

Contents

	<i>Preface</i>	<i>page xv</i>
	<i>Acknowledgements</i>	<i>xvii</i>
1	Introduction to Software Testing	1
	1.1 The Software Industry	1
	1.1.1 Software Testing and Quality	3
	1.1.2 Software Testing and Risk Management	3
	1.2 Mistakes, Faults, and Failures	4
	1.2.1 Mistakes	5
	1.2.2 Software Faults	5
	1.2.3 Software Failures	6
	1.2.4 Need for Testing	8
	1.3 The Role of Specifications	9
	1.4 Manual Test Example	10
	1.5 The Theory of Software Testing	11
	1.6 Exhaustive Testing	11
	1.6.1 Exhaustive Test Data	11
	1.6.2 Feasibility of Exhaustive Testing	12
	1.7 Test Heuristics	12
	1.7.1 Random Testing	13
	1.7.2 Black-Box and White-Box Testing	14
	1.7.3 Experience-Based Testing	15
	1.7.4 Fault Insertion	15
	1.8 When to Finish Testing	15
	1.9 Static and Dynamic Testing	16
	1.9.1 Review-Based Techniques	16
	1.9.2 Program Proving	17
	1.10 Testing in the Software Development Process	17
	1.11 Software Testing Activities	19
	1.11.1 Analysis	19
	1.11.2 Test Coverage Items	20
	1.11.3 Test Cases	21
	1.11.4 Test Design Verification	22
	1.11.5 Test Implementation	22

1.12	Test Artefacts	23
1.13	Fault Models	24
1.14	Using this Book	24
1.14.1	Book Structure	24
1.14.2	Order of Testing	25
1.14.3	Documenting Test Design	25
1.14.4	The Programming Language	25
1.14.5	Level of Detail	26
1.14.6	Examples	26
1.14.7	Software Test Tools	26
1.15	Notes on Terminology	26
2	Equivalence Partitions	27
2.1	Testing with Equivalence Partitions	27
2.2	Example	27
2.2.1	Analysis: Identifying the Equivalence Partitions	28
2.2.2	Test Coverage Items	32
2.2.3	Test Cases	32
2.2.4	Verification of the Test Cases	35
2.3	Test Implementation and Results	36
2.3.1	Manual Test Output	36
2.3.2	Automated Test Implementation	37
2.3.3	Test Results	39
2.4	Equivalence Partitions in More Detail	40
2.4.1	Fault Model	40
2.4.2	Description	40
2.4.3	Analysis: Identifying Equivalence Partitions	40
2.4.4	Test Coverage Items	42
2.4.5	Test Cases	42
2.4.6	Pitfalls	42
2.5	Evaluation	43
2.5.1	Limitations	43
2.5.2	Strengths and Weaknesses	46
2.6	Key Points	47
2.7	Notes for Experienced Testers	47
3	Boundary Value Analysis	48
3.1	Testing with Boundary Value Analysis	48
3.2	Example	48
3.2.1	Analysis: Identifying the Boundary Values	48
3.2.2	Test Coverage Items	49
3.2.3	Test Cases	49
3.2.4	Verification of the Test Cases	51

	Contents	vii
3.3	Test Implementation and Results	52
3.3.1	Implementation	52
3.3.2	Test Results	53
3.4	Boundary Value Analysis in More Detail	54
3.4.1	Fault Model	54
3.4.2	Description	54
3.4.3	Analysis: Identifying Boundary Values	54
3.4.4	Test Coverage Items	55
3.4.5	Test Cases	55
3.4.6	Pitfalls	55
3.5	Evaluation	56
3.5.1	Limitations	56
3.5.2	Strengths and Weaknesses	58
3.6	Key Points	59
3.7	Notes for Experienced Testers	59
4	Decision Table Testing	60
4.1	Testing Combinations with Decision Tables	60
4.2	Example	60
4.2.1	Analysis	60
4.2.2	Test Coverage Items	67
4.2.3	Test Cases	67
4.2.4	Verification of the Test Cases	69
4.3	Test Implementation and Results	70
4.3.1	Implementation	70
4.3.2	Test Results	71
4.4	A More Detailed Look at Testing with Decision Tables	71
4.4.1	Fault Model	71
4.4.2	Description	72
4.4.3	Analysis: Developing Decision Tables	72
4.4.4	Test Coverage Items	81
4.4.5	Test Cases	82
4.4.6	Pitfalls	82
4.5	Evaluation	82
4.5.1	Limitations	83
4.5.2	Strengths and Weaknesses	85
4.6	Key Points	85
4.7	Notes for Experienced Testers	85
5	Statement Coverage	86
5.1	Introduction to White-Box Testing	86
5.2	Testing with Statement Coverage	86
5.2.1	Statement Coverage Measurement	87

5.3	Example	87
5.3.1	Analysis: Identifying Unexecuted Statements	88
5.3.2	Test Coverage Items	90
5.3.3	Test Cases	91
5.3.4	Verification of the Test Cases	91
5.4	Test Implementation and Results	92
5.4.1	Implementation	92
5.4.2	Test Results	93
5.5	Statement Coverage Testing in More Detail	94
5.5.1	Fault Model	94
5.5.2	Description	94
5.5.3	Analysis: Identifying Unexecuted Statements	94
5.5.4	Test Coverage Items	94
5.5.5	Test Cases	94
5.6	Evaluation	95
5.6.1	Limitations	95
5.6.2	Strengths and Weaknesses	97
5.7	Key Points	98
5.8	Notes for Experienced Testers	98
6	Branch Coverage	99
6.1	Testing with Branch Coverage	99
6.1.1	Branch Coverage Measurement	99
6.2	Example	99
6.2.1	Analysis: Identifying Untaken Branches	100
6.2.2	Test Coverage Items	103
6.2.3	Test Cases	103
6.2.4	Verification of the Test Cases	103
6.3	Test Implementation and Results	104
6.3.1	Implementation	104
6.3.2	Test Results	105
6.4	Branch Coverage in More Detail	107
6.4.1	Fault Model	107
6.4.2	Description	107
6.4.3	Goal	108
6.4.4	Analysis: Identifying Untaken Branches	109
6.4.5	Test Coverage Item	109
6.4.6	Test Cases	109
6.5	Evaluation	109
6.5.1	Limitations	109
6.5.2	Strengths and Weaknesses	111
6.6	Key Points	112
6.7	Notes for Experienced Testers	112

7	All Paths Coverage	113
	7.1 Testing with All Paths Coverage	113
	7.2 Example	114
	7.2.1 Code Analysis	114
	7.2.2 Test Coverage Items	119
	7.2.3 Test Cases	119
	7.2.4 Verification of the Test Cases	120
	7.3 Test Implementation and Results	120
	7.3.1 Implementation	120
	7.3.2 Test Results	121
	7.4 All Paths Coverage Testing in More Detail	123
	7.4.1 Fault Model	123
	7.4.2 Description	123
	7.4.3 Analysis: Developing Control-Flow Graphs	123
	7.4.4 Analysis: Identifying End-to-End Paths	128
	7.4.5 The Possible Paths in Fault 6	131
	7.4.6 Test Coverage Items	136
	7.4.7 Test Cases	136
	7.5 Evaluation	136
	7.5.1 Limitations	136
	7.5.2 Strengths and Weaknesses	139
	7.6 Key Points	139
	7.7 Notes for the Experienced Tester	140
8	Black-Box and White-Box Testing	141
	8.1 Comparison of Black-Box and White-Box Testing	142
	8.1.1 Black-Box Testing	143
	8.1.2 White-Box Testing	143
	8.1.3 Errors of <i>Omission</i> and <i>Commission</i>	145
	8.1.4 Usage	146
	8.2 Black-Box Testing: Additional Matters	147
	8.2.1 Strings and Arrays	147
	8.2.2 Discontinuous Input Partitions	148
	8.2.3 Overlapping Output Partitions	149
	8.2.4 In-band Error Reporting	150
	8.2.5 Handling Relative Values	151
	8.2.6 Classic Triangle Problem	152
	8.2.7 Testing Sequences of Inputs/State-Based Testing	153
	8.2.8 Floating Point Numbers	155
	8.2.9 Numeric Processing	156
	8.3 White-Box Testing: Some More Techniques	157
	8.3.1 Dataflow Coverage/Definition–Use Pairs	157
	8.3.2 Condition Coverage	158

8.3.3	Decision Coverage	159
8.3.4	Decision Condition Coverage	159
8.3.5	Multiple Condition Coverage	160
8.3.6	Modified Condition/Decision Coverage	161
8.3.7	Test Ranking	162
8.4	Repair-Based Testing	163
8.4.1	Specific Repair Test	163
8.4.2	Generic Repair Test	163
8.4.3	Abstracted Repair Test	163
8.4.4	Example	163
8.4.5	Using Repair-Based Tests	164
9	Testing Object-Oriented Software	165
9.1	Testing in Class Context	165
9.2	Example	165
9.2.1	Analysis	168
9.2.2	Deciding on Which Methods to Test	168
9.2.3	Selecting a Test Technique	169
9.2.4	Test Coverage Items	171
9.2.5	Test Cases	171
9.2.6	Verification of the Test Cases	172
9.3	Test Implementation and Results	174
9.3.1	Test Implementation	174
9.3.2	Test Results	175
9.4	A More Detailed Look at Testing Object-Oriented Software	176
9.4.1	Object-Oriented Programming	176
9.4.2	Testing Object-Oriented Software	177
9.4.3	Fault Models	177
9.4.4	Testing in Class Context	178
9.4.5	Analysis for OO Testing	180
9.4.6	Test Coverage Items	180
9.4.7	Test Cases	180
9.4.8	Test Implementation	181
9.4.9	Overview of Advanced OO Testing	181
9.4.10	Inheritance Testing	181
9.4.11	State-Based Testing	182
9.4.12	UML-Based Testing	185
9.4.13	Built-In Testing	185
9.5	Evaluation	187
9.5.1	Limitations	187
9.5.2	Simple Typo Fault	187
9.5.3	State-Based Fault	189
9.5.4	Inheritance Fault	190
9.5.5	Strengths and Weaknesses	191

	9.6 Key Points	192
	9.7 Notes for Experienced Testers	192
10	Application Testing	193
	10.1 Testing Web Applications with User Stories	193
	10.2 Example	193
	10.2.1 Analysis	194
	10.2.2 Test Coverage Items	199
	10.2.3 Test Cases	200
	10.2.4 Verification of the Test Cases	202
	10.3 Test Implementation and Results	202
	10.3.1 Implementation	202
	10.3.2 Test Results	207
	10.4 Application Testing in More Detail	207
	10.4.1 System Test Model	207
	10.4.2 Application Test Model	208
	10.4.3 System Testing and Integration Testing	209
	10.4.4 Fault Models for Application Testing	211
	10.4.5 Analysis	213
	10.4.6 Test Coverage Items	217
	10.4.7 Test Cases	217
	10.4.8 Implementation	217
	10.4.9 Interacting with HTML Elements	220
	10.4.10 Test Output Messages	220
	10.4.11 Record and Playback Testing	221
	10.5 Evaluation	222
	10.5.1 Limitations	222
	10.5.2 Strengths and Weaknesses	226
	10.6 Key Points	226
	10.7 Key Differences between Unit Testing and Application Testing	227
	10.8 Notes for Experienced Testers	228
11	Test Automation	230
	11.1 Introduction	230
	11.1.1 Interpreting Test Results	231
	11.1.2 Documenting Automated Tests	231
	11.1.3 Software Test Automation and Version Control	232
	11.2 Test Frameworks: TestNG	232
	11.2.1 A Detailed Look at a TestNG Example	233
	11.3 Organising Automated Test Code	235
	11.3.1 Organising TestNG Tests with an XML File	235
	11.4 Setup and Cleanup Methods	237
	11.5 Inline Tests vs Parameterised Tests	239

11.6	Test Coverage Measurement	242
11.7	Timeouts	244
11.8	Exceptions	245
11.9	Inheritance Testing	247
11.9.1	Using the Class Name	248
11.9.2	Inheriting Superclass Tests	250
11.9.3	Inheritance Test Selection	253
11.10	Interfacing to Web Applications	255
11.11	Interfacing to Desktop Applications	257
11.12	Interfacing to Mobile Applications	257
12	Random Testing	259
12.1	Introduction to Random Testing	259
12.2	Random Data Selection	259
12.3	Unit Test Example	260
12.3.1	Test Coverage Items	260
12.3.2	Test Cases	261
12.3.3	Test Implementation	262
12.3.4	Test Results	263
12.4	Application Testing Example	264
12.4.1	Analysis	264
12.4.2	Test Coverage Items	265
12.4.3	Test Cases	265
12.4.4	Test Implementation	265
12.4.5	Test Results	270
12.5	Random Testing in More Detail	270
12.5.1	Barriers to Full Automation	270
12.5.2	Overview of Other Types of Random Testing	273
12.6	Evaluation	274
12.6.1	Limitations	274
12.6.2	Strengths and Weaknesses	275
12.7	Key Points	276
12.8	Notes for the Experienced Tester	277
13	Testing in the Software Process	278
13.1	Test Planning	279
13.2	Software Development Life Cycle	279
13.3	The Waterfall Model	280
13.4	The V-Model	281
13.5	Incremental and Agile Development	282
13.5.1	Incremental Development	283
13.5.2	Extreme Programming	284
13.5.3	Scrum	285
13.6	Process-Related Quality Standards and Models	286

14	Wrap Up	287
	14.1 Summary	287
	14.2 A Reverse Look at Testing	289
	14.2.1 A Test Implementation	289
	14.2.2 Test Cases	289
	14.2.3 Test Coverage Items	290
	14.2.4 Test Analysis	290
	14.2.5 The Software Specification	291
	14.2.6 Discussion	291
	14.3 Further Reading	292
	14.3.1 Testing in the Software Process	292
	14.3.2 Standards on Software Testing	292
	14.3.3 Software Testing Techniques	293
	14.3.4 Testing Object-Oriented Software	293
	14.3.5 Integration Testing	293
	14.3.6 Random Testing	294
	14.3.7 Testing for Language-Specific Hazards	294
	14.3.8 Program Proving	295
	14.3.9 Testing Safety-Critical Software	295
	14.3.10 Going <i>Off-Piste</i>	296
	14.4 Research Directions	296
	14.4.1 Conferences	296
	14.4.2 Research Topics	297
	 <i>Appendix: Running the Examples</i>	 298
	<i>Bibliography</i>	299
	<i>Listing</i>	301
	<i>Index</i>	303