CONTENTS

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

PART I OVERVIEW AND BASICS

1 Overview

1.1 Meeting People’s Quality Expectations 3
1.2 Book Organization and Chapter Overview 6
1.3 Dependency and Suggested Usage 9
1.4 Reader Preparation and Background Knowledge 11
   Problems 13

2 What Is Software Quality?

2.1 Quality: Perspectives and Expectations 15
2.2 Quality Frameworks and ISO-9126 18
2.3 Correctness and Defects: Definitions, Properties, and Measurements 20
2.4 A Historical Perspective of Quality 24
2.5 So, What Is Software Quality? 25
   Problems 26
3 Quality Assurance
  3.1 Classification: QA as Dealing with Defects
  3.2 Defect Prevention
    3.2.1 Education and training
    3.2.2 Formal method
    3.2.3 Other defect prevention techniques
  3.3 Defect Reduction
    3.3.1 Inspection: Direct fault detection and removal
    3.3.2 Testing: Failure observation and fault removal
    3.3.3 Other techniques and risk identification
  3.4 Defect Containment
    3.4.1 Software fault tolerance
    3.4.2 Safety assurance and failure containment
  3.5 Concluding Remarks
  Problems

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

5 Quality Engineering
  5.1 Quality Engineering: Activities and Process
  5.2 Quality Planning: Goal Setting and Strategy Formation
  5.3 Quality Assessment and Improvement
  5.4 Quality Engineering in Software Processes
  5.5 Concluding Remarks
  Problems

PART II SOFTWARE TESTING

  6.1 Purposes, Activities, Processes, and Context
  6.2 Questions About Testing
  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
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

10 Coverage and Usage Testing Based on Finite-State Machines and Markov Chains

10.1 Finite-State Machines 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 validation

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 for Testing

10.4.1 Markov chains and operational profiles

10.4.2 From individual Markov chains to unified Markov models

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

10.6.1 Usage-based web testing: Motivations and basic approach

10.6.2 Constructing UMMs for statistical web testing

10.6.3 Statistical web testing: Details and examples

10.7 Concluding Remarks

Problems

11 Control Flow, Data Dependency, and Interaction Testing

11.1 Basic Control Flow Testing

11.1.1 General concepts
CONTENTS

13.5.3 Process and quality improvement 233
13.6 Concluding Remarks 234
Problems 235

14 Software Inspection 237
14.1 Basic Concepts and Generic Process 237
14.2 Fagan inspection 239
14.3 Other Inspections and Related Activities 242
14.3.1 Inspections of reduced scope or team size 242
14.3.2 Inspections of enlarged scope or team size 243
14.3.3 Informal desk checks, reviews, and walkthroughs 244
14.3.4 Code reading 244
14.3.5 Other formal reviews and static analyses 246
14.4 Defect Detection Techniques, Tool/Process Support, and Effectiveness 247
14.5 Concluding Remarks 249
Problems 250

15 Formal Verification 251
15.1 Basic Concepts: Formal Verification and Formal Specification 251
15.2 Formal Verification: Axiomatic Approach 254
15.2.1 Formal logic specifications 254
15.2.2 Axioms 255
15.2.3 Axiomatic proofs and a comprehensive example 257
15.3 Other Approaches 259
15.3.1 Weakest pre-conditions and backward chaining 260
15.3.2 Functional approach and symbolic execution 260
15.3.3 Seeking alternatives: Model checking and other approaches 261
15.4 Applications, Effectiveness, and Integration Issues 263
15.5 Concluding Remarks 265
Problems 266

16 Fault Tolerance and Failure Containment 267
16.1 Basic Ideas and Concepts 267
16.2 Fault Tolerance with Recovery Blocks 270
16.3 Fault Tolerance with N-Version Programming 272
16.3.1 NVP: Basic technique and implementation 272
16.3.2 Ensuring version independence 273
16.3.3 Applying NVP ideas in other QA activities 274
16.4 Failure Containment: Safety Assurance and Damage Control 275
16.4.1 Hazard analysis using fault-trees and event-trees 275
16.4.2 Hazard resolution for accident prevention 278
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.4.3</td>
<td>Accident analysis and post-accident damage control</td>
<td>278</td>
</tr>
<tr>
<td>16.5</td>
<td>Application in Heterogeneous Systems</td>
<td>279</td>
</tr>
<tr>
<td>16.5.1</td>
<td>Modeling and analyzing heterogeneous systems</td>
<td>279</td>
</tr>
<tr>
<td>16.5.2</td>
<td>Prescriptive specifications for safety</td>
<td>281</td>
</tr>
<tr>
<td>16.6</td>
<td>Concluding Remarks</td>
<td>282</td>
</tr>
<tr>
<td></td>
<td>Problems</td>
<td>282</td>
</tr>
<tr>
<td>17</td>
<td>Comparing Quality Assurance Techniques and Activities</td>
<td>285</td>
</tr>
<tr>
<td>17.1</td>
<td>General Questions: Cost, Benefit, and Environment</td>
<td>285</td>
</tr>
<tr>
<td>17.2</td>
<td>Applicability to Different Environments</td>
<td>289</td>
</tr>
<tr>
<td>17.3</td>
<td>Effectiveness Comparison</td>
<td>291</td>
</tr>
<tr>
<td>17.3.1</td>
<td>Defect perspective</td>
<td>291</td>
</tr>
<tr>
<td>17.3.2</td>
<td>Problem types</td>
<td>292</td>
</tr>
<tr>
<td>17.3.3</td>
<td>Defect level and pervasiveness</td>
<td>293</td>
</tr>
<tr>
<td>17.3.4</td>
<td>Result interpretation and constructive information</td>
<td>294</td>
</tr>
<tr>
<td>17.4</td>
<td>Cost Comparison</td>
<td>295</td>
</tr>
<tr>
<td>17.5</td>
<td>Comparison Summary and Recommendations</td>
<td>297</td>
</tr>
<tr>
<td></td>
<td>Problems</td>
<td>298</td>
</tr>
<tr>
<td>PART IV</td>
<td>QUANTIFIABLE QUALITY IMPROVEMENT</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Feedback Loop and Activities for Quantifiable Quality Improvement</td>
<td>303</td>
</tr>
<tr>
<td>18.1</td>
<td>QA Monitoring and Measurement</td>
<td>304</td>
</tr>
<tr>
<td>18.1.1</td>
<td>Direct vs. indirect quality measurements</td>
<td>304</td>
</tr>
<tr>
<td>18.1.2</td>
<td>Direct quality measurements: Result and defect measurements</td>
<td>306</td>
</tr>
<tr>
<td>18.1.3</td>
<td>Indirect quality measurements: Environmental, product internal, and activity measurements</td>
<td>306</td>
</tr>
<tr>
<td>18.2</td>
<td>Immediate Follow-up Actions and Feedback</td>
<td>308</td>
</tr>
<tr>
<td>18.3</td>
<td>Analyses and Follow-up Actions</td>
<td>309</td>
</tr>
<tr>
<td>18.3.1</td>
<td>Analyses for product release decisions</td>
<td>309</td>
</tr>
<tr>
<td>18.3.2</td>
<td>Analyses for other project management decisions</td>
<td>311</td>
</tr>
<tr>
<td>18.3.3</td>
<td>Other feedback and follow-up actions</td>
<td>312</td>
</tr>
<tr>
<td>18.4</td>
<td>Implementation, Integration, and Tool Support</td>
<td>313</td>
</tr>
<tr>
<td>18.4.1</td>
<td>Feedback loop: Implementation and integration</td>
<td>314</td>
</tr>
<tr>
<td>18.4.2</td>
<td>A refined quality engineering process</td>
<td>316</td>
</tr>
<tr>
<td>18.4.3</td>
<td>Tool support: Strategy, implementation, and integration</td>
<td>317</td>
</tr>
<tr>
<td>18.5</td>
<td>Concluding Remarks</td>
<td>320</td>
</tr>
<tr>
<td></td>
<td>Problems</td>
<td>320</td>
</tr>
<tr>
<td>19</td>
<td>Quality Models and Measurements</td>
<td>323</td>
</tr>
<tr>
<td>19.1</td>
<td>Models for Quality Assessment</td>
<td>323</td>
</tr>
</tbody>
</table>
## CONTENTS

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 362  
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 IDRM s 374  
22.4 Longer-Term Reliability Analyses Using SRGMs 377
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.5 TBRMs for Reliability Analysis and Improvement</td>
<td>380</td>
</tr>
<tr>
<td>22.5.1 Constructing and using TBRMs</td>
<td>381</td>
</tr>
<tr>
<td>22.5.2 TBRM Applications</td>
<td>382</td>
</tr>
<tr>
<td>22.5.3 TBRM’s impacts on reliability improvement</td>
<td>384</td>
</tr>
<tr>
<td>22.6 Implementation and Software Tool Support</td>
<td>385</td>
</tr>
<tr>
<td>22.7 SRE: Summary and Perspectives</td>
<td>386</td>
</tr>
<tr>
<td>Problems</td>
<td>387</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>389</td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>403</td>
</tr>
</tbody>
</table>