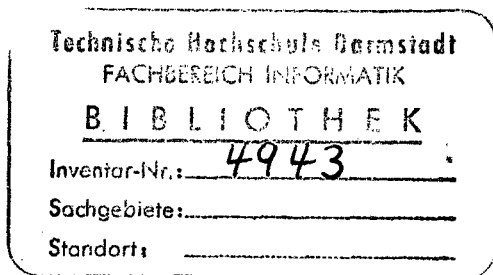


Software Reliability

Principles and Practices

Glenford J. Myers

Staff Member,
IBM Systems Research Institute
Lecturer in Computer Science,
Polytechnic Institute of New York



A WILEY-INTERSCIENCE PUBLICATION

JOHN WILEY & SONS

New York • Chichester • Brisbane • Toronto • Singapore

Contents

PART 1

Concepts of Software Reliability

1. Definition of Software Reliability 3
 - Is the Moon an Enemy Rocket?* 4
 - What is an Error?* 4
 - What is Reliability?* 7
 - Are Engineers Smarter than Programmers?* 8
 - Hardware Reliability,* 9
 - The High Cost of Software,* 11
 - Software Experimentation,* 12
 - References,* 14
2. Errors—Their Causes and Consequences 15
 - The Macroscopic Translation Model,* 17
 - The Microscopic Translation Model,* 21
 - Consequences of Errors,* 23
 - References,* 25

PART 2

Designing Reliable Software

3. Basic Design Principles 29
 - Four Approaches to Reliability,* 30
 - The Design Processes,* 34
 - Complexity,* 37
 - The User Relationship,* 38
 - Problem Solving,* 40

	<i>Design Correctness,</i>	43
	<i>References,</i>	45
4.	Requirements, Objectives, and Specifications	46
	<i>Establishing Requirements,</i>	46
	<i>Software Objectives,</i>	48
	<i>Understanding the Tradeoffs,</i>	51
	<i>Writing Software Objectives,</i>	56
	<i>External Design,</i>	59
	<i>Designing the User Interface,</i>	61
	<i>Writing the External Specification,</i>	65
	<i>Verifying the External Specification,</i>	69
	<i>Planning for Change,</i>	72
	<i>References,</i>	73
5.	System Architecture	74
	<i>Levels of Abstraction,</i>	75
	<i>Port-Driven Subsystems,</i>	79
	<i>Sequential Programs,</i>	85
	<i>Documentation,</i>	85
	<i>Verification,</i>	86
	<i>References,</i>	87
6.	Program Structure Design	88
	<i>Module Independence,</i>	89
	<i>Module Strength,</i>	89
	<i>Module Coupling,</i>	92
	<i>Further Guidelines,</i>	95
	<i>Composite Analysis,</i>	97
	<i>Composite Analysis Example,</i>	98
	<i>Verification,</i>	108
	<i>References,</i>	109
7.	Design Practices	110
	<i>Passive Fault Detection,</i>	110
	<i>Active Fault Detection,</i>	114
	<i>Fault Correction and Tolerance,</i>	116
	<i>Error Isolation,</i>	118
	<i>Handling Hardware Failures,</i>	120
	<i>References,</i>	121

8. Module Design and Coding	123
<i>Module External Design,</i>	124
<i>Module Logic Design,</i>	126
<i>Structured Programming and Step-Wise Refinement,</i>	129
<i>Defensive Programming,</i>	138
<i>High-Level Languages,</i>	139
<i>Attitude,</i>	143
<i>Documentation,</i>	145
<i>Standards,</i>	146
<i>Verification,</i>	147
<i>References,</i>	149
9. Programming Style	151
<i>Program Clarity,</i>	152
<i>Using the Language,</i>	155
<i>Microefficiencies,</i>	156
<i>Comments,</i>	160
<i>Data Definitions,</i>	161
<i>Program Structure,</i>	164
<i>References,</i>	165

PART 3

Software Testing

10. Testing Principles	169
<i>Basic Definitions,</i>	172
<i>Testing Philosophies,</i>	173
<i>Module Integration,</i>	178
<i>Bottom-Up Testing,</i>	178
<i>Top-Down Testing,</i>	181
<i>Modified Top-Down Testing,</i>	184
<i>Big-Bang Testing,</i>	185
<i>Sandwich Testing,</i>	186
<i>Modified Sandwich Testing,</i>	187
<i>Which One is Best?</i>	187
<i>Testing Axioms,</i>	189
<i>References,</i>	195

11. Module Testing	196
<i>Test Case Design,</i>	196
<i>Test Execution,</i>	206
<i>Module Driver Tools,</i>	208
<i>Static Flow Analysis,</i>	212
<i>Run-Time Tools,</i>	212
<i>Quadratic Equation Tests,</i>	214
<i>References,</i>	215
12. Function and System Testing	216
<i>External Function Testing,</i>	216
<i>Cause-Effect Graphing,</i>	218
<i>System Integration,</i>	227
<i>System Testing,</i>	230
<i>Function and System Testing Tools,</i>	238
<i>Test Planning and Control,</i>	242
<i>Acceptance Testing,</i>	244
<i>Installation Testing,</i>	245
<i>References,</i>	245
13. Debugging	247
<i>Finding the Error,</i>	247
<i>Making the Correction,</i>	251
<i>Debugging Tools,</i>	252
<i>Monitoring the Debugging Process,</i>	257
<i>References,</i>	258

PART 4

Additional Topics in Software Reliability

14. Management Techniques for Reliability	263
<i>Organization and Staffing,</i>	264
<i>Programming Librarians,</i>	267
<i>Programming Teams,</i>	269
<i>The Elements of a Good Manager,</i>	272
<i>References,</i>	273
15. Programming Languages and Reliability	275
<i>Uniformity,</i>	278
<i>Simplicity,</i>	281

	<i>Data Declaration,</i>	282
	<i>Data Types and Conversion,</i>	284
	<i>Procedures and Data Scoping,</i>	287
	<i>Control Structures,</i>	289
	<i>Data Operations,</i>	291
	<i>Compile-Time Error Detection,</i>	292
	<i>Execution-Time Error Detection,</i>	295
	<i>References,</i>	296
16.	Computer Architecture and Reliability	298
	<i>Memory Structure,</i>	300
	<i>Program Structure,</i>	304
	<i>Debugging Aids,</i>	306
	<i>High-Level Language Machines,</i>	307
	<i>References,</i>	308
17.	Proving Program Correctness	310
	<i>The Method of Inductive Assertions,</i>	311
	<i>What Can and Cannot be Proved by Proofs,</i>	319
	<i>Formal and Automated Proofs,</i>	322
	<i>Additional Applications of Proof Techniques,</i>	324
	<i>References,</i>	327
18.	Reliability Models	329
	<i>A Reliability Growth Model,</i>	330
	<i>Other Probabilistic Models</i>	335
	<i>An Error Seeding Model,</i>	336
	<i>Simple Intuitive Models,</i>	338
	<i>Complexity Models,</i>	340
	<i>References,</i>	342
19.	Software Support Systems	344
	<i>Program Library Systems,</i>	344
	<i>Design Aids,</i>	351
	<i>References,</i>	353
	Index	355