

Contents

List of Tables	xv
List of Figures	xvii
Foreword	xxiii
Preface	xxv
1 Introduction	1
1.1 Software Quality	1
1.1.1 Quality Through the Eyes of the User, the Builder, and the Manager	2
1.1.2 Quality Attributes	4
1.1.3 A World of Tensions	7
1.2 How to Read This Book	9
1.2.1 Typographical Conventions	10
1.2.2 Diagrams	11
1.2.3 Charts	13
1.2.4 Assembly Code	13
1.2.5 Exercises	14
1.2.6 Supplementary Material	14
1.2.7 Tools	14
2 Reliability	17
2.1 Input Problems	17
2.2 Output Problems	21
2.2.1 Incomplete or Missing Output	21
2.2.2 Correct Results at the Wrong Time	23

2.2.3	Wrong Format	24
2.3	Logic Problems	26
2.3.1	Off-by-One Errors and Loop Iterations	26
2.3.2	Neglected Extreme Conditions	27
2.3.3	Forgotten Cases, Condition Tests, or Steps	29
2.3.4	Missing Methods	34
2.3.5	Unnecessary Functionality	37
2.3.6	Misinterpretation	40
2.4	Computation Problems	42
2.4.1	Incorrect Algorithm or Computation	42
2.4.2	Incorrect Operand in an Expression	43
2.4.3	Incorrect Operator in an Expression	47
2.4.4	Operator Precedence Problems	48
2.4.5	Overflow, Underflow, and Sign Conversion-Errors	49
2.5	Concurrency and Timing Problems	51
2.6	Interface Problems	56
2.6.1	Incorrect Routine or Arguments	57
2.6.2	Failure to Test a Return Value	59
2.6.3	Missing Error Detection or Recovery	62
2.6.4	Resource Leaks	65
2.6.5	Misuse of Object-Oriented Facilities	68
2.7	Data-Handling Problems	69
2.7.1	Incorrect Data Initialization	69
2.7.2	Referencing the Wrong Data Variable	71
2.7.3	Out-of-Bounds References	75
2.7.4	Incorrect Subscripting	77
2.7.5	Incorrect Scaling or Data Units	79
2.7.6	Incorrect Data Packing or Unpacking	80
2.7.7	Inconsistent Data	82
2.8	Fault Tolerance	85
2.8.1	Management Strategy	85
2.8.2	Redundancy in Space	87
2.8.3	Redundancy in Time	89
2.8.4	Recoverability	90

3 Security

3.1	Vulnerable Code	102
-----	---------------------------	-----

3.2	The Buffer Overflow	106
3.3	Race Conditions	112
3.4	Problematic APIs	115
3.4.1	Functions Susceptible to Buffer Overflows	115
3.4.2	Format String Vulnerabilities	118
3.4.3	Path and Shell Metacharacter Vulnerabilities	119
3.4.4	Temporary Files	121
3.4.5	Functions Unsuitable for Cryptographic Use	122
3.4.6	Forgeable Data	124
3.5	Untrusted Input	125
3.6	Result Verification	131
3.7	Data and Privilege Leakage	134
3.7.1	Data Leakage	135
3.7.2	Privilege Leakage	138
3.7.3	The Java Approach	140
3.7.4	Isolating Privileged Code	141
3.8	Trojan Horse	143
3.9	Tools	146
4	Time Performance	151
4.1	Measurement Techniques	156
4.1.1	Workload Characterization	157
4.1.2	I/O-Bound Tasks	158
4.1.3	Kernel-Bound Tasks	161
4.1.4	CPU-Bound Tasks and Profiling Tools	163
4.2	Algorithm Complexity	173
4.3	Stand-Alone Code	179
4.4	Interacting with the Operating System	182
4.5	Interacting with Peripherals	190
4.6	Involuntary Interactions	191
4.7	Caching	194
4.7.1	A Simple System Call Cache	195
4.7.2	Replacement Strategies	197
4.7.3	Precomputing Results	199
5	Space Performance	207
5.1	Data	209
5.1.1	Basic Data Types	209

- 5.1.2 Aggregate Data Types 213
- 5.1.3 Alignment 215
- 5.1.4 Objects 222
- 5.2 Memory Organization 227
- 5.3 Memory Hierarchies 231
 - 5.3.1 Main Memory and Its Caches 232
 - 5.3.2 Disk Cache and Banked Memory 235
 - 5.3.3 Swap Area and File-Based Disk Storage 238
- 5.4 The Process/Operating System Interface 239
 - 5.4.1 Memory Allocation 239
 - 5.4.2 Memory Mapping 241
 - 5.4.3 Data Mapping 241
 - 5.4.4 Code Mapping 242
 - 5.4.5 Accessing Hardware Resources 244
 - 5.4.6 Interprocess Communication 244
- 5.5 Heap Memory Management 246
 - 5.5.1 Heap Fragmentation 247
 - 5.5.2 Heap Profiling 254
 - 5.5.3 Memory Leaks 256
 - 5.5.4 Garbage Collection 262
- 5.6 Stack Memory Management 264
 - 5.6.1 The Stack Frame 264
 - 5.6.2 Stack Space 269
- 5.7 Code 274
 - 5.7.1 Design Time 276
 - 5.7.2 Coding Time 279
 - 5.7.3 Build Time 280

6 Portability 289

- 6.1 Operating Systems 290
- 6.2 Hardware and Processor Architectures 296
 - 6.2.1 Data Type Properties 296
 - 6.2.2 Data Storage 298
 - 6.2.3 Machine-Specific Code 300
- 6.3 Compilers and Language Extensions 302
 - 6.3.1 Compiler Bugs 302
- 6.4 Graphical User Interfaces 307

6.5	Internationalization and Localization	309
6.5.1	Character Sets	309
6.5.2	Locale	313
6.5.3	Messages	316
7	Maintainability	325
7.1	Measuring Maintainability	326
7.1.1	The Maintainability Index	327
7.1.2	Metrics for Object-Oriented Programs	333
7.1.3	Dependency Metrics on Packages	343
7.2	Analyzability	351
7.2.1	Consistency	353
7.2.2	Expression Formatting	354
7.2.3	Statement Formatting	356
7.2.4	Naming Conventions	357
7.2.5	Statement-Level Comments	360
7.2.6	Versioning Comments	362
7.2.7	Visual Structure: Blocks and Indentation	363
7.2.8	Length of Expressions, Functions, and Methods	364
7.2.9	Control Structures	368
7.2.10	Boolean Expressions	372
7.2.11	Recognizability and Cohesion	374
7.2.12	Dependencies and Coupling	377
7.2.13	Code Block Comments	389
7.2.14	Data Declaration Comments	393
7.2.15	Appropriate Identifier Names	394
7.2.16	Locality of Dependencies	394
7.2.17	Ambiguity	396
7.2.18	Reviewability	397
7.3	Changeability	403
7.3.1	Identification	403
7.3.2	Separation	408
7.4	Stability	418
7.4.1	Encapsulation and Data Hiding	419
7.4.2	Data Abstraction	423
7.4.3	Type Checking	425
7.4.4	Compile-Time Assertions	428

7.4.5	Runtime Checks and Inspection-Time Assertions	431
7.5	Testability	432
7.5.1	Unit Testing	433
7.5.2	Integration Testing	437
7.5.3	System Testing	439
7.5.4	Test Coverage Analysis	441
7.5.5	Incidental Testing	444
7.6	Effects of the Development Environment	451
7.6.1	Incremental Builds	452
7.6.2	Tuning Build Performance	454
8	Floating-Point Arithmetic	465
8.1	Floating-Point Representation	466
8.1.1	Measuring Error	469
8.1.2	Rounding	470
8.1.3	Memory Format	472
8.1.4	Normalization and the Implied 1-Bit	474
8.1.5	Exponent Biasing	474
8.1.6	Negative Numbers	475
8.1.7	Denormal Numbers	475
8.1.8	Special Values	476
8.2	Rounding	478
8.3	Overflow	481
8.4	Underflow	483
8.5	Cancellation	487
8.6	Absorption	491
8.7	Invalid Operations	495
A	Source Code Credits	503
	Bibliography	505
	Index	523
	Author Index	563