<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
</tr>
<tr>
<td><strong>Part I</strong></td>
</tr>
<tr>
<td><strong>Chapter 1</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Part II</strong></td>
</tr>
<tr>
<td><strong>Chapter 2</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Topics</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Risks Associated with Not Meeting Customer Needs</td>
</tr>
<tr>
<td>Developing a Role for Software Testers</td>
</tr>
<tr>
<td>Writing a Policy for Software Testing</td>
</tr>
<tr>
<td>Criteria for a Testing Policy</td>
</tr>
<tr>
<td>Methods for Establishing a Testing Policy</td>
</tr>
<tr>
<td>Economics of Testing</td>
</tr>
<tr>
<td>Testing—An Organizational Issue</td>
</tr>
<tr>
<td>Management Support for Software Testing</td>
</tr>
<tr>
<td>Building a Structured Approach to Software Testing</td>
</tr>
<tr>
<td>Requirements</td>
</tr>
<tr>
<td>Design</td>
</tr>
<tr>
<td>Program</td>
</tr>
<tr>
<td>Test</td>
</tr>
<tr>
<td>Installation</td>
</tr>
<tr>
<td>Maintenance</td>
</tr>
<tr>
<td>Developing a Test Strategy</td>
</tr>
<tr>
<td>Use Work Paper 2-1</td>
</tr>
<tr>
<td>Use Work Paper 2-2</td>
</tr>
<tr>
<td>Summary</td>
</tr>
</tbody>
</table>

**Chapter 3 Building the Software Testing Process**

<table>
<thead>
<tr>
<th>Guidelines</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guideline #1: Testing Should Reduce Software Development Risk</td>
<td>63</td>
</tr>
<tr>
<td>Guideline #2: Testing Should Be Performed Effectively</td>
<td>64</td>
</tr>
<tr>
<td>Guideline #3: Testing Should Uncover Defects</td>
<td>65</td>
</tr>
<tr>
<td>Defects Versus Failures</td>
<td>65</td>
</tr>
<tr>
<td>Why Are Defects Hard to Find?</td>
<td>65</td>
</tr>
<tr>
<td>Guideline #4: Testing Should Be Performed Using Business Logic</td>
<td>66</td>
</tr>
<tr>
<td>Guideline #5: Testing Should Occur Throughout the Development Life Cycle</td>
<td>67</td>
</tr>
<tr>
<td>Guideline #6: Testing Should Test Both Function and Structure</td>
<td>68</td>
</tr>
<tr>
<td>Why Use Both Testing Methods?</td>
<td>69</td>
</tr>
<tr>
<td>Structural and Functional Tests Using Verification and Validation Techniques</td>
<td>69</td>
</tr>
<tr>
<td>Workbench Concept</td>
<td>71</td>
</tr>
<tr>
<td>Testing That Parallels the Software Development Process</td>
<td>72</td>
</tr>
<tr>
<td>Customizing the Software-Testing Process</td>
<td>74</td>
</tr>
<tr>
<td>Determining the Test Strategy Objectives</td>
<td>74</td>
</tr>
<tr>
<td>Determining the Type of Development Project</td>
<td>75</td>
</tr>
<tr>
<td>Determining the Type of Software System</td>
<td>76</td>
</tr>
<tr>
<td>Determining the Project Scope</td>
<td>77</td>
</tr>
<tr>
<td>Identifying the Software Risks</td>
<td>77</td>
</tr>
<tr>
<td>Determining When Testing Should Occur</td>
<td>79</td>
</tr>
<tr>
<td>Defining the System Test Plan Standard</td>
<td>79</td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Task 6: Inspect the Test Plan</th>
<th>254</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspection Concerns</td>
<td>255</td>
</tr>
<tr>
<td>Products/Deliverables to Inspect</td>
<td>256</td>
</tr>
<tr>
<td>Formal Inspection Roles</td>
<td>256</td>
</tr>
<tr>
<td>Formal Inspection Defect Classification</td>
<td>258</td>
</tr>
<tr>
<td>Inspection Procedures</td>
<td>259</td>
</tr>
<tr>
<td>Check Procedures</td>
<td>262</td>
</tr>
<tr>
<td>Output</td>
<td>262</td>
</tr>
<tr>
<td>Guidelines</td>
<td>262</td>
</tr>
<tr>
<td>Summary</td>
<td>263</td>
</tr>
</tbody>
</table>

### Chapter 9

**Step 3: Verification Testing**

- **Overview**
  - 292
- **Objective**
  - 293
- **Concerns**
  - 294
- **Workbench**
  - 294
- **Input**
  - The Requirements Phase
    - 296
  - The Design Phase
    - 296
  - The Programming Phase
    - 297
- **Do Procedures**
  - 298
  - Task 1: Test During the Requirements Phase
    - 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
    - Scoring Success Factors
      - 316
    - Analyzing Test Factors
      - 318
    - Conducting a Design Review
      - 320
    - Inspecting Design Deliverables
      - 322
  - Task 3: Test During the Programming Phase
    - Desk Debugging the Program
      - 323
    - 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**

- **Overview**
  - 409
- **Objective**
  - 410
- **Concerns**
  - 410
- **Workbench**
  - 410
- **Input**
  - 411
Do Procedures 412
   Task 1: Build the Test Data 412
      Sources of Test Data/Test Scripts 412
      Testing File Design 413
      Defining Design Goals 414
      Entering Test Data 414
      Applying Test Files Against Programs That Update
         Master Records 414
      Creating and Using Test Data 415
      Payroll Application Example 416
      Creating Test Data for Stress/Load Testing 430
      Creating Test Scripts 430
   Task 2: Execute Tests 434
   Task 3: Record Test Results 436
      Documenting the Deviation 437
      Documenting the Effect 438
      Documenting the Cause 438
Check Procedures 439
Output 439
Guidelines 439
Summary 440

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  491

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
Chapter 13  Step 7: Post-Implementation Analysis  
Overview  
Concerns  
Workbench  
Input  
Do Procedures  
Task 1: Establish Assessment Objectives  
Task 2: Identify What to Measure  
Task 3: Assign Measurement Responsibility  
Task 4: Select Evaluation Approach  
Task 5: Identify Needed Facts  
Task 6: Collect Evaluation Data  
Task 7: Assess the Effectiveness of Testing  
Using Testing Metrics  
Check Procedures  
Output  
Guidelines  
Summary  

Part IV  Incorporating Specialized Testing Responsibilities  

Chapter 14  Software Development Methodologies  
How Much Testing Is Enough?  
Software Development Methodologies  
Overview  
Methodology Types  
Software Development Life Cycle  
Defining Requirements  
Categories  
Attributes  
Methodology Maturity  
Competencies Required  
Staff Experience  
Configuration-Management Controls  
Basic CM Requirements  
Planning  
Data Distribution and Access  
CM Administration  
Configuration Identification  
Configuration Control  
Measuring the Impact of the Software Development Process  
Summary  

Chapter 15  Testing Client/Server Systems  
Overview  
Concerns  
Workbench  
Input
Do Procedures

Task 1: Assess Readiness

Software Development Process Maturity Levels
Conducting the Client/Server Readiness Assessment
Preparing a Client/Server Readiness Footprint Chart

Task 2: Assess Key Components

Task 3: Assess Client Needs

Check Procedures

Output

Guidelines

Summary

**Chapter 16  Rapid Application Development Testing**

Overview

Objective

Concerns

Testing Iterations

Testing Components

Testing Performance

Recording Test Information

Workbench

Input

Do Procedures

Testing Within Iterative RAD

Spiral Testing

Task 1: Determine Appropriateness of RAD

Task 2: Test Planning Iterations

Task 3: Test Subsequent Planning Iterations

Task 4: Test the Final Planning Iteration

Check Procedures

Output

Guidelines

Summary

**Chapter 17  Testing Internal Controls**

Overview

Internal Controls

Control Objectives

Preventive Controls

Source-Data Authorization

Data Input

Source-Data Preparation

Turnaround Documents

Prenumbered Forms

Input Validation

File Auto-Updating

Processing Controls
Chapter 18 Testing COTS and Contracted Software

Overview

COTS Software Advantages, Disadvantages, and Risks

COTS Versus Contracted Software

COTS Advantages

COTS Disadvantages

Implementation Risks

Testing COTS Software

Testing Contracted Software

Objective

Concerns

Workbench

Input

Do Procedures

Task 1: Test Business Fit

Step 1: Testing Needs Specification

Step 2: Testing CSFs

Task 2: Test Operational Fit

Step 1: Test Compatibility

Step 2: Integrate the Software into Existing Work Flows

Step 3: Demonstrate the Software in Action

Task 3: Test People Fit
| Step 7: Develop and Execute an Implementation Plan | 891 |
| Planning | 891 |
| Implementing Ideas | 891 |
| Requisite Resources | 893 |
| Quality Control Checklist | 894 |
| Conclusion | 894 |
| Summary | 895 |

**Index**

| Index | 929 |