

Contents

SECTION I	SOFTWARE QUALITY IN PERSPECTIVE	1
1	Quality Assurance Framework	5
	What Is Quality?	5
	Prevention versus Detection	6
	Verification versus Validation	7
	Software Quality Assurance	8
	Components of Quality Assurance	9
	Software Testing	10
	Quality Control	11
	Software Configuration Management	12
	Elements of Software Configuration Management	12
	Component Identification	13
	Version Control	14
	Configuration Building	14
	Change Control	15
	Software Quality Assurance Plan	16
	Steps to Develop and Implement a Software Quality Assurance Plan	16
	Step 1. Document the Plan	16
	Step 2. Obtain Management Acceptance	18
	Step 3. Obtain Development Acceptance	18
	Step 4. Plan for Implementation of the SQA Plan	19
	Step 5. Execute the SQA Plan	19
	Quality Standards	19
	ISO9000	19
	Capability Maturity Model (CMM)	20
	Level 1 — Initial	21
	Level 2 — Repeatable	21
	Level 3 — Defined	22
	Level 4 — Managed	22
	Level 5 — Optimized	23
	PCMM	23
	CMMI	24
	Malcom Baldrige National Quality Award	24
	Notes	27

2	Overview of Testing Techniques	29
	Black-Box Testing (Functional)	29
	White-Box Testing (Structural)	30
	Gray-Box Testing (Functional and Structural)	30
	Manual versus Automated Testing	31
	Static versus Dynamic Testing	31
	Taxonomy of Software Testing Techniques	32
3	Quality through Continuous Improvement Process	41
	Contribution of Edward Deming	41
	Role of Statistical Methods	42
	Cause-and-Effect Diagram	42
	Flow Chart	42
	Pareto Chart	42
	Run Chart	42
	Histogram	43
	Scatter Diagram	43
	Control Chart	43
	Deming's 14 Quality Principles	43
	Point 1: Create Constancy of Purpose	43
	Point 2: Adopt the New Philosophy	44
	Point 3: Cease Dependence on Mass Inspection	44
	Point 4: End the Practice of Awarding Business on Price Tag Alone	44
	Point 5: Improve Constantly and Forever the System of Production and Service	45
	Point 6: Institute Training and Retraining	45
	Point 7: Institute Leadership	45
	Point 8: Drive Out Fear	46
	Point 9: Break Down Barriers between Staff Areas	46
	Point 10: Eliminate Slogans, Exhortations, and Targets for the Workforce	47
	Point 11: Eliminate Numerical Goals	47
	Point 12: Remove Barriers to Pride of Workmanship	47
	Point 13: Institute a Vigorous Program of Education and Retraining	48
	Point 14: Take Action to Accomplish the Transformation	48
	Continuous Improvement through the Plan, Do, Check, Act Process	48
	Going around the PDCA Circle	50
	SECTION II LIFE CYCLE TESTING REVIEW	51
4	Overview	53
	Waterfall Development Methodology	53
	Continuous Improvement "Phased" Approach	54

Psychology of Life Cycle Testing	54
Software Testing as a Continuous Improvement Process	55
The Testing Bible: Software Test Plan	58
Major Steps to Develop a Test Plan	60
1. Define the Test Objectives	60
2. Develop the Test Approach	60
3. Define the Test Environment	60
4. Develop the Test Specifications	61
5. Schedule the Test	61
6. Review and Approve the Test Plan	61
Components of a Test Plan	61
Technical Reviews as a Continuous Improvement Process	61
Motivation for Technical Reviews	65
Types of Reviews	66
Structured Walkthroughs	66
Inspections	66
Participant Roles	69
Steps for an Effective Review	70
1. Plan for the Review Process	70
2. Schedule the Review	70
3. Develop the Review Agenda	71
4. Create a Review Report	71
5 Verifying the Requirements Phase	73
Testing the Requirements with Technical Reviews	74
Inspections and Walkthroughs	74
Checklists	74
Methodology Checklist	75
Requirements Traceability Matrix	76
Building the System/Acceptance Test Plan	76
6 Verifying the Logical Design Phase	79
Data Model, Process Model, and the Linkage	79
Testing the Logical Design with Technical Reviews	80
Refining the System/Acceptance Test Plan	81
7 Verifying the Physical Design Phase	83
Testing the Physical Design with Technical Reviews	83
Creating Integration Test Cases	85
Methodology for Integration Testing	85
Step 1: Identify Unit Interfaces	85
Step 2: Reconcile Interfaces for Completeness	85
Step 3: Create Integration Test Conditions	86
Step 4: Evaluate the Completeness of Integration Test Conditions	86

8	Verifying the Program Unit Design Phase	87
	Testing the Program Unit Design with Technical Reviews	87
	Sequence	87
	Selection	87
	Iteration	87
	Creating Unit Test Cases	88
9	Verifying the Coding Phase	91
	Testing Coding with Technical Reviews	91
	Executing the Test Plan	91
	Unit Testing	92
	Integration Testing	93
	System Testing	93
	Acceptance Testing	94
	Defect Recording	95
SECTION III SOFTWARE TESTING METHODOLOGY		97
10	Development Methodology Overview	99
	Limitations of Life Cycle Development	99
	The Client/Server Challenge	100
	Psychology of Client/Server Spiral Testing	101
	The New School of Thought	101
	Tester/Developer Perceptions	102
	Project Goal: Integrate QA and Development	103
	Iterative/Spiral Development Methodology	104
	Role of JADs	106
	Role of Prototyping	107
	Methodology for Developing Prototypes	108
	1. Develop the Prototype	108
	2. Demonstrate Prototypes to Management	110
	3. Demonstrate Prototype to Users	110
	4. Revise and Finalize Specifications	111
	5. Develop the Production System	111
	Continuous Improvement “Spiral” Testing Approach	112
11	Information Gathering (Plan)	117
	Step 1: Prepare for the Interview	117
	Task 1: Identify the Participants	117
	Task 2: Define the Agenda	118
	Step 2: Conduct the Interview	118
	Task 1: Understand the Project	118
	Task 2: Understand the Project Objectives	121
	Task 3: Understand the Project Status	121
	Task 4: Understand the Project Plans	122
	Task 5: Understand the Project Development Methodology	122
	Task 6: Identify the High-Level Business Requirements	123

Task 7: Perform Risk Analysis	124
Computer Risk Analysis	124
Method 1 — Judgment and Instinct	125
Method 2 — Dollar Estimation	125
Method 3 — Identifying and Weighting Risk Attributes	125
Step 3: Summarize the Findings	126
Task 1: Summarize the Interview	126
Task 2: Confirm the Interview Findings	127
12 Test Planning (Plan)	129
Step 1: Build a Test Plan	130
Task 1: Prepare an Introduction	130
Task 2: Define the High-Level Functional Requirements (in Scope)	131
Task 3: Identify Manual/Automated Test Types	132
Task 4: Identify the Test Exit Criteria	133
Task 5: Establish Regression Test Strategy	134
Task 6: Define the Test Deliverables	136
Task 7: Organize the Test Team	137
Task 8: Establish a Test Environment	138
Task 9: Define the Dependencies	139
Task 10: Create a Test Schedule	139
Task 11: Select the Test Tools	142
Task 12: Establish Defect Recording/Tracking Procedures	143
Task 13: Establish Change Request Procedures	145
Task 14: Establish Version Control Procedures	147
Task 15: Define Configuration Build Procedures	147
Task 16: Define Project Issue Resolution Procedures	148
Task 17: Establish Reporting Procedures	148
Task 18: Define Approval Procedures	149
Step 2: Define the Metric Objectives	149
Task 1: Define the Metrics	150
Task 2: Define the Metric Points	151
Step 3: Review/Approve the Plan	154
Task 1: Schedule/Conduct the Review	154
Task 2: Obtain Approvals	154
13 Test Case Design (Do)	157
Step 1: Design Function Tests	157
Task 1: Refine the Functional Test Requirements	157
Task 2: Build a Function/Test Matrix	159
Step 2: Design GUI Tests	163
Ten Guidelines for Good GUI Design	164
Task 1: Identify the Application GUI Components	165
Task 2: Define the GUI Tests	165

Step 3: Define the System/Acceptance Tests	167
Task 1: Identify Potential System Tests.....	167
Task 2: Design System Fragment Tests	168
Task 3: Identify Potential Acceptance Tests.....	169
Step 4: Review/Approve Design	169
Task 1: Schedule/Prepare for Review	169
Task 2: Obtain Approvals	169
14 Test Development (Do)	173
Step 1: Develop Test Scripts	173
Task 1: Script the Manual/Automated GUI/Function Tests	173
Task 2: Script the Manual/Automated System Fragment	
Tests	173
Step 2: Review/Approve Test Development	174
Task 1: Schedule/Prepare for Review	174
Task 2: Obtain Approvals	174
15 Test Coverage through Traceability	177
Use Cases and Traceability.....	178
Summary	180
16 Test Execution/Evaluation (Do/Check).....	181
Step 1: Setup and Testing	181
Task 1: Regression Test the Manual/Automated Spiral Fixes. . .	181
Task 2: Execute the Manual/Automated New Spiral Tests	182
Task 3: Document the Spiral Test Defects	183
Step 2: Evaluation.....	183
Task 1: Analyze the Metrics	183
Step 3: Publish Interim Report	184
Task 1: Refine the Test Schedule	184
Task 2: Identify Requirement Changes	185
17 Prepare for the Next Spiral (Act)	187
Step 1: Refine the Tests	187
Task 1: Update the Function/GUI Tests.....	187
Task 2: Update the System Fragment Tests	188
Task 3: Update the Acceptance Tests	189
Step 2: Reassess the Team, Procedures, and Test Environment....	189
Task 1: Evaluate the Test Team	189
Task 2: Review the Test Control Procedures	189
Task 3: Update the Test Environment	190
Step 3: Publish Interim Test Report.....	191
Task 1: Publish the Metric Graphics	191
Test Case Execution Status	191
Defect Gap Analysis	191
Defect Severity Status.....	191
Test Burnout Tracking	192

18	Conduct the System Test	195
	Step 1: Complete System Test Plan	195
	Task 1: Finalize the System Test Types.	195
	Task 2: Finalize System Test Schedule	197
	Task 3: Organize the System Test Team	197
	Task 4: Establish the System Test Environment	197
	Task 5: Install the System Test Tools	200
	Step 2: Complete System Test Cases	200
	Task 1: Design/Script the Performance Tests	200
	Monitoring Approach	201
	Probe Approach	202
	Test Drivers	202
	Task 2: Design/Script the Security Tests	203
	A Security Design Strategy	203
	Task 3: Design/Script the Volume Tests	204
	Task 4: Design/Script the Stress Tests	205
	Task 5: Design/Script the Compatibility Tests.	206
	Task 6: Design/Script the Conversion Tests.	206
	Task 7: Design/Script the Usability Tests	207
	Task 8: Design/Script the Documentation Tests	208
	Task 9: Design/Script the Backup Tests	208
	Task 10: Design/Script the Recovery Tests	209
	Task 11: Design/Script the Installation Tests.	209
	Task 12: Design/Script Other System Test Types	210
	Step 3: Review/Approve System Tests	211
	Task 1: Schedule/Conduct the Review	211
	Task 2: Obtain Approvals	212
	Step 4: Execute the System Tests.	212
	Task 1: Regression Test the System Fixes	212
	Task 2: Execute the New System Tests	213
	Task 3: Document the System Defects	213
19	Conduct Acceptance Testing.	215
	Step 1: Complete Acceptance Test Planning	215
	Task 1: Finalize the Acceptance Test Types.	215
	Task 2: Finalize the Acceptance Test Schedule	215
	Task 3: Organize the Acceptance Test Team	215
	Task 4: Establish the Acceptance Test Environment	217
	Task 5: Install Acceptance Test Tools.	218
	Step 2: Complete Acceptance Test Cases	218
	Task 1: Subset the System-Level Test Cases	218
	Task 2: Design/Script Additional Acceptance Tests	219
	Step 3: Review/Approve Acceptance Test Plan	219
	Task 1: Schedule/Conduct the Review	219
	Task 2: Obtain Approvals	220

Step 4: Execute the Acceptance Tests	220
Task 1: Regression Test the Acceptance Fixes	220
Task 2: Execute the New Acceptance Tests	220
Task 3: Document the Acceptance Defects	221
20 Summarize/Report Spiral Test Results	223
Step 1: Perform Data Reduction	223
Task 1: Ensure All Tests Were Executed/Resolved	223
Task 2: Consolidate Test Defects by Test Number	223
Task 3: Post Remaining Defects to a Matrix	223
Step 2: Prepare Final Test Report	224
Task 1: Prepare the Project Overview	225
Task 2: Summarize the Test Activities	225
Task 3: Analyze/Create Metric Graphics	225
Defects by Function	225
Defects by Tester	227
Defect Gap Analysis	227
Defect Severity Status	227
Test Burnout Tracking	227
Root Cause Analysis	227
Defects by How Found	230
Defects by Who Found	230
Functions Tested and Not	230
System Testing Defect Types	230
Acceptance Testing Defect Types	232
Task 4: Develop Findings/Recommendations	232
Step 3: Review/Approve the Final Test Report	233
Task 1: Schedule/Conduct the Review	233
Task 2: Obtain Approvals	236
Task 3: Publish the Final Test Report	236
SECTION IV TEST PROJECT MANAGEMENT	237
21 Overview of General Project Management	239
Define the Objectives	239
Define the Scope of the Project	240
Identify the Key Activities	240
Estimate Correctly	240
Design	241
Manage People	241
Leadership	242
Communication	242
Solving Problems	243
Continuous Monitoring	243
Manage Changes	243

22	Test Project Management	245
	Understand the Requirements	246
	Test Planning	246
	Test Execution	247
	Identify and Improve Processes	247
	Essential Characteristics of a Test Project Manager	248
	Requirement Analysis	248
	Gap Analysis	248
	Lateral Thinking in Developing Test Cases	248
	Avoid Duplication and Repetition	249
	Test Data Generation	249
	Validate the Test Environment	249
	Test to Destroy	249
	Analyze the Test Results	249
	Do Not Hesitate to Accept Help from Others	250
	Convey Issues as They Arise	250
	Improve Communication	250
	Always Keep Updating Your Business Knowledge	250
	Learn the New Testing Technologies and Tools	250
	Deliver Quality	251
	Improve the Process	251
	Create a Knowledge Base	251
	Repeat the Success	251
23	Test Estimation	253
	Finish-to-Start: (FS)	255
	Start-to-Start: (SS)	255
	Finish-to-Finish: (FF)	255
	Start-to-Finish (SF)	255
	Critical Activities for Test Estimation	255
	Test Scope Document	256
	Test Strategy	256
	Test Condition	257
	Test Case	257
	Test Script	257
	Execution/Run Plan	257
	Factors Affecting Test Estimation	257
	Test Planning Estimation	258
	Test Execution and Controlling Effort	259
	Test Result Analysis	259
	Effort Estimation — Model Project	259
24	Defect Monitoring and Management Process	263
	Defect Reporting	264
	Defect Meetings	265

	Defect Classifications	265
	Defect Priority	266
	Defect Category	266
	Defect Metrics	267
25	Integrating Testing into Development Methodology	269
	Step 1. Organize the Test Team	270
	Step 2. Identify Test Steps and Tasks to Integrate	270
	Step 3. Customize Test Steps and Tasks	271
	Step 4. Select Integration Points	271
	Step 5. Modify the Development Methodology	272
	Step 6. Incorporate Defect Recording	272
	Step 7. Train in Use of the Test Methodology	272
26	On-Site/Offshore Model	275
	Step 1: Analysis	275
	Step 2: Determine the Economic Tradeoffs	276
	Step 3: Determine the Selection Criteria	276
	Project Management and Monitoring	276
	Outsourcing Methodology	277
	On-Site Activities	278
	Offshore Activities	278
	Implementing the On-Site/Offshore Model	279
	Knowledge Transfer	279
	Detailed Design	280
	Milestone-Based Transfer	280
	Steady State	280
	Application Management	280
	Relationship Model	281
	Standards	282
	Benefits of On-Site/Offshore Methodology	283
	On-Site/Offshore Model Challenges	285
	Out of Sight	285
	Establish Transparency	285
	Security Considerations	285
	Project Monitoring	285
	Management Overhead	285
	Cultural Differences	285
	Software Licensing	285
	The Future of Onshore/Offshore	286
	SECTION V MODERN SOFTWARE TESTING TOOLS	287
27	A Brief History of Software Testing	289
	Evolution of Automated Testing Tools	293
	Static Capture/Replay Tools (without Scripting Language)	294
	Static Capture/Replay Tools (with Scripting Language)	294

Variable Capture/Replay Tools	295
Functional Decomposition Approach	295
Test Plan Driven (“Keyword”) Approach	296
Historical Software Testing and Development Parallels	298
Extreme Programming	299
28 Software Testing Trends	301
Automated Capture/Replay Testing Tools	301
Test Case Builder Tools	302
Advanced Leading-Edge Automated Testing Tools	302
Advanced Leading-Edge Test Case Builder Tools	304
Necessary and Sufficient Conditions	304
Test Data/Test Case Generation	305
Sampling from Production	305
Starting from Scratch	306
Seeding the Data	306
Generating Data Based upon the Database	307
Generating Test Data/Test Cases Based upon the Requirements	308
29 Taxonomy of Testing Tools	311
Testing Tool Selection Checklist	311
Vendor Tool Descriptions	312
When You Should Consider Test Automation	312
When You Should NOT Consider Test Automation	320
30 Methodology to Evaluate Automated Testing Tools	323
Step 1: Define Your Test Requirements	323
Step 2: Set Tool Objectives	323
Step 3a: Conduct Selection Activities for Informal Procurement	324
Task 1: Develop the Acquisition Plan	324
Task 2: Define Selection Criteria	324
Task 3: Identify Candidate Tools	324
Task 4: Conduct the Candidate Review	325
Task 5: Score the Candidates	325
Task 6: Select the Tool	325
Step 3b: Conduct Selection Activities for Formal Procurement	326
Task 1: Develop the Acquisition Plan	326
Task 2: Create the Technical Requirements Document	326
Task 3: Review Requirements	326
Task 4: Generate the Request for Proposal	326
Task 5: Solicit Proposals	326
Task 6: Perform the Technical Evaluation	327
Task 7: Select a Tool Source	327

Step 4: Procure the Testing Tool	327
Step 5: Create the Evaluation Plan	327
Step 6: Create the Tool Manager's Plan	328
Step 7: Create the Training Plan	328
Step 8: Receive the Tool	328
Step 9: Perform the Acceptance Test	329
Step 10: Conduct Orientation	329
Step 11: Implement Modifications	329
Step 12: Train Tool Users	329
Step 13: Use the Tool in the Operating Environment	330
Step 14: Write the Evaluation Report	330
Step 15: Determine Whether Goals Have Been Met	330
APPENDICES	331
A Spiral Testing Methodology	333
B Software Quality Assurance Plan	343
C Requirements Specification	345
D Change Request Form	347
E Test Templates	349
E1: Unit Test Plan	349
E2: System/Acceptance Test Plan	349
E3: Requirements Traceability Matrix	351
E4: Test Plan (Client/Server and Internet Spiral Testing)	353
E5: Function/Test Matrix	355
E6: GUI Component Test Matrix (Client/Server and Internet Spiral Testing)	355
E7: GUI-Based Functional Test Matrix (Client/Server and Internet Spiral Testing)	356
E8: Test Case	357
E9: Test Case Log	357
E10: Test Log Summary Report	359
E11: System Summary Report	361
E12: Defect Report	362
E13: Test Schedule	364
E14: Retest Matrix	366
E15: Spiral Testing Summary Report (Client/Server and Internet Spiral Testing)	368
E16: Minutes of the Meeting	368
E17: Test Approvals	370
E18: Test Execution Plan	371
E19: Test Project Milestones	372
E20: PDCA Test Schedule	373
E21: Test Strategy	374

E22: Clarification Request	377
E23: Screen Data Mapping.	378
E24: Test Condition versus Test Case	379
E25: Project Status Report	380
E26: Test Defect Details Report	381
E27: Defect Report	383
E28: Test Execution Tracking Manager	383
E29: Final Test Summary Report	384
F Checklists	387
F1: Requirements Phase Defect Checklist.	388
F2: Logical Design Phase Defect Checklist	389
F3: Physical Design Phase Defect Checklist	390
F4: Program Unit Design Phase Defect Checklist.	393
F5: Coding Phase Defect Checklist.	394
F6: Field Testing Checklist	396
F7: Record Testing Checklist	398
F8: File Test Checklist	400
F9: Error Testing Checklist	401
F10: Use Test Checklist	403
F11: Search Test Checklist	404
F12: Match/Merge Checklist	405
F13: Stress Test Checklist	407
F14: Attributes Testing Checklist	409
F15: States Testing Checklist	411
F16: Procedures Testing Checklist.	412
F17: Control Testing Checklist	413
F18: Control Flow Testing Checklist.	418
F19: Testing Tool Selection Checklist	419
F20: Project Information Gathering Checklist	421
F21: Impact Analysis Checklist	423
F22: Environment Readiness Checklist	425
F23: Project Completion Checklist.	427
F24: Unit Testing Checklist	429
F25: Ambiguity Review Checklist	433
F26: Architecture Review Checklist	435
F27: Data Design Review Checklist	436
F28: Functional Specification Review Checklist	437
F29: Prototype Review Checklist	442
F30: Requirements Review Checklist	443
F31: Technical Design Review Checklist	447
F32: Test Case Preparation Review Checklist.	449
G Software Testing Techniques	451
G1: Basis Path Testing	451
PROGRAM: FIELD-COUNT.	451

G2: Black-Box Testing	452
Extra Program Logic	453
G3: Bottom-Up Testing	453
G4: Boundary Value Testing	453
Numeric Input Data	454
Field Ranges	454
Numeric Output Data	454
Output Range of Values	454
Nonnumeric Input Data	454
Tables or Arrays	454
Number of Items	454
Nonnumeric Output Data	454
Tables or Arrays	454
Number of Outputs	454
GUI	454
G5: Branch Coverage Testing	455
PROGRAM: FIELD-COUNT	455
G6: Branch/Condition Coverage Testing	455
PROGRAM: FIELD-COUNT	456
G7: Cause-Effect Graphing	456
Cause-Effect Methodology	457
Specification	458
Causes	458
Effects	458
G8: Condition Coverage	460
PROGRAM: FIELD-COUNT	460
G9: CRUD Testing	461
G10: Database Testing	461
Integrity Testing	461
Entity Integrity	462
Primary Key Integrity	462
Column Key Integrity	462
Domain Integrity	463
User-Defined Integrity	463
Referential Integrity	463
Data Modeling Essentials	464
What Is a Model?	465
Why Do We Create Models?	465
Tables — A Definition	466
Table Names	467
Columns	467
Rows	467
Order	467
Entities — A Definition	467

Identification — Primary Key	468
Compound Primary Keys.....	468
Null Values	468
Identifying Entities	469
Entity Classes	469
Relationships — A Definition.....	470
Relationship Types.....	470
One-to-One.....	470
One-to-Many	472
Many-to-Many	473
Multiple Relationships	475
Entities versus Relationships	475
Attributes — A Definition.....	476
Domain	477
Domain Names.....	478
Attributes versus Relationships	478
Normalization — What Is It?	479
Problems of Unnormalized Entities	479
Steps in Normalization	480
First Normal Form (1NF)	480
Second Normal Form (2NF).....	482
Third Normal Form (3NF)	484
Model Refinement.....	485
Entity Subtypes.....	486
A Definition	486
Referential Integrity	486
Dependency Constraints	488
Constraint Rule	488
Recursion.....	489
Using the Model in Database Design	491
Relational Design	491
G11: Decision Tables	492
PROGRAM: FIELD-COUNT.....	492
G12: Desk Checking	493
G13: Equivalence Partitioning	493
Numeric Input Data	494
Field Ranges.....	494
Numeric Output Data	494
Output Range of Values	494
Nonnumeric Input Data	494
Tables or Arrays	494
Number of Items	494
Nonnumeric Output Data	494
Tables or Arrays	494
Number of Outputs.....	494

G14: Exception Testing	494
G15: Free Form Testing	494
G16: Gray-Box Testing	495
G17: Histograms	496
G18: Inspections	496
G19: JADs	497
G20: Orthogonal Array Testing	498
G21: Pareto Analysis	499
G22: Positive and Negative Testing	501
G23: Prior Defect History Testing	502
G24: Prototyping	502
Cyclic Models	502
Fourth-Generation Languages and Prototyping	503
Iterative Development Accounting	504
Evolutionary and Throwaway	504
Application Prototyping	505
Prototype Systems Development	505
Data-Driven Prototyping	505
Replacement of the Traditional Life Cycle	506
Early-Stage Prototyping	506
User Software Engineering	507
G25: Random Testing	507
G26: Range Testing	507
G27: Regression Testing	509
G28: Risk-Based Testing	509
G29: Run Charts	510
G30: Sandwich Testing	510
G31: Statement Coverage Testing	511
PROGRAM: FIELD-COUNT	511
G32: State Transition Testing	511
PROGRAM: FIELD-COUNT	512
G33: Statistical Profile Testing	512
G34: Structured Walkthroughs	512
G35: Syntax Testing	514
G36: Table Testing	514
G37: Thread Testing	515
G38: Top-Down Testing	515
G39: White-Box Testing	516
Bibliography	517
Glossary	523
Index	529