

SOFTWARE TESTING AND ANALYSIS

PROCESS, PRINCIPLES, AND TECHNIQUES



Mauro Pezzè
Michal Young

Contents

List of Figures	xi
List of Tables	xv
I Fundamentals of Test and Analysis	1
1 Software Test and Analysis in a Nutshell	3
1.1 Engineering Processes and Verification	3
1.2 Basic Questions	5
1.3 When Do Verification and Validation Start and End?	5
1.4 What Techniques Should Be Applied?	7
1.5 How Can We Assess the Readiness of a Product?	10
1.6 How Can We Ensure the Quality of Successive Releases?	11
1.7 How Can the Development Process Be Improved?	11
2 A Framework for Test and Analysis	15
2.1 Validation and Verification	15
2.2 Degrees of Freedom	18
2.3 Varieties of Software	23
3 Basic Principles	29
3.1 Sensitivity	29
3.2 Redundancy	32
3.3 Restriction	33
3.4 Partition	35
3.5 Visibility	36
3.6 <i>Feedback</i>	36
4 Test and Analysis Activities Within a Software Process	39
4.1 The Quality Process	39
4.2 Planning and Monitoring	41
4.3 Quality Goals	42
4.4 Dependability Properties	43
4.5 Analysis	46

4.6	Testing	48
4.7	Improving the Process	49
4.8	Organizational Factors	50
II	Basic Techniques	53
5	Finite Models	55
5.1	Overview	55
5.2	Finite Abstractions of Behavior	58
5.3	Control Flow Graphs	59
5.4	Call Graphs	63
5.5	Finite State Machines	65
6	Dependence and Data Flow Models	77
6.1	Definition-Use Pairs	77
6.2	Data Flow Analysis	82
6.3	Classic Analyses: Live and Avail	85
6.4	From Execution to Conservative Flow Analysis	91
6.5	Data Flow Analysis with Arrays and Pointers	94
6.6	Interprocedural Analysis	96
7	Symbolic Execution and Proof of Properties	101
7.1	Symbolic State and Interpretation	102
7.2	Summary Information	104
7.3	Loops and Assertions	105
7.4	Compositional Reasoning	108
7.5	Reasoning about Data Structures and Classes	109
8	Finite State Verification	113
8.1	Overview	113
8.2	State Space Exploration	116
8.3	The State Space Explosion Problem	126
8.4	The Model Correspondence Problem	129
8.5	Granularity of Modeling	131
8.6	Intensional Models	134
8.7	Model Refinement	138
8.8	Data Model Verification with Relational Algebra	140
III	Problems and Methods	149
9	Test Case Selection and Adequacy	151
9.1	Overview	151
9.2	Test Specifications and Cases	152
9.3	Adequacy Criteria	154
9.4	Comparing Criteria	157

10 Functional Testing	161
10.1 Overview	161
10.2 Random versus Partition Testing Strategies	162
10.3 A Systematic Approach	167
10.4 Choosing a Suitable Approach	174
11 Combinatorial Testing	179
11.1 Overview	180
11.2 Category-Partition Testing	180
11.3 Pairwise Combination Testing	188
11.4 Catalog-Based Testing	194
12 Structural Testing	211
12.1 Overview	212
12.2 Statement Testing	215
12.3 Branch Testing	217
12.4 Condition Testing	219
12.5 Path Testing	222
12.6 Procedure Call Testing	229
12.7 Comparing Structural Testing Criteria	230
12.8 The Infeasibility Problem	230
13 Data Flow Testing	235
13.1 Overview	236
13.2 Definition-Use Associations	236
13.3 Data Flow Testing Criteria	239
13.4 Data Flow Coverage with Complex Structures	241
13.5 The Infeasibility Problem	243
14 Model-Based Testing	245
14.1 Overview	245
14.2 Deriving Test Cases from Finite State Machines	246
14.3 Testing Decision Structures	251
14.4 Deriving Test Cases from Control and Data Flow Graphs	257
14.5 Deriving Test Cases from Grammars	257
15 Testing Object-Oriented Software	271
15.1 Overview	271
15.2 Issues in Testing Object-Oriented Software	272
15.3 An Orthogonal Approach to Test	280
15.4 Intra-class Testing	282
15.5 Testing with State Machine Models	282
15.6 Interclass Testing	286
15.7 Structural Testing of Classes	293
15.8 Oracles for Classes	298
15.9 Polymorphism and Dynamic Binding	301

15.10 Inheritance	303
15.11 Genericity	306
15.12 Exceptions	308
16 Fault-Based Testing	313
16.1 Overview	313
16.2 Assumptions in Fault-Based Testing	314
16.3 Mutation Analysis	315
16.4 Fault-Based Adequacy Criteria	319
16.5 Variations on Mutation Analysis	321
17 Test Execution	327
17.1 Overview	327
17.2 From Test Case Specifications to Test Cases	328
17.3 Scaffolding	329
17.4 Generic versus Specific Scaffolding	330
17.5 Test Oracles	332
17.6 Self-Checks as Oracles	334
17.7 Capture and Replay	337
18 Inspection	341
18.1 Overview	341
18.2 The Inspection Team	343
18.3 The Inspection Process	344
18.4 Checklists	345
18.5 Pair Programming	351
19 Program Analysis	355
19.1 Overview	355
19.2 Symbolic Execution in Program Analysis	356
19.3 Symbolic Testing	358
19.4 Summarizing Execution Paths	359
19.5 Memory Analysis	360
19.6 Lockset Analysis	363
19.7 Extracting Behavior Models from Execution	365
IV Process	373
20 Planning and Monitoring the Process	375
20.1 Overview	375
20.2 Quality and Process	376
20.3 Test and Analysis Strategies	377
20.4 Test and Analysis Plans	382
20.5 Risk Planning	386
20.6 Monitoring the Process	389

20.7	Improving the Process	394
20.8	The Quality Team	399
21	Integration and Component-based Software Testing	405
21.1	Overview	405
21.2	Integration Testing Strategies	408
21.3	Testing Components and Assemblies	413
22	System, Acceptance, and Regression Testing	417
22.1	Overview	417
22.2	System Testing	418
22.3	Acceptance Testing	421
22.4	Usability	423
22.5	Regression Testing	427
22.6	Regression Test Selection Techniques	428
22.7	Test Case Prioritization and Selective Execution	434
23	Automating Analysis and Test	439
23.1	Overview	439
23.2	Automation and Planning	441
23.3	Process Management	441
23.4	Static Metrics	443
23.5	Test Case Generation and Execution	445
23.6	Static Analysis and Proof	445
23.7	Cognitive Aids	448
23.8	Version Control	449
23.9	Debugging	449
23.10	Choosing and Integrating Tools	451
24	Documenting Analysis and Test	455
24.1	Overview	455
24.2	Organizing Documents	456
24.3	Test Strategy Document	458
24.4	Analysis and Test Plan	458
24.5	Test Design Specification Documents	460
24.6	Test and Analysis Reports	462
	Bibliography	467
	Index	479