Contents

Preface v

Part 1 Overview 1

Chapter 1 Introduction 3

1.1 FAQs about software engineering 5

1.2 Professional and ethical responsibility 14

Key Points 17
Further Reading 18
Exercises 18

Chapter 2 Socio-technical systems 20

2.1 Emergent system properties 23

2.2 Systems engineering 25

2.3 Organisations, people and computer systems 34

2.4 Legacy systems 38

Key Points 40
Further Reading 41
Exercises 41

Chapter 3 Critical systems 43

3.1 A simple safety-critical system 46

3.2 System dependability 47

3.3 Availability and reliability 51

3.4 Safety 55

3.5 Security 58

Key Points 60
Further Reading 61
Exercises 61

Chapter 4 Software processes 63

4.1 Software process models 65

4.2 Process iteration 71

4.3 Process activities 74

4.4 The Rational Unified Process 82

4.5 Computer-Aided Software Engineering 85

Key Points 89
Further Reading 90
Exercises 91

Chapter 5 Project management 92

5.1 Management activities 94

5.2 Project planning 96

5.3 Project scheduling 99

5.4 Risk management 104

Key Points 111
Further Reading 112
Exercises 112

Part 2 Requirements 115

Chapter 6 Software requirements 117

6.1 Functional and non-functional requirements 119

6.2 User requirements 127

6.3 System requirements 129

6.4 Interface specification 135

6.5 The software requirements document 136

Key Points 140
Further Reading 140
Exercises 141

Chapter 7 Requirements engineering processes 142

7.1 Feasibility studies 144

7.2 Requirements elicitation and analysis 146

7.3 Requirements validation 158

7.4 Requirements management 161

Key Points 166
Further Reading 167
Exercises 167

Chapter 8 System models 169

8.1 Context models 171

8.2 Behavioural models 173

8.3 Data models 177

8.4 Object models 181

8.5 Structured methods 187

Key Points 190
Further Reading 191
Exercises 191

Chapter 9 Critical systems specification 193

9.1 Risk-driven specification 195

9.2 Safety specification 202

9.3 Security specification 204

9.4 Software reliability specification 207

Key Points 213
Further Reading 214
Exercises 214

Chapter 10 Formal specification 217

10.1 Formal specification in the software process 219

10.2 Sub-system interface specification 222

10.3 Behavioural specification 229

Key Points 236
Further Reading 236
Exercises 237

Part 3 Design 239

Chapter 11 Architectural Design 241

11.1 Architectural design decisions 245

11.2 System organisation 247

11.3 Modular decomposition styles 252

11.4 Control styles 256

11.5 Reference architectures 260

Key Points 263
Further Reading 264
Exercises 264

Chapter 12 Distributed Systems Architectures 266

12.1 Multiprocessor architectures 269

12.2 Client–server architectures 270

12.3 Distributed object architectures 275

12.4 Inter-organisational distributed computing 282

Key Points 289
Further Reading 290
Exercises 290

Chapter 13 Application Architectures 292

13.1 Data processing systems 295

13.2 Transaction processing systems 298

13.3 Event processing systems 304

13.4 Language processing systems 307

Key Points 310
Further Reading 310
Exercises 311

Chapter 14 Object-oriented Design 313

14.1 Objects and object classes 316

14.2 An object-oriented design process 320

14.3 Design evolution 335

Key Points 336
Further Reading 337
Exercises 337

Chapter 15 Real-Time Software Design 339

15.1 System design 342

15.2 Real-time operating systems 346

15.3 Monitoring and control systems 349

15.4 Data acquisition systems 355

Key Points 357
Further Reading 359
Exercises 359

Chapter 16 User Interface Design 362

16.1 Design issues 366

16.2 The UI design process 376

16.3 User analysis 378

16.4 User interface prototyping 381

16.5 Interface evaluation 383

Key Points 385
Further Reading 386
Exercises 386

Part 4 Critical Systems 389

Chapter 17 Rapid Software Development 391

17.1 Agile methods 396

17.2 Extreme programming 398

17.3 Rapid application development 405

17.4 Software prototyping 409

Key Points 412
Further Reading 413
Exercises 414

Chapter 18 Software Reuse 415

18.1 The reuse landscape 418

18.2 Design patterns 421

18.3 Generator-based reuse 423

18.4 Application frameworks 426

18.5 Application system reuse 428

Key Points 437
Further Reading 437
Exercises 438

Chapter 19 Component-based Software Engineering 439

19.1 Components and component models 442

19.2 The CBSE process 450

19.3 Component composition 452

Key Points 460
Further Reading 460
Exercises 461

Chapter 20 Critical Systems Development 462

20.1 Dependable processes 466

20.2 Dependable programming 467

20.3 Fault tolerance 475

20.4 Fault-tolerant architectures 482

Key Points 486
Further Reading 486
Exercises 487

Chapter 21 Software Evolution 488

21.1 Program evolution dynamics 490

21.2 Software maintenance 492

21.3 Evolution processes 498

21.4 Legacy system evolution 504

Key Points 509
Further Reading 510
Exercises 510

Part 5 Verification and Validation 513

Chapter 22 Verification and Validation 515

22.1 Planning verification and validation 519

22.2 Software inspections 521

22.3 Automated static analysis 527

22.4 Verification and formal methods 530

Key Points 535
Further Reading 535
Exercises 536

Chapter 23 Software Testing 537

23.1 System testing 540

23.2 Component testing 547

23.3 Test case design 551

23.4 Test automation 561

Key Points 563
Further Reading 564
Exercises 565

Chapter 24 Critical Systems Validation 566

24.1 Reliability validation 568

24.2 Safety assurance 574

24.3 Security assessment 581

24.4 Safety and dependability cases 583

Key Points 586
Further Reading 587
Exercises 587

Part 6 Managing People 589

Chapter 25 Managing People 591

25.1 Selecting staff 593

25.2 Motivating people 596

25.3 Managing Groups 599

25.4 The People Capability Maturity Model 607

Key Points 610
Further Reading 610
Exercises 611

Chapter 26 Software Cost Estimation 612

26.1 Software productivity 614

26.2 Estimation techniques 620

26.3 Algorithmic cost modelling 623

26.4 Project duration and staffing 637

Key Points 638
Further Reading 639
Exercises 639

Chapter 27 Quality Management 641

27.1 Process and product quality 644

27.2 Quality assurance and standards 645

27.3 Quality planning 652

27.4 Quality control 653

27.5 Software measurement and metrics 655

Key Points 663
Further Reading 663
Exercises 664

Chapter 28 Process Improvement 665

28.1 Process and product quality 667

28.2 Process classification 669

28.3 Process measurement 672

28.4 Process analysis and modelling 673

28.5 Process change 678

28.6 The CMMI process improvement framework 680

Key Points 687

Further Reading 687
Exercising 688

Chapter 29 Configuration Management 689

29.1 Configuration management planning 692

29.2 Change management 695

29.3 Version and release management 698

29.4 System building 705

29.5 CASE tools for configuration management 706

Key Points 711
Furhter Reading 711
Exercises 712

Glossaary 715
References 727
Index 743

Trademark Notice
The following are trademarks or registered trademarks of their respective companies:

Java, JavaBeans and Modula-2 are trademarks of Sun Microsystems, Inc.; Lotus Notes is a trademark of Lotus Development Corporation; Mac OS is a trademark of Apple Computer, Inc.; Microsoft, PowerPoint, Windows, Visual Basic and Visual C++ are trademarks of Microsoft Corporation; Unix is a trademark licensed through X/Open Company Ltd.