

SOFTWARE TESTING

A Craftsman's Approach

SECOND EDITION



Paul C. Jorgensen



CRC PRESS

Contents

PART I A MATHEMATICAL CONTEXT

1	A Perspective on Testing.....	3
1.1	Basic Definitions	3
1.2	Test Cases	4
1.3	Insights from a Venn Diagram.....	5
1.4	Identifying Test Cases.....	7
1.4.1	Functional Testing	7
1.4.2	Structural Testing	8
1.4.3	The Functional versus Structural Debate	9
1.5	Error and Fault Taxonomies	10
1.6	Levels of Testing	13
	References	13
	Exercises.....	14
2	Examples	15
2.1	Generalized Pseudocode	15
2.2	The Triangle Problem.....	17
2.2.1	Problem Statements	17
2.2.2	Discussion	17
2.2.3	Traditional Implementation.....	18
2.2.4	Structured Implementation	20
2.3	The NextDate Function	22
2.3.1	Problem Statements	22
2.3.2	Discussion	22
2.3.3	Implementation	22
2.4	The Commission Problem.....	25
2.4.1	Problem Statement	25
2.4.2	Discussion	25
2.4.3	Implementation	26
2.5	The SATM System	26
2.5.1	Problem Statement	27
2.5.2	Discussion	29
2.6	The Currency Converter	29
2.7	Saturn Windshield Wiper Controller	30

References	30
Exercises	31
3 Discrete Math for Testers	33
3.1 Set Theory	33
3.1.1 Set Membership	34
3.1.2 Set Definition	34
3.1.3 The Empty Set	35
3.1.4 Venn Diagrams	35
3.1.5 Set Operations	36
3.1.6 Set Relations	38
3.1.7 Set Partitions	38
3.1.8 Set Identities	39
3.2 Functions	40
3.2.1 Domain and Range	40
3.2.2 Function Types	41
3.2.3 Function Composition	42
3.3 Relations	43
3.3.1 Relations among Sets	43
3.3.2 Relations on a Single Set	45
3.4 Propositional Logic	46
3.4.1 Logical Operators	46
3.4.2 Logical Expressions	47
3.4.3 Logical Equivalence	47
3.5 Probability Theory	48
Reference	50
Exercises	50
4 Graph Theory for Testers	53
4.1 Graphs	53
4.1.1 Degree of a Node	54
4.1.2 Incidence Matrices	55
4.1.3 Adjacency Matrices	55
4.1.4 Paths	56
4.1.5 Connectedness	57
4.1.6 Condensation Graphs	57
4.1.7 Cyclomatic Number	58
4.2 Directed Graphs	59
4.2.1 Indegrees and Outdegrees	60
4.2.2 Types of Nodes	60
4.2.3 Adjacency Matrix of a Directed Graph	61
4.2.4 Paths and Semipaths	61
4.2.5 Reachability Matrix	62
4.2.6 n-Connectedness	63
4.2.7 Strong Components	63
4.3 Graphs for Testing	64
4.3.1 Program Graphs	65
4.3.2 Finite State Machines	66
4.3.3 Petri Nets	67
4.3.4 Event-Driven Petri Nets	70
4.3.5 StateCharts	73
Reference	75
Exercises	75

PART II FUNCTIONAL TESTING

5	Boundary Value Testing.....	79
5.1	Boundary Value Analysis.....	79
5.1.1	Generalizing Boundary Value Analysis.....	81
5.1.2	Limitations of Boundary Value Analysis.....	82
5.2	Robustness Testing.....	82
5.3	Worst-Case Testing.....	83
5.4	Special Value Testing.....	84
5.5	Examples.....	84
5.5.1	Test Cases for the Triangle Problem.....	85
5.5.2	Test Cases for the NextDate Function.....	88
5.5.3	Test Cases for the Commission Problem.....	91
5.6	Random Testing.....	93
5.7	Guidelines for Boundary Value Testing.....	94
	Exercises.....	95
6	Equivalence Class Testing.....	97
6.1	Equivalence Classes.....	97
6.1.1	Weak Normal Equivalence Class Testing.....	98
6.1.2	Strong Normal Equivalence Class Testing.....	98
6.1.3	Weak Robust Equivalence Class Testing.....	99
6.1.4	Strong Robust Equivalence Class Testing.....	100
6.2	Equivalence Class Test Cases for the Triangle Problem.....	101
6.3	Equivalence Class Test Cases for the NextDate Function.....	103
6.3.1	Equivalence Class Test Cases.....	105
6.4	Equivalence Class Test Cases for the Commission Problem.....	106
6.4.1	Output Range Equivalence Class Test Cases.....	107
6.4.2	Output Range Equivalence Class Test Cases.....	108
6.5	Guidelines and Observations.....	108
	References.....	109
	Exercises.....	109
7	Decision Table-Based Testing.....	111
7.1	Decision Tables.....	111
7.1.2	Technique.....	112
7.2	Test Cases for the Triangle Problem.....	116
7.3	Test Cases for the NextDate Function.....	116
7.3.1	First Try.....	116
7.3.2	Second Try.....	117
7.3.3	Third Try.....	119
7.4	Test Cases for the Commission Problem.....	120
7.5	Guidelines and Observations.....	120
	References.....	122
	Exercises.....	122
8	Retrospective on Functional Testing.....	123
8.1	Testing Effort.....	123
8.2	Testing Efficiency.....	126
8.3	Testing Effectiveness.....	127
8.4	Guidelines.....	128
8.5	Case Study.....	129

PART III STRUCTURAL TESTING

9	Path Testing.....	137
9.1	DD-Paths.....	138
9.2	Test Coverage Metrics.....	142
9.2.1	Metric-Based Testing.....	143
9.2.2	Test Coverage Analyzers.....	145
9.3	Basis Path Testing.....	145
9.3.1	McCabe's Basis Path Method.....	146
9.3.2	Observations on McCabe's Basis Path Method.....	149
9.3.3	Essential Complexity.....	150
9.4	Guidelines and Observations.....	153
	References.....	155
	Exercises.....	155
10	Data Flow Testing	157
10.1	Define/Use Testing.....	158
10.1.1	Example.....	159
10.1.2	du-Paths for Stocks.....	163
10.1.3	du-Paths for Locks.....	164
10.1.4	du-Paths for totalLocks.....	164
10.1.5	du-Paths for Sales.....	165
10.1.6	du-Paths for Commission.....	165
10.1.7	du-Path Test Coverage Metrics.....	166
10.2	Slice-Based Testing.....	167
10.2.1	Example.....	169
10.2.2	Style and Technique.....	172
10.3	Guidelines and Observations.....	173
	References.....	174
	Exercises.....	174
11	Retrospective on Structural Testing.....	175
11.1	Gaps and Redundancies.....	176
11.2	Metrics for Method Evaluation.....	176
11.3	Case Study Revisited.....	179
11.3.1	Path-Based Testing.....	182
11.3.2	Data Flow Testing.....	182
11.3.3	Slice Testing.....	182
	References.....	183
	Exercises.....	183

PART IV INTEGRATION AND SYSTEM TESTING

12	Levels of Testing	187
12.1	Traditional View of Testing Levels.....	187
12.2	Alternative Life Cycle Models.....	189
12.2.1	Waterfall Spin-Offs.....	189
12.2.2	Specification-Based Life Cycle Models.....	190
12.3	The SATM System.....	192
12.4	Separating Integration and System Testing.....	201
12.4.1	Structural Insights.....	202
12.4.2	Behavioral Insights.....	203
	References.....	204

13	Integration Testing	205
13.1	A Closer Look at the SATM System	205
13.2	Decomposition-Based Integration.....	207
13.2.1	Top-down Integration	209
13.2.2	Bottom-up Integration.....	211
13.2.3	Sandwich Integration	212
13.2.4	Pros and Cons	212
13.3	Call Graph-Based Integration.....	212
13.3.1	Pair-Wise Integration.....	213
13.3.2	Neighborhood Integration	214
13.3.3	Pros and Cons	215
13.4	Path-Based Integration.....	216
13.4.1	New and Extended Concepts.....	216
13.4.2	MM-Paths in the SATM System	219
13.4.4	MM-Path Complexity.....	222
13.4.5	Pros and Cons	223
13.5	Case Study	224
13.5.1	Decomposition Based Integration.....	228
13.5.2	Call Graph Based Integration.....	228
13.5.3	MM-Path Based Integration	228
	Reference.....	229
	Exercises.....	229
14	System Testing.....	231
14.1	Threads	231
14.1.1	Thread Possibilities.....	232
14.1.2	Thread Definitions.....	233
14.2	Basis Concepts for Requirements Specification.....	235
14.2.1	Data	235
14.2.2	Actions.....	236
14.2.3	Devices.....	236
14.2.4	Events	237
14.2.5	Threads.....	237
14.2.6	Relationships among Basis Concepts	237
14.2.7	Modeling with the Basis Concepts	238
14.3	Finding Threads	240
14.4	Structural Strategies for Thread Testing.....	242
14.4.1	Bottom-up Threads	244
14.4.2	Node and Edge Coverage Metrics	245
14.5	Functional Strategies for Thread Testing	246
14.5.1	Event-Based Thread Testing.....	246
14.5.2	Port-Based Thread Testing.....	248
14.5.3	Data-Based Thread Testing.....	248
14.6	SATM Test Threads	250
14.7	System Testing Guidelines	255
14.7.1	Pseudostructural System Testing.....	255
14.7.2	Operational Profiles.....	256
14.7.3	Progression vs. Regression Testing.....	258
	References	258
	Exercises.....	258
15	Interaction Testing.....	261
15.1	Context of Interaction.....	261

15.2	A Taxonomy of Interactions	263
15.2.1	Static Interactions in a Single Processor.....	264
15.2.2	Static Interactions in Multiple Processors.....	266
15.2.3	Dynamic Interactions in a Single Processor	266
15.2.4	Dynamic Interactions in Multiple Processors.....	272
15.3	Interaction, Composition, and Determinism.....	278
15.4	Client-Server Testing	281
	References	283
	Exercises	283

PART V OBJECT-ORIENTED TESTING

16	Issues in Object-Oriented Testing	287
16.1	Units for Object-Oriented Testing	288
16.2	Implications of Composition and Encapsulation.....	288
16.3	Implications of Inheritance	290
16.4	Implications of Polymorphism.....	292
16.5	Levels of Object-Oriented Testing	292
16.6	GUI Testing	292
16.7	Data Flow Testing for Object-Oriented Software.....	292
16.8	Examples for Part V.....	292
16.8.1	The Object-Oriented Calendar	293
16.8.2	The Currency Conversion Application	294
	References	298
	Exercises	298
17	Class Testing.....	299
17.1	Methods as Units	299
17.1.1	Pseudocode for o-oCalendar	300
17.1.1.1	Class: CalendarUnit	300
17.1.1.2	Class: testIt.....	300
17.1.1.3	Class: Date	301
17.1.1.4	Class: Day	302
17.1.1.5	Class: Month	303
17.1.1.6	Class: Year	304
17.1.2	Unit Testing for Date.increment.....	305
17.2	Classes as Units.....	305
17.2.1	Pseudocode for the windshieldWiper Class	306
17.2.2	Unit Testing for the windshieldWiper Class	307
18	Object-Oriented Integration Testing	311
18.1	UML Support for Integration Testing	311
18.2	MM-Paths for Object-Oriented Software	314
18.2.1	Pseudocode for o-oCalendar	314
18.3	A Framework for Object-Oriented Data Flow Integration Testing.....	320
18.3.1	Event- and Message-Driven Petri Nets	320
18.3.2	Inheritance-Induced Data Flow	322
18.3.3	Message-Induced Data Flow	323
18.3.4	Slices?.....	323
	Exercises	324
	Reference	326
19	GUI Testing.....	327
19.1	The Currency Conversion Program.....	327

19.2	Unit Testing for the Currency Conversion Program	331
19.3	Integration Testing for the Currency Conversion Program	332
19.4	System Testing for the Currency Conversion Program	334
	Exercises	338
20	Object-Oriented System Testing	339
20.1	Currency Converter UML Description	339
20.1.1	Problem Statement	339
20.1.2	System Functions	340
20.1.3	Presentation Layer	340
20.1.4	High-Level Use Cases	341
20.1.5	Essential Use Cases	341
20.1.6	Detailed GUI Definition	343
20.1.7	Expanded Essential Use Cases	343
20.1.8	Real Use Cases	348
20.2	UML-Based System Testing	348
20.3	StateChart-Based System Testing	350
	References	351
Index		353