## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>xvii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xxi</td>
</tr>
<tr>
<td>Preface</td>
<td>xxv</td>
</tr>
</tbody>
</table>

### PART I  OVERVIEW AND BASICS

1 **Overview**

1.1 Meeting People’s Quality Expectations  
1.2 Book Organization and Chapter Overview  
1.3 Dependency and Suggested Usage  
1.4 Reader Preparation and Background Knowledge  
Problems  

2 **What Is Software Quality?**

2.1 Quality: Perspectives and Expectations  
2.2 Quality Frameworks and ISO-9126  
2.3 Quality, Correctness, and Defects  
2.3.1 Definitions: Error, fault, failure, and defect  
2.3.2 Concepts and relations illustrated  
2.3.3 Correctness-centered properties and measurements  
2.4 A Historical Perspective of Quality  

vii
2.5 So, What Is Software Quality? 24
Problems 24

3 Quality Assurance 25
3.1 Classification: QA as Dealing with Defects 25
3.2 Defect Prevention 29
  3.2.1 Education and training 29
  3.2.2 Formal method 30
  3.2.3 Other defect prevention techniques 31
3.3 Defect Reduction 32
  3.3.1 Inspection: Direct fault detection and removal 32
  3.3.2 Testing: Failure observation and fault removal 33
  3.3.3 Other techniques and risk identification 34
3.4 Defect Containment 35
  3.4.1 Software fault tolerance 35
  3.4.2 Safety assurance and failure containment 36
3.5 Concluding Remarks 36
Problems 37

4 Quality Assurance in Context 39
4.1 Handling Discovered Defect During QA Activities 39
4.2 QA Activities in Software Processes 41
4.3 Verification and Validation Perspectives 44
4.4 Reconciling the Two Views 47
4.5 Concluding Remarks 49
Problems 50

5 Quality Engineering 51
5.1 Quality Engineering: Activities and Process 51
5.2 Quality Planning: Goal Setting and Strategy Formation 54
5.3 Quality Assessment and Improvement 57
5.4 Quality Engineering in Software Processes 57
  5.4.1 Activity distribution and integration 58
  5.4.2 Effort profile 59
5.5 Concluding Remarks 61
Problems 62

PART II SOFTWARE TESTING

6.1 Purposes, Activities, Processes, and Context 65
6.2 Questions about Testing 69
6.3 Functional vs Structural Testing: What to Test?  
6.4 Coverage-Based vs Usage-Based Testing: When to Stop Testing?  
6.5 Concluding Remarks  
Problems  

7 Test Activities, Management, and Automation  
7.1 Test Planning and Preparation  
7.1.1 Test planning: Goals, strategies and techniques  
7.1.2 Testing models and test cases  
7.1.3 Preparation of individual test cases  
7.1.4 Test suite preparation and management  
7.1.5 Preparation of test procedure  
7.2 Test Execution and Measurement  
7.2.1 Overall activities and management  
7.2.2 Result checking: The oracle problem  
7.2.3 Test measurement  
7.3 Analysis and Followup  
7.4 Activities, People, and Management  
7.5 Test Automation  
7.6 Concluding Remarks  
Problems  

8 Coverage and Usage Testing Based on Checklists and Partitions  
8.1 Checklist-Based Testing and Its Limitations  
8.2 Testing for Partition Coverage  
8.2.1 Some motivational examples  
8.2.2 Partition: Concepts and definitions  
8.2.3 Testing decisions and predicates for partition coverage  
8.3 Usage-Based Statistical Testing with Musa’s Operational Profiles (OPs)  
8.3.1 The cases for usage-based statistical testing  
8.3.2 Musa OP: Basic ideas  
8.3.3 Using OPs for statistical testing and other purposes  
8.4 Constructing Operational Profiles  
8.4.1 Generic methods and participants  
8.4.2 OP development procedure: Musa-1  
8.4.3 OP development procedure: Musa-2  
8.5 Case Study: OP for the Cartridge Support Software  
8.5.1 Background and participants  
8.5.2 OP development in five steps  
8.5.3 Metrics collection, result validation, and lessons learned  
8.6 Concluding Remarks
9 Input Domain Partitioning and Boundary Testing

9.1 Input Domain Partitioning and Testing
  9.1.1 Basic concepts, definitions, and terminology
  9.1.2 Input domain testing
  9.1.3 Partition and boundary problems

9.2 Simple Domain Analysis and the Extreme Point Combination (EPC) Strategy

9.3 Testing Strategies Based on Boundary Analysis
  9.3.1 Weak $N \times 1$ strategy
  9.3.2 Weak $1 \times 1$ strategy

9.4 Other Boundary Test Strategies and Applications
  9.4.1 Strong and approximate strategies
  9.4.2 Other types of boundaries and extensions
  9.4.3 Queuing testing as boundary testing

9.5 Concluding Remarks

Problems 124

10 Coverage and Usage Testing Based on FSMs and Markov Chains

10.1 Finite-State Machines (FSMs) and Testing
  10.1.1 Overcoming limitations of simple processing models
  10.1.2 FSMs: Basic concepts and examples
  10.1.3 Representations of FSMs

10.2 FSM Testing: State and Transition Coverage
  10.2.1 Some typical problems with systems modeled by FSMs
  10.2.2 Model construction and checking for missing or extra states or transitions
  10.2.3 Testing for correct states and transitions
  10.2.4 Applications and limitations

10.3 Case Study: FSM-Based Testing of Web-Based Applications
  10.3.1 Characteristics of web-based applications
  10.3.2 What to test: Characteristics of web problems
  10.3.3 FSMs for web testing

10.4 Markov Chains and Unified Markov Models (UMMs) for Testing
  10.4.1 Markov chains and operational profiles
  10.4.2 From individual Markov chains to unified Markov models (UMMs)
  10.4.3 UMM construction

10.5 Using UMMs for Usage-based Statistical Testing
  10.5.1 Testing based on usage frequencies in UMMs
  10.5.2 Testing based on other criteria and UMM hierarchies
  10.5.3 Implementation, application, and other issues

10.6 Case Study Continued: Testing Based on Web Usages

Problems 145
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>13.2 Education and Training for Defect Prevention</td>
<td>226</td>
</tr>
<tr>
<td>13</td>
<td>13.3 Other Techniques for Defect Prevention</td>
<td>228</td>
</tr>
<tr>
<td>13</td>
<td>13.3.1 Analysis and modeling for defect prevention</td>
<td>228</td>
</tr>
<tr>
<td>13</td>
<td>13.3.2 Technologies, standards, and methodologies for defect prevention</td>
<td>229</td>
</tr>
<tr>
<td>13</td>
<td>13.3.3 Software tools to block defect injection</td>
<td>230</td>
</tr>
<tr>
<td>13</td>
<td>13.4 Focusing on Software Processes</td>
<td>231</td>
</tr>
<tr>
<td>13</td>
<td>13.4.1 Process selection, definition, and conformance</td>
<td>232</td>
</tr>
<tr>
<td>13</td>
<td>13.4.2 Process maturity</td>
<td>232</td>
</tr>
<tr>
<td>13</td>
<td>13.4.3 Process and quality improvement</td>
<td>234</td>
</tr>
<tr>
<td>13</td>
<td>13.5 Concluding Remarks</td>
<td>234</td>
</tr>
<tr>
<td>13</td>
<td>Problems</td>
<td>235</td>
</tr>
<tr>
<td>14</td>
<td>Software Inspection</td>
<td>237</td>
</tr>
<tr>
<td>14</td>
<td>14.1 Basic Concepts and Generic Process</td>
<td>237</td>
</tr>
<tr>
<td>14</td>
<td>14.2 Fagan inspection</td>
<td>239</td>
</tr>
<tr>
<td>14</td>
<td>14.3 Other Inspections and Related Activities</td>
<td>242</td>
</tr>
<tr>
<td>14</td>
<td>14.3.1 Inspections of reduced scope or team size</td>
<td>242</td>
</tr>
<tr>
<td>14</td>
<td>14.3.2 Inspections of enlarged scope or team size</td>
<td>243</td>
</tr>
<tr>
<td>14</td>
<td>14.3.3 Informal desk checks, reviews, and walkthroughs</td>
<td>244</td>
</tr>
<tr>
<td>14</td>
<td>14.3.4 Code reading</td>
<td>245</td>
</tr>
<tr>
<td>14</td>
<td>14.3.5 Other formal reviews and static analyses</td>
<td>246</td>
</tr>
<tr>
<td>14</td>
<td>14.3.6 Tool support and process integration</td>
<td>247</td>
</tr>
<tr>
<td>14</td>
<td>14.4 Defect Detection Techniques and Inspection Effectiveness</td>
<td>247</td>
</tr>
<tr>
<td>14</td>
<td>14.5 Concluding Remarks</td>
<td>249</td>
</tr>
<tr>
<td>14</td>
<td>Problems</td>
<td>250</td>
</tr>
<tr>
<td>15</td>
<td>Formal Verification</td>
<td>251</td>
</tr>
<tr>
<td>15</td>
<td>15.1 Basic Concepts: Formal Verification and Formal Specification</td>
<td>251</td>
</tr>
<tr>
<td>15</td>
<td>15.2 Formal Verification: Axiomatic Approach</td>
<td>254</td>
</tr>
<tr>
<td>15</td>
<td>15.2.1 Formal logic specifications</td>
<td>254</td>
</tr>
<tr>
<td>15</td>
<td>15.2.2 Axioms</td>
<td>255</td>
</tr>
<tr>
<td>15</td>
<td>15.2.3 Axiomatic proofs and a comprehensive example</td>
<td>257</td>
</tr>
<tr>
<td>15</td>
<td>15.3 Other Approaches</td>
<td>259</td>
</tr>
<tr>
<td>15</td>
<td>15.3.1 Weakest pre-conditions and backward chaining</td>
<td>260</td>
</tr>
<tr>
<td>15</td>
<td>15.3.2 Functional approach and symbolic execution</td>
<td>260</td>
</tr>
<tr>
<td>15</td>
<td>15.3.3 General observations</td>
<td>261</td>
</tr>
<tr>
<td>15</td>
<td>15.3.4 Model checking and other approaches</td>
<td>262</td>
</tr>
<tr>
<td>15</td>
<td>15.4 Applications, Effectiveness, and Integration Issues</td>
<td>263</td>
</tr>
<tr>
<td>15</td>
<td>15.5 Concluding Remarks</td>
<td>265</td>
</tr>
<tr>
<td>15</td>
<td>Problems</td>
<td>266</td>
</tr>
</tbody>
</table>
18.3.1 Analyses for product release decisions 309
18.3.2 Analyses for other project management decisions 311
18.3.3 Feedback to analyses and models themselves 312
18.3.4 Longer term and broader scope followup actions 313
18.4 Implementation, Integration, and Tool Support 314
  18.4.1 Feedback loop: Implementation and integration 314
  18.4.2 A refined quality engineering process 315
  18.4.3 Tool support: Strategy, implementation, and integration 317
18.5 Concluding Remarks 320
  Problems 320

19 Quality Models and Measurements 323
  19.1 Models for Quality Assessment 323
  19.2 Generalized Models 324
  19.3 Product-Specific Models 327
  19.4 Model Comparison and Interconnections 328
  19.5 Data Requirements and Measurement 330
  19.6 Selecting Measurements and Models 333
  19.7 Concluding Remarks 335
  Problems 337

20 Defect Classification and Analysis 339
  20.1 General Types of Defect Analyses 339
    20.1.1 Defect Distribution Analysis 340
    20.1.2 Defect Trend Analysis and Defect Dynamics Model 343
    20.1.3 Defect Causal Analysis 344
  20.2 Defect Classification and ODC 345
    20.2.1 ODC concepts 345
    20.2.2 Defect classification using ODC: A comprehensive example 346
    20.2.3 Adapting ODC to analyze web errors 347
  20.3 Defect Analysis for Classified Data 348
    20.3.1 One-way analysis: Analyzing a single defect attribute 348
    20.3.2 Two-way and multi-way analysis: Examining cross-interactions 349
  20.4 Concluding Remarks 350
  Problems 351

21 Risk Identification for Quantifiable Quality Improvement 353
  21.1 Basic Ideas and Concepts 353
  21.2 Traditional Statistical Analysis Techniques 355
  21.3 New Techniques for Risk Identification 356
    21.3.1 Principal component and discriminant analyses 356
21.3.2 Artificial neural networks and learning algorithms 358
21.3.3 Data partitions and tree-based modeling 359
21.3.4 Pattern matching and optimal set reduction 362
21.4 Comparisons and Integration 363
21.5 Risk Identification for Classified Defect Data 365
21.6 Concluding Remarks 368
Problems 369

22 Software Reliability Engineering 371
22.1 SRE: Basic Concepts and General Approaches 371
22.2 Large Software Systems and Reliability Analyses 372
22.3 Reliability Snapshots Using IDRMs 374
22.4 Longer Term Reliability Analyses Using SRGMs 377
22.5 TBRMs for Reliability Analysis and Improvement 380
   22.5.1 Constructing and using TBRMs 381
   22.5.2 TBRM Applications 382
   22.5.3 TBRM’s impacts on reliability improvement 384
22.6 Implementation and Software Tool Support 385
22.7 SRE: Summary and Perspectives 386
Problems 387

Index 401


