<i>Extensions</i>	318
<i>Rolling Baseline</i>	320
<i>Reports</i>	321
<i>Using the Point System as a Test Method</i>	321

14 Inspecting Test Plans and Test Cases **330**

How to Conduct Inspections	330
Inspecting Test Plans	330
<i>Definitions</i>	330
<i>Objective</i>	332
Test Plan Description	332
<i>Test Plan Process</i>	333
<i>Participants</i>	333
<i>Preparation</i>	333
<i>Materials</i>	333
<i>Inspection</i>	334
<i>Rework</i>	334
<i>Follow-up</i>	334
Inspecting Test Cases	334
<i>Test Case Description</i>	334
<i>Test Case Process</i>	335

	<i>Participants</i>	335
	<i>Preparation</i>	335
	<i>Materials</i>	335
	<i>Inspection</i>	336
	<i>Rework</i>	336
	<i>Follow-up</i>	336
	<i>Form Completion Instructions: Work Paper 14.1 Inspection Report</i> . . .	336
	<i>Form Completion Instructions: Work Paper 14.2 Test Plan</i>	
	<i>Inspection Report</i>	337
	<i>Form Completion Instructions: Work Paper 14.3 Test Case Inspection</i> .	337
15	Assessing Client-Server and LAN Risks	341
	Client-Server Readiness	341
	Responsibility for Readiness	341
	Readiness Dimensions	341
	Software Development Process Maturity Levels	342
	<i>The Ad Hoc Process (Level 1)</i>	343
	<i>The Repeatable Process (Level 2)</i>	344
	<i>The Consistent Process (Level 3)</i>	345
	<i>The Measured Process (Level 4)</i>	345
	<i>The Optimized Process (Level 5)</i>	346
	Conducting the Client-Server Readiness Assessment	347
	Preparing a Client-Server Readiness Kiviat Chart	347
16	A Testing Strategy for Rapid Prototyping	350
	Prototype-Based Testing	350
	<i>Test Information from Iteration</i>	351
	<i>Test Information about Components</i>	351
	<i>Test Information about Performance</i>	351
	<i>Recording Test Information</i>	351
	Overview of a Testing Support System	352
	<i>Shimeall's Testing Within Iterative Rapid Prototyping</i>	352
	Spiral Testing	352
	<i>The First Test Planning Iterations</i>	353
	<i>Subsequent Test Planning Iterations</i>	354
	<i>The Final Test Planning Iteration</i>	355
	Spiral Testing Summary: Advantages and Disadvantages	356
17	Testing Techniques	357
	The Concept of Application Fit	357
	Testing Techniques/Tool Selection Process	358
	<i>Structural versus Functional Testing</i>	358
	<i>Dynamic versus Static Testing</i>	359
	<i>Manual versus Automated Tests</i>	360
	<i>Selecting Techniques/Tools Flowcharts</i>	360
	Difference between Testing Techniques and Tools	361
	Structural System Testing Techniques	361
	<i>Stress Testing Techniques</i>	361

	<i>Execution Testing Technique</i>	363
	<i>Recovery Testing Technique</i>	364
	<i>Operations Testing Technique</i>	365
	<i>Compliance Testing Technique</i>	366
	<i>Security Testing Technique</i>	366
	Functional System Testing Techniques	367
	<i>Requirements Testing Technique</i>	368
	<i>Regression Testing Technique</i>	369
	<i>Error-Handling Testing Technique</i>	370
	<i>Manual-Support Testing Techniques</i>	371
	<i>Intersystem Testing Technique</i>	372
	<i>Control Testing Technique</i>	373
	<i>Parallel Testing Techniques</i>	374
	Unit Testing Technique	374
	Functional Testing and Analysis	375
	Functional Analysis	375
	Functional Testing	375
	<i>Testing Independent of the Specification Technique</i>	376
	<i>Testing Dependent on the Specification Technique</i>	377
	<i>Structural Testing and Analysis</i>	377
	<i>Error-Oriented Testing and Analysis</i>	379
	<i>Managerial Aspects of Unit Testing and Analysis</i>	381
	Test Factor/Test Technique Matrix	382
18	Testing Tools	384
	Testing Tools—The Hammers of Testing	384
	<i>Overview</i>	384
	Selecting and Using the Test Tools	391
	1. <i>Matching the Tool to Its Use</i>	391
	2. <i>Selecting a Tool Appropriate to the Life Cycle Phase</i>	392
	3. <i>Matching the Tool to the Skill Level of the Tester</i>	392
	4. <i>Selecting an Affordable Tool</i>	397
	<i>What if You Finish Step 4 and All Tools Are Excluded?</i>	401
19	Test Documentation	404
	Preparation	404
	Uses	404
	Types	405
	Responsibility	405
	Storage	406
	Test Plan Documentation	406
	<i>Section 1—General Information</i>	406
	<i>Section 2—Plan</i>	406
	<i>Section 3—Specifications and Evaluation</i>	411
	<i>Section 4—Test Descriptions</i>	411
	Test Analysis Report Documentation	412
	<i>Section 1—General Information</i>	414
	<i>Section 2—Test Results and Findings</i>	415
	<i>Section 3—Software Function Findings</i>	415
	<i>Section 4—Analysis Summary</i>	417

Part 5	Evaluating and Reporting Test Results	417
20	Reporting Test Results	419
	What Data Does Testing Produce?	419
	<i>Test Results Data</i>	419
	<i>Test Transactions, Test Suites, Test Events</i>	420
	<i>Defects</i>	420
	<i>Efficiency</i>	420
	<i>Interim Reporting</i>	421
	<i>Functional Testing Status Report</i>	422
	<i>Functions Working Timeline</i>	422
	<i>Expected versus Actual Defects Uncovered Timeline</i>	423
	<i>Defects Uncovered versus Corrected Gap Timeline</i>	424
	<i>Average Age Uncorrected Defects by Type</i>	424
	<i>Defect Distribution Report</i>	426
	<i>Relative Defect Distribution Report</i>	426
	<i>Testing Action Report</i>	427
	Final Test Reporting	428
	<i>Unit Test Reports</i>	429
	<i>Integration Test Report</i>	429
	<i>System Test Report</i>	430
	<i>Acceptance Test Report</i>	430
21	Evaluating Test Effectiveness	434
	Objectives	434
	The Test Measurement Process	434
	<i>Step 1—Establish Assessment Objectives</i>	435
	<i>Step 2—Identify What to Measure</i>	436
	<i>Step 3—Assign Measurement Responsibility</i>	436
	<i>Step 4—Select Evaluation Approach</i>	436
	<i>Step 5—Identify Needed Facts</i>	437
	<i>Step 6—Collect Evaluation Data</i>	437
	<i>Step 7—Assess the Effectiveness of Testing</i>	437
	Use of Testing Metrics	437
	Improving the Test Process	441
Appendices		443
A	Testing Tools	445
B	Testing Metrics	489
C	Software Testing Survey Report	520
	Glossary	529
	Bibliography	533
	Index	535

**Developing a
Test Approach**

Addressing the Software System Business Risk

WHAT IS SOFTWARE TESTING?

The software development effort is a process. The process cycle is comprised of the following four components (see Figure 1.1):

- **Plan (P): Devise a plan.** Define your objective and determine the strategy and supporting methods required to achieve that objective. The plan should be based on an assessment of your current situation, and the strategy should clearly focus upon the strategic initiatives/key units that will drive your improvement plan. Express a specific objective numerically. Determine the procedures and conditions for the means and methods you will use to achieve the objective.
- **Do (D): Execute the plan.** Create the conditions and perform the necessary training to execute the plan. Make sure everyone thoroughly understands the objectives and the plan. Teach workers the procedures and skills they need to fulfill the plan and thoroughly understand the job. Then perform the work according to these procedures.

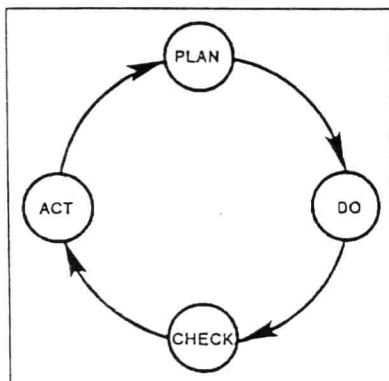


Figure 1.1 The continuous improvement cycle.