

Contents

PREFACE	xiv
PART I INTRODUCTION TO SOFTWARE TESTING	1
1 Software Engineering: A Discipline Like No Other	3
1.1 A Young, Restless Discipline / 3	
1.2 An Industry Under Stress / 5	
1.3 Large, Complex Products / 5	
1.4 Expensive Products / 7	
1.5 Absence of Reuse Practice / 9	
1.6 Fault-Prone Designs / 9	
1.7 Paradoxical Economics / 10	
1.7.1 A Labor-Intensive Industry / 10	
1.7.2 Absence of Automation / 11	
1.7.3 Limited Quality Control / 11	
1.7.4 Unbalanced Lifecycle Costs / 12	
1.7.5 Unbalanced Maintenance Costs / 12	
1.8 Chapter Summary / 13	
1.9 Bibliographic Notes / 13	

2 Software Quality Attributes	14
2.1 Functional Attributes / 15	
2.1.1 Boolean Attributes / 15	
2.1.2 Statistical Attributes / 15	
2.2 Operational Attributes / 17	
2.3 Usability Attributes / 18	
2.4 Business Attributes / 19	
2.5 Structural Attributes / 20	
2.6 Chapter Summary / 21	
2.7 Exercises / 21	
2.8 Bibliographic Notes / 22	
3 A Software Testing Lifecycle	23
3.1 A Software Engineering Lifecycle / 23	
3.2 A Software Testing Lifecycle / 27	
3.3 The V-Model of Software Testing / 32	
3.4 Chapter Summary / 33	
3.5 Bibliographic Notes / 34	
PART II FOUNDATIONS OF SOFTWARE TESTING	35
4 Software Specifications	37
4.1 Principles of Sound Specification / 38	
4.1.1 A Discipline of Specification / 38	
4.2 Relational Mathematics / 39	
4.2.1 Sets and Relations / 39	
4.2.2 Operations on Relations / 39	
4.2.3 Properties of Relations / 41	
4.3 Simple Input Output Programs / 42	
4.3.1 Representing Specifications / 42	
4.3.2 Ordering Specifications / 46	
4.3.3 Specification Generation / 48	
4.3.4 Specification Validation / 53	
4.4 Reliability Versus Safety / 60	
4.5 State-based Systems / 61	
4.5.1 A Relational Model / 62	
4.5.2 Axiomatic Representation / 64	
4.5.3 Specification Validation / 70	
4.6 Chapter Summary / 72	
4.7 Exercises / 72	
4.8 Problems / 76	
4.9 Bibliographic Notes / 78	

5 Program Correctness and Verification	79
5.1 Correctness: A Definition / 80	
5.2 Correctness: Propositions / 83	
5.2.1 Correctness and Refinement / 83	
5.2.2 Set Theoretic Characterizations / 85	
5.2.3 Illustrations / 86	
5.3 Verification / 88	
5.3.1 Sample Formulas / 89	
5.3.2 An Inference System / 91	
5.3.3 Illustrative Examples / 94	
5.4 Chapter Summary / 98	
5.5 Exercises / 99	
5.6 Problems / 100	
5.7 Bibliographic Notes / 100	
6 Failures, Errors, and Faults	101
6.1 Failure, Error, and Fault / 101	
6.2 Faults and Relative Correctness / 103	
6.2.1 Fault, an Evasive Concept / 103	
6.2.2 Relative Correctness / 104	
6.3 Contingent Faults and Definite Faults / 107	
6.3.1 Contingent Faults / 107	
6.3.2 Monotonic Fault Removal / 109	
6.3.3 A Framework for Monotonic Fault Removal / 114	
6.3.4 Definite Faults / 114	
6.4 Fault Management / 116	
6.4.1 Lines of Defense / 116	
6.4.2 Hybrid Validation / 118	
6.5 Chapter Summary / 121	
6.6 Exercises / 122	
6.7 Problems / 123	
6.8 Bibliographic Notes / 124	
7 A Software Testing Taxonomy	125
7.1 The Trouble with Hyphenated Testing / 125	
7.2 A Classification Scheme / 126	
7.2.1 Primary Attributes / 127	
7.2.2 Secondary Attributes / 131	
7.3 Testing Taxonomy / 136	
7.3.1 Unit-Level Testing / 136	
7.3.2 System-Level Testing / 138	
7.4 Exercises / 139	
7.5 Bibliographic Notes / 140	

PART III TEST DATA GENERATION	141
8 Test Generation Concepts	143
8.1 Test Generation and Target Attributes / 143	
8.2 Test Outcomes / 146	
8.3 Test Generation Requirements / 148	
8.4 Test Generation Criteria / 152	
8.5 Empirical Adequacy Assessment / 155	
8.6 Chapter Summary / 160	
8.7 Exercises / 161	
8.8 Bibliographic Notes / 162	
8.9 Appendix: Mutation Program / 163	
9 Functional Criteria	165
9.1 Domain Partitioning / 165	
9.2 Test Data Generation from Tabular Expressions / 171	
9.3 Test Generation for State Based Systems / 176	
9.4 Random Test Data Generation / 184	
9.5 Tourism as a Metaphor for Test Data Selection / 188	
9.6 Chapter Summary / 190	
9.7 Exercises / 190	
9.8 Bibliographic Notes / 192	
10 Structural Criteria	193
10.1 Paths and Path Conditions / 194	
10.1.1 Execution Paths / 194	
10.1.2 Path Functions / 196	
10.1.3 Path Conditions / 201	
10.2 Control Flow Coverage / 202	
10.2.1 Statement Coverage / 202	
10.2.2 Branch Coverage / 204	
10.2.3 Condition Coverage / 207	
10.2.4 Path Coverage / 209	
10.3 Data Flow Coverage / 214	
10.3.1 Definitions and Uses / 214	
10.3.2 Test Generation Criteria / 217	
10.3.3 A Hierarchy of Criteria / 220	
10.4 Fault-Based Test Generation / 220	
10.4.1 Sensitizing Faults / 221	
10.4.2 Selecting Input Data for Fault Sensitization / 225	
10.4.3 Selecting Input Data for Error Propagation / 227	
10.5 Chapter Summary / 228	

- 10.6 Exercises / 229
- 10.7 Bibliographic Notes / 232

PART IV TEST DEPLOYMENT AND ANALYSIS 233

11 Test Oracle Design 235

- 11.1 Dilemmas of Oracle Design / 235
- 11.2 From Specifications to Oracles / 238
- 11.3 Oracles for State-Based Products / 242
 - 11.3.1 From Axioms to Oracles / 243
 - 11.3.2 From Rules to Oracles / 244
- 11.4 Chapter Summary / 250
- 11.5 Exercises / 251

12 Test Driver Design 253

- 12.1 Selecting a Specification / 253
- 12.2 Selecting a Process / 255
- 12.3 Selecting a Specification Model / 257
 - 12.3.1 Random Test Generation / 257
 - 12.3.2 Pre-Generated Test Data / 263
 - 12.3.3 Faults and Fault Detection / 266
- 12.4 Testing by Symbolic Execution / 269
- 12.5 Chapter Summary / 274
- 12.6 Exercises / 275
- 12.7 Bibliographic Notes / 279

13 Test Outcome Analysis 280

- 13.1 Logical Claims / 281
 - 13.1.1 Concrete Testing / 281
 - 13.1.2 Symbolic Testing / 282
 - 13.1.3 Concolic Testing / 283
- 13.2 Stochastic Claims: Fault Density / 284
- 13.3 Stochastic Claims: Failure Probability / 287
 - 13.3.1 Faults Are Not Created Equal / 287
 - 13.3.2 Defining/Quantifying Reliability / 289
 - 13.3.3 Modeling Software Reliability / 291
 - 13.3.4 Certification Testing / 294
 - 13.3.5 Reliability Estimation and Reliability Improvement / 295
 - 13.3.6 Reliability Standards / 299
 - 13.3.7 Reliability as an Economic Function / 300
- 13.4 Chapter Summary / 307

- 13.5 Exercises / 308
- 13.6 Problems / 310
- 13.7 Bibliographic Notes / 310

PART V MANAGEMENT OF SOFTWARE TESTING 311

14 Metrics for Software Testing 313

- 14.1 Fault Proneness / 314
 - 14.1.1 Cyclomatic Complexity / 315
 - 14.1.2 Volume / 316
- 14.2 Fault Detectability / 317
- 14.3 Error Detectability / 320
- 14.4 Error Maskability / 323
- 14.5 Failure Avoidance / 324
- 14.6 Failure Tolerance / 326
- 14.7 An Illustrative Example / 327
 - 14.7.1 Cyclomatic Complexity / 327
 - 14.7.2 Volume / 328
 - 14.7.3 State Redundancy / 328
 - 14.7.4 Functional Redundancy / 328
 - 14.7.5 Non-injectivity / 329
 - 14.7.6 Non-determinacy / 329
 - 14.7.7 Summary / 330
- 14.8 Chapter Summary / 330
- 14.9 Exercises / 331
- 14.10 Bibliographic Notes / 332

15 Software Testing Tools 333

- 15.1 A Classification Scheme / 333
- 15.2 Scripting Tools / 334
 - 15.2.1 CppTest / 334
 - 15.2.2 SilkTest / 335
- 15.3 Record-and-Replay Tools / 336
 - 15.3.1 TestComplete / 336
 - 15.3.2 Selenium IDE / 337
- 15.4 Performance-Testing Tools / 338
 - 15.4.1 LoadRunner / 338
 - 15.4.2 Grinder / 339
 - 15.4.3 QF-Test / 340
 - 15.4.4 Appvance PerformanceCloud / 340
 - 15.4.5 JMeter / 341

15.5 Oracle Design Tools / 342	
15.5.1 JUnit / 342	
15.5.2 TestNG / 343	
15.6 Exception Discovery / 343	
15.6.1 Rational Purify / 343	
15.6.2 Astree / 344	
15.7 Collaborative Tools / 345	
15.7.1 FitNesse / 345	
15.8 Chapter Summary / 345	
16 Testing Product Lines	347
16.1 PLE: A Streamlined Reuse Model / 347	
16.2 Testing Issues / 351	
16.3 Testing Approaches / 353	
16.4 Illustration / 354	
16.4.1 Domain Analysis / 354	
16.4.2 Domain Modeling / 356	
16.4.3 A Reference Architecture / 359	
16.4.4 Domain Implementation / 360	
16.4.5 Testing at Domain Engineering / 365	
16.4.6 Testing at Application Engineering / 369	
16.5 Chapter Summary / 372	
16.6 Exercises / 372	
16.7 Problems / 372	
16.8 Bibliographic References / 373	
BIBLIOGRAPHY	374
INDEX	377