Contents

Preface to the Second Edition xiii

The Role of Object Orientation xiv

The Purpose of Case Studies xiv

Instructor Resources xv

Preface to the First Edition xvii

Audience xviii

Prerequisites xviii

Organization and Content xviii

Exercises xix

Case Studies xix

Laboratory Course xix

Reading Graph x

Acknowledgments x

Chapter 1 Software Engineering: A Preview 1

1.1 The Role of Software Engineering in System Design 2

1.2 A Shortened History of Software Engineering 3

1.3 The Role of The Software Engineer 5

1.4 The Software Life Cycle 6

1.5 The Relationship of Software Engineering to Other Areas of Computer Science 8

1.5.1 Programming Languages 9

1.5.2 Operating Systems 10

1.5.3 Data Bases 10

1.5.4 Artificial Intelligence 11

1.5.5 Theoretical Models 12

1.6 The Relationship of Software Engineering to Other Disciplines 12

1.6.1 Management Science 13

1.6.2 Systems Engineering 13

1.7 Concluding Remarks 13

Bibliographic Notes 14

Ghezzi, Carlo
Fundamentals of software engineering

digitalisiert durch:
IDS Basel Bern

5.5.2 UML Diagrams for Specifying Behaviors 177
5.5.3 Finite State Machines: Describing Control Flow 179
5.5.4 Petri Nets: Specifying Asynchronous Systems 185
5.6 Descriptive Specifications 210
5.6.1 Entity-Relationship Diagrams 210
5.6.2 Logic Specifications 213
5.6.3 Algebraic Specifications 229
5.7 Building and Using Specifications in Practice 236
5.7.1 Requirements for Specification Notations 236
5.7.2 Building Modular Specifications 240
5.7.3 Specifications for the End User 257
Concluding Remarks 258
Further Exercises 259
Hints and Sketchy Solutions 262
Bibliographic Notes 266

Chapter 6 Verification 269

6.1 Goals and Requirements of Verification 270
6.1.1 Everything Must Be Verified 271
6.1.2 The Results of Verification May Not Be Binary 271
6.1.3 Verification May Be Objective or Subjective 272
6.1.4 Even Implicit Qualities Must Be Verified 273
6.2 Approaches to Verification 274
6.3 Testing 274
6.3.1 Goals for Testing 275
6.3.2 Theoretical Foundations of Testing 277
6.3.3 Empirical Testing Principles 280
6.3.4 Testing in the Small 282
6.3.5 Testing in the Large 302
6.3.6 Separate Concerns in the Testing Activity 312
6.3.7 Testing Concurrent and Real-Time Systems 313
6.4 Analysis 316
6.4.1 Informal Analysis Techniques 317
6.4.2 Correctness Proofs 320
6.5 Symbolic Execution 337
6.5.1 Basic Concepts of Symbolic Execution 339
6.5.2 Programs with Arrays 342
6.5.3 The Use of Symbolic Execution in Testing 345
6.6 Model Checking 347
6.7 Putting it All Together 349
6.8 Debugging 351
6.9 Verifying Other Software Properties 355
6.9.1 Verifying Performance 356

6.9.2 Verifying Reliability 356
6.9.3 Verifying Subjective Qualities 360
Concluding Remarks 371
Further Exercises 372
Hints and Sketchy Solutions 378
Bibliographic Notes 381

Chapter 7 The Software Production Process 385

7.1 What is a Software Process Model? 386
7.2 Why Are Software Process Models Important? 388
7.3 The Main Activities of Software production 391
7.3.1 Feasibility Study 391
7.3.2 Eliciting, Understanding, and Specifying Requirements 392
7.3.3 Definition of the Software Architecture and Detailed Design 399
7.3.4 Coding and Module Testing 399
7.3.5 Integration and System Testing 400
7.3.6 Delivery, Deployment, and Maintenance 400
7.3.7 Other Activities 401
7.4 An Overview of Software Process Models 403
7.4.1 Waterfall Models 403
7.4.2 Evolutionary Models 410
7.4.3 Transformation Model 413
7.4.4 Spiral Model 416
7.4.5 An Assessment of Process Models 417
7.5 Dealing with Legacy Software 420
7.6 Case Studies 421
7.6.1 Case Study: A Telephone Switching System 421
7.6.2 Case Study: A Budget Control System 426
7.6.3 Case study: The Microsoft Synchronize and Stabilize Process 430
7.6.4 Case study: The Open Source Approach 431
7.7 Organizing the Process 433
7.7.1 Structured Analysis/Structured Design 434
7.7.2 Jackson's System Development and Structured Programming 439
7.7.3 The Unified Software Development Process 444
7.8 Organizing Artifacts: Configuration Management 447
7.9 Software Standards 451
7.10 Concluding Remarks 451
Further Exercises 452
Hints and Sketchy Solutions 453
Bibliographic Notes 454

Chapter 8 Management of Software Engineering 457

8.1 Management Functions 459

8.2 Project Planning 460
8.2.1 Software Productivity 462
8.2.2 People and Productivity 469
8.2.3 Cost Estimation 469
8.3 Project Control 476
8.3.1 Work Breakdown Structures 477
8.3.2 Gantt Charts 478
8.3.3 PERT Charts 479
8.3.4 Dealing with Deviations from the Plan 482
8.4 Organization 483
8.4.1 Centralized-Control Team Organization 486
8.4.2 Decentralized-Control Team Organization 487
8.4.3 Mixed-Control Team Organization 488
8.4.4 An Assessment of Team Organizations 489
8.5 Risk Management 490
8.5.1 Typical Management Risks in Software Engineering 491
8.6 Capability Maturity Model 492
8.7 Concluding Remarks 495
Further Exercises 497
Hints and Sketchy Solutions 498
Bibliographic Notes 499

Chapter 9 Software Engineering Tools and Environments 501

9.1 Historical Evolution of Tools and Environments 502
9.2 Dimensions for Comparing SoftwareTools 503
9.3 Representative Tools 507
9.3.1 Editors 507
9.3.2 Linkers 508
9.3.3 Interpreters 508
9.3.4 Code Generators 509
9.3.5 Debuggers 510
9.3.6 Tools Used in Software Testing 511
9.3.7 Static Analyzers 512
9.3.8 Graphical User Interface Tools 514
9.3.9 Configuration Management Tools 516
9.3.10 Tracking tools 520
9.3.11 Reverse and reengineering tools 521
9.3.12 Process support tools 522
9.3.13 Management Tools 522
9.4 Tool Integration 523
9.5 Forces Influencing the Evolution of Tools 524
9.6 Concluding Remarks 525
Further Exercises 525
Hints and Sketchy Solutions 526
Bibliographic Notes 526

Chapter 10 Epilogue 529

10.1 The Future 529
10.2 Ethics and Social Responsibility 532
10.3 Software Engineering Code of Ethics 533
10.4 Concluding Remarks 534
Bibliographic notes 534

Appendix 536

Case Studies

Case Study A: Automating a Law Office 536
A.1 Economic and Financial Planning 538
A.2 Technical Planning and Management 538
A.3 Project Monitoring 540
A.4 Initial Delivery 540
A.5 Partial Recovery 541
Case Study B: Building a Family of Compliers 541
B.1 Initial Product Planning 541
B.2 Economic and Financial Planning 542
B.3 Technical Planning and Management 542
B.4 Early Development Phases 543
B.5 Project Monitoring 543
B.6 Project Reexamination, Revival, And Goal Setting 544
B.7 Assignment of Responsibility 545
B.8 Steady Progress and Release of the Product 546
B.9 Product Distribution 546
B.10 Remarks 546
Case Study C: Incremental Delivery 549
Case Study D. Applying Formal Methods in Industry 550
D.1 Education and Training 551
D.2 Requirements Specification 552
D.3 Requirements Validation and Verification Planning 553
D.4 Design, Implementation, and Verification 555
D.5 Overall Evaluation and Assessment 555
D.6 The Impact of the Project on Company's Strategy 557
Concluding Remarks 558
Bibliographic Notes 559
Bibliography 560
Index 589