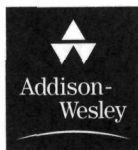


Software Engineering

Sixth Edition

Ian Sommerville



An imprint of **Pearson Education**

Harlow, England · London · New York · Reading, Massachusetts · San Francisco · Toronto · Don Mills, Ontario · Sydney
Tokyo · Singapore · Hong Kong · Seoul · Taipei · Cape Town · Madrid · Mexico City · Amsterdam · Munich · Paris · Milan



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	Computer-based system engineering	20
	2.1 Emergent system properties	22
	2.2 Systems and their environment	24
	2.3 System modelling	26
	2.4 The system engineering process	29
	2.5 System procurement	37

Key points	39
Further reading	40
Exercises	40
Chapter 3 Software processes	42
3.1 Software process models	44
3.2 Process iteration	51
3.3 Software specification	55
3.4 Software design and implementation	56
3.5 Software validation	60
3.6 Software evolution	63
3.7 Automated process support	63
Key points	68
Further reading	68
Exercises	69
Chapter 4 Project management	71
4.1 Management activities	73
4.2 Project planning	75
4.3 Project scheduling	78
4.4 Risk management	84
Key points	90
Further reading	91
Exercises	92

Part 2 Requirements 95

Chapter 5 Software requirements	97
5.1 Functional and non-functional requirements	100
5.2 User requirements	106
5.3 System requirements	109

5.4	The software requirements document	115
	Key points	119
	Further reading	119
	Exercises	120
Chapter 6	Requirements engineering processes	121
6.1	Feasibility studies	123
6.2	Requirements elicitation and analysis	124
6.3	Requirements validation	137
6.4	Requirements management	139
	Key points	145
	Further reading	145
	Exercises	146
Chapter 7	System models	148
7.1	Context models	150
7.2	Behavioural models	153
7.3	Data models	158
7.4	Object models	160
7.5	CASE workbenches	166
	Key points	168
	Further reading	169
	Exercises	169
Chapter 8	Software prototyping	171
8.1	Prototyping in the software process	174
8.2	Rapid prototyping techniques	180
8.3	User interface prototyping	188
	Key points	189
	Further reading	190
	Exercises	190

Chapter 9	Formal specification	192
	9.1 Formal specification in the software process	194
	9.2 Interface specification	197
	9.3 Behavioural specification	204
	Key points	209
	Further reading	210
	Exercises	210
Part 3 Design		213
Chapter 10	Architectural design	215
	10.1 System structuring	219
	10.2 Control models	224
	10.3 Modular decomposition	229
	10.4 Domain-specific architectures	233
	Key points	236
	Further reading	237
	Exercises	237
Chapter 11	Distributed systems architectures	239
	11.1 Multiprocessor architectures	243
	11.2 Client-server architectures	244
	11.3 Distributed object architectures	249
	11.4 CORBA	252
	Key points	257
	Further reading	258
	Exercises	258
Chapter 12	Object-oriented design	260
	12.1 Objects and object classes	262
	12.2 An object-oriented design process	267

12.3	Design evolution	280
	Key points	282
	Further reading	282
	Exercises	283
Chapter 13	Real-time software design	285
13.1	System design	287
13.2	Real-time executives	291
13.3	Monitoring and control systems	295
13.4	Data acquisition systems	300
	Key points	303
	Further reading	303
	Exercises	304
Chapter 14	Design with reuse	306
14.1	Component-based development	310
14.2	Application families	318
14.3	Design patterns	322
	Key points	325
	Further reading	325
	Exercises	326
Chapter 15	User interface design	327
15.1	User interface design principles	330
15.2	User interaction	332
15.3	Information presentation	334
15.4	User support	340
15.5	Interface evaluation	345
	Key points	347
	Further reading	348
	Exercises	348

Part 4 Critical Systems 351

Chapter 16 Dependability	353
16.1 Critical systems	356
16.2 Availability and reliability	359
16.3 Safety	364
16.4 Security	367
Key points	369
Further reading	369
Exercises	370
Chapter 17 Critical systems specification	371
17.1 Software reliability specification	373
17.2 Safety specification	379
17.3 Security specification	387
Key points	389
Further reading	389
Exercises	390
Chapter 18 Critical systems development	392
18.1 Fault minimisation	393
18.2 Fault tolerance	400
18.3 Fault-tolerant architectures	410
18.4 Safe system design	413
Key points	414
Further reading	415
Exercises	415

Part 5 Verification and Validation 417

Chapter 19 Verification and validation	419
19.1 Verification and validation planning	423
19.2 Software inspections	425

19.3	Automated static analysis	431
19.4	Cleanroom software development	434
	Key points	437
	Further reading	438
	Exercises	438
Chapter 20	Software testing	440
20.1	Defect testing	442
20.2	Integration testing	452
20.3	Object-oriented testing	458
20.4	Testing workbenches	462
	Key points	464
	Further reading	465
	Exercises	466
Chapter 21	Critical systems validation	467
21.1	Formal methods and critical systems	469
21.2	Reliability validation	470
21.3	Safety assurance	476
21.4	Security assessment	483
	Key points	484
	Further reading	484
	Exercises	485
Part 6 Management		487
Chapter 22	Managing people	489
22.1	Limits to thinking	490
22.2	Group working	497
22.3	Choosing and keeping people	503

	22.4 The People Capability Maturity Model	506
	Key points	508
	Further reading	509
	Exercises	509
Chapter 23	Software cost estimation	511
	23.1 Productivity	513
	23.2 Estimation techniques	518
	23.3 Algorithmic cost modelling	520
	23.4 Project duration and staffing	531
	Key points	533
	Further reading	533
	Exercises	534
Chapter 24	Quality management	535
	24.1 Quality assurance and standards	539
	24.2 Quality planning	544
	24.3 Quality control	546
	24.4 Software measurement and metrics	547
	Key points	555
	Further reading	555
	Exercises	556
Chapter 25	Process improvement	557
	25.1 Process and product quality	560
	25.2 Process analysis and modelling	562
	25.3 Process measurement	566
	25.4 The SEI Process Capability Maturity Model	568
	25.5 Process classification	573
	Key points	576
	Further reading	576
	Exercises	577

Part 7 Evolution 579

Chapter 26	Legacy systems	581
26.1	Legacy system structures	583
26.2	Legacy system design	587
26.3	Legacy system assessment	592
	Key points	598
	Further reading	599
	Exercises	599
Chapter 27	Software change	601
27.1	Program evolution dynamics	603
27.2	Software maintenance	605
27.3	Architectural evolution	614
	Key points	620
	Further reading	620
	Exercises	621
Chapter 28	Software re-engineering	622
28.1	Source code translation	626
28.2	Reverse engineering	628
28.3	Program structure improvement	629
28.4	Program modularisation	632
28.5	Data re-engineering	634
	Key points	638
	Further reading	639
	Exercises	639
Chapter 29	Configuration management	641
29.1	Configuration management planning	644
29.2	Change management	647

29.3	Version and release management	650
29.4	System building	655
29.5	CASE tools for configuration management	656
	Key points	660
	Further reading	661
	Exercises	661
	References	663
	Index	679

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.