Contents

1 **Introduction** 1

Chapter 1 Introduction 1
- 1.1 What is Software Engineering? . . . 5
- 1.2 Phases in the Development of Software . . . 10
- 1.3 Maintenance or Evolution . . . 16
- 1.4 From the Trenches . . . 17
  - 1.4.1 Ariane 5, Flight 501 . . . 18
  - 1.4.2 Therac-25 . . . 19
  - 1.4.3 The London Ambulance Service . . . 21
  - 1.4.4 Who Counts the Votes? . . . 23
- 1.5 Software Engineering Ethics . . . 25
- 1.6 Quo Vadis? . . . 27
- 1.7 Summary . . . 29
- 1.8 Further Reading . . . 29
- Exercises . . . 30

I **Software Management** 33

2 **Introduction to Software Engineering Management** 34

Chapter 2 Introduction to Software Engineering Management 34
- 2.1 Planning a Software Development Project . . . 37
- 2.2 Controlling a Software Development Project . . . 40
- 2.3 Summary . . . 42
- Exercises . . . 43

3 **The Software Life Cycle Revisited** 45

Chapter 3 The Software Life Cycle Revisited 45
- 3.1 The Waterfall Model . . . 48
- 3.2 Agile Methods . . . 50

3.2.1 Prototyping 51
3.2.2 Incremental Development 56
3.2.3 Rapid Application Development and DSDM 57
3.2.4 Extreme Programming 61
3.3 The Rational Unified Process (RUP) 64
3.4 Intermezzo: Maintenance or Evolution 66
3.5 Software Product Lines 70
3.6 Process Modeling 71
3.7 Summary 75
3.8 Further Reading 75
Exercises 76

4 Configuration Management **78**

Chapter 4 Configuration Management 78
4.1 Tasks and Responsibilities 80
4.2 Configuration Management Plan 85
4.3 Summary 86
4.4 Further Reading 88
Exercises 88

5 People Management and Team Organization **89**

Chapter 5 People Management and Team Organization 89
5.1 People Management 91
5.1.1 Coordination Mechanisms 93
5.1.2 Management Styles 94
5.2 Team Organization 96
5.2.1 Hierarchical Organization 96
5.2.2 Matrix Organization 98
5.2.3 Chief Programmer Team 99
5.2.4 SWAT Team 100
5.2.5 Agile Team 100
5.2.6 Open Source Software Development 101
5.2.7 General Principles for Organizing a Team 103
5.3 Summary 104
5.4 Further Reading 105
Exercises 105

6 On Managing Software Quality **107**

Chapter 6 On Managing Software Quality 107
6.1 On Measures and Numbers 110
6.2 A Taxonomy of Quality Attributes 116

6.3 Perspectives on Quality 123
6.4 The Quality System 127
6.5 Software Quality Assurance 128
6.6 The Capability Maturity Model (CMM) 130
6.7 Some Critical Notes 136
6.8 Getting Started 137
6.9 Summary 140
6.10 Further Reading 140
Exercises 141

7 Cost Estimation 144

Chapter 7 Cost Estimation 144
7.1 Algorithmic Models 148
7.1.1 Walston--Felix 151
7.1.2 COCOMO 153
7.1.3 Putnam 155
7.1.4 Function Point Analysis 156
7.1.5 COCOMO 2: Variations on a Theme 159
7.2 Guidelines for Estimating Cost 166
7.3 Distribution of Manpower over Time 169
7.4 Summary 171
7.5 Further Reading 174
Exercises 174

8 Project Planning and Control 176

Chapter 8 Project Planning and Control 176
8.1 A Systems View of Project Control 177
8.2 A Taxonomy of Software Development Projects 179
8.3 Risk Management 184
8.4 Techniques for Project Planning and Control 189
8.5 Summary 194
8.6 Further Reading 194
Exercises 195

II The Software Life Cycle 197

9 Requirements Engineering 199

Chapter 9 Requirements Engineering 199
9.1 Requirements Elicitation 205
9.1.1 Requirements Engineering Paradigms 210

9.1.2 Requirements Elicitation Techniques 212
9.1.3 Goals and Viewpoints 220
9.1.4 Prioritizing Requirements 223
9.1.5 COTS selection 224
9.2 Requirements Documentation and Management 227
9.2.1 Requirements Management 234
9.3 Requirements Specification Techniques 236
9.3.1 Specifying Non-Functional Requirements 238
9.4 Verification and Validation 239
9.5 Summary 240
9.6 Further Reading 242
Exercises 243

10 Modeling **246**

Chapter 10 Modeling 246
10.1 Classic Modeling Techniques 248
10.1.1 Entity--Relationship Modeling 248
10.1.2 Finite State Machines 250
10.1.3 Data Flow Diagrams (DFD) 252
10.1.4 CRC Cards 252
10.2 On Objects and Related Stuff 254
10.3 The Unified Modeling Language 260
10.3.1 The Class Diagram 260
10.3.2 The State Machine Diagram 265
10.3.3 The Sequence Diagram 268
10.3.4 The Communication Diagram 271
10.3.5 The Component Diagram 272
10.3.6 The Use Case 273
10.4 Summary 274
10.5 Further Reading 274
Exercises 274

11 Software Architecture **276**

Chapter 11 Software Architecture 276
11.1 Software Architecture and the Software Life Cycle 280
11.2 Architecture design 281
11.2.1 Architecture as a set of design decisions 284
11.3 Architectural views 285
11.4 Architectural Styles 291
11.5 Software Architecture Assessment 306
11.6 Summary 309
11.7 Further Reading 310

Exercises 311

12 Software Design 313

Chapter 12Software Design 313
12.1 Design Considerations 317
12.1.1 Abstraction 318
12.1.2 Modularity 321
12.1.3 Information Hiding 325
12.1.4 Complexity 325
12.1.5 System Structure 333
12.1.6 Object-Oriented Metrics 337
12.2 Classical Design Methods 340
12.2.1 Functional Decomposition 342
12.2.2 Data Flow Design (SA/SD) 346
12.2.3 Design based on Data Structures 351
12.3 Object-Oriented Analysis and Design Methods 359
12.3.1 The Booch Method 366
12.3.2 Fusion 367
12.3.3 RUP Revisited 369
12.4 How to Select a Design Method 370
12.4.1 Object Orientation: Hype or the Answer? 373
12.5 Design Patterns 375
12.6 Design Documentation 380
12.7 Verification and Validation 383
12.8 Summary 384
12.9 Further Reading 388
Exercises 389

13 Software Testing 394

Chapter 13Software Testing 394
13.1 Test Objectives 398
13.1.1 Test Adequacy Criteria 401
13.1.2 Fault Detection Versus Confidence Building 402
13.1.3 From Fault Detection to Fault Prevention 403
13.2 Testing and the Software Life Cycle 406
13.2.1 Requirements Engineering 407
13.2.2 Design 408
13.2.3 Implementation 409
13.2.4 Maintenance 409
13.2.5 Test-Driven Development (TDD) 410
13.3 Verification and Validation Planning and Documentation 411
13.4 Manual Test Techniques 413

13.4.1 Reading 414
13.4.2 Walkthroughs and Inspections 415
13.4.3 Correctness Proofs 417
13.4.4 Stepwise Abstraction 418
13.5 Coverage-Based Test Techniques 419
13.5.1 Control-Flow Coverage 420
13.5.2 Dataflow Coverage 423
13.5.3 Coverage-Based Testing of Requirements Specifications . . . 424
13.6 Fault-Based Test Techniques 425
13.6.1 Error Seeding 425
13.6.2 Mutation Testing 428
13.7 Error-Based Test Techniques 429
13.8 Comparison of Test Techniques 431
13.8.1 Comparison of Test Adequacy Criteria 432
13.8.2 Properties of Test Adequacy Criteria 434
13.8.3 Experimental Results 436
13.9 Different Test Stages 438
13.10Estimating Software Reliability 439
13.11Summary 447
13.12Further Reading 448
Exercises 449

14 Software Maintenance **453**

Chapter 14 Software Maintenance 453
14.1 Maintenance Categories Revisited 456
14.2 Major Causes of Maintenance Problems 459
14.3 Reverse Engineering and Refactoring 463
14.3.1 Refactoring 466
14.3.2 Inherent Limitations 469
14.3.3 Tools 473
14.4 Software Evolution Revisited 474
14.5 Organizational and Managerial Issues 476
14.5.1 Organization of Maintenance Activities 477
14.5.2 Software Maintenance from a Service Perspective 480
14.5.3 Control of Maintenance Tasks 486
14.5.4 Quality Issues 489
14.6 Summary 490
14.7 Further Reading 491
Exercises 492

15 Software Tools **494**

Chapter 15 Software Tools 494

15.1 Toolkits 499
15.2 Language-Centered Environments 500
15.3 Integrated Environments and Workbenches 501
15.3.1 Analyst WorkBenches 501
15.3.2 Programmer Workbenches 503
15.3.3 Management WorkBenches 507
15.3.4 Integrated Project Support Environments 508
15.4 Process-Centered Environments 508
15.5 Summary 510
15.6 Further Reading 511
Exercises 512

Bibliography **514**