

Inhaltsverzeichnis

Vorwort	17
Webintegration	18
Zielgruppe	19
Änderungen zu vorigen Ausgaben	19
Das Buch als Vorlesungsskript	20
Zusatzmaterialien	20
Danksagungen	21
Vorwort zur deutschen Ausgabe	22
Webinhalte zum vorliegenden Buch	22
Anmerkungen zur deutschen Übersetzung	23
Teil I Einführung in Software-Engineering	25
Kapitel 1 Einführung	27
1.1 Professionelle Softwareentwicklung	30
1.1.1 Software-Engineering	33
1.1.2 Vielfalt des Software-Engineerings	36
1.1.3 Software-Engineering und das Web	38
1.2 Ethik des Software-Engineerings	40
1.3 Fallstudien	43
1.3.1 Ein Steuerungssystem für Insulinpumpen	44
1.3.2 Ein Patienteninformationssystem für die psychiatrische Ambulanz	46
1.3.3 Eine Wetterstation in Wildnisgebieten	48
Zusammenfassung	50
Ergänzende Literatur	50
Übungen	51
Kapitel 2 Softwareprozesse	53
2.1 Vorgehensmodelle	56
2.1.1 Das Wasserfallmodell	56
2.1.2 Inkrementelle Entwicklung	59
2.1.3 Wiederverwendungsorientiertes Software-Engineering	61
2.2 Prozessaktivitäten	63
2.2.1 Softwarespezifikation	64
2.2.2 Softwareentwurf und -implementierung	65
2.2.3 Softwarevalidierung	68
2.2.4 Weiterentwicklung von Software	70

2.3	Umgang mit Änderungen	71
2.3.1	Softwareprototypen	72
2.3.2	Inkrementelle Auslieferung	74
2.3.3	Das Spiralmodell nach Boehm	76
2.4	Der Rational Unified Process	78
	Zusammenfassung	81
	Ergänzende Literatur	82
	Übungen	82

Kapitel 3 Agile Softwareentwicklung 85

3.1	Agile Methoden	88
3.2	Plangesteuerte und agile Entwicklung	92
3.3	Extreme Programming	94
3.3.1	Testen in XP	99
3.3.2	Pair Programming	102
3.4	Agiles Projektmanagement	103
3.5	Skalieren von agilen Methoden	106
	Zusammenfassung	109
	Ergänzende Literatur	109
	Übungen	110

Kapitel 4 Requirements-Engineering 113

4.1	Funktionale und nichtfunktionale Anforderungen	116
4.1.1	Funktionale Anforderungen	117
4.1.2	Nichtfunktionale Anforderungen	119
4.2	Die Gesamtsystemspezifikation	123
4.3	Anforderungsspezifikation	126
4.3.1	Spezifikation in natürlicher Sprache	128
4.3.2	Strukturierte Spezifikationen	129
4.4	Prozesse des Requirements-Engineerings	132
4.5	Anforderungserhebung und -analyse	133
4.5.1	Sammeln der Anforderungen	136
4.5.2	Ethnografie	142
4.6	Validierung von Anforderungen	144
4.7	Anforderungsmanagement	145
4.7.1	Planung des Anforderungsmanagements	147
4.7.2	Anforderungsänderungsmanagement	148
	Zusammenfassung	150
	Ergänzende Literatur	150
	Übungen	151

Kapitel 5 Systemmodellierung 153

5.1	Kontextmodelle	157
5.2	Interaktionsmodelle	160
5.2.1	Anwendungsfallmodellierung	160
5.2.2	Sequenzdiagramme	162

5.3	Strukturelle Modelle	165
5.3.1	Klassendiagramme	165
5.3.2	Generalisierung	168
5.3.3	Aggregation	169
5.4	Verhaltensmodelle	170
5.4.1	Datenorientierte Modellierung	170
5.4.2	Ereignisgesteuerte Modellierung	172
5.5	Modellgetriebene Softwareentwicklung	175
5.5.1	Modellgetriebene Architektur	176
5.5.2	Ausführbares UML	179
	Zusammenfassung	180
	Ergänzende Literatur	180
	Übungen	181

Kapitel 6 Entwurf der Architektur 183

6.1	Architektonische Entwurfsentscheidungen	187
6.2	Architektursichten	190
6.3	Architekturmuster	192
6.3.1	Schichtenarchitektur	194
6.3.2	Repository-Architektur	196
6.3.3	Client-Server-Architektur	198
6.3.4	Pipes-and-Filter-Architektur	200
6.4	Anwendungsarchitekturen	202
6.4.1	Transaktionsverarbeitende Systeme	204
6.4.2	Informationssysteme	205
6.4.3	Sprachverarbeitende Systeme	207
	Zusammenfassung	211
	Ergänzende Literatur	211
	Übungen	212

Kapitel 7 Entwurf und Implementierung 213

7.1	Objektorientierter Entwurf mit UML	216
7.1.1	Systemkontext und Interaktionen	217
7.1.2	Entwurf der Architektur	220
7.1.3	Bestimmung der Objektklassen	221
7.1.4	Entwurfsmodelle	223
7.1.5	Schnittstellenspezifikation	227
7.2	Entwurfsmuster	228
7.3	Implementierungsaspekte	232
7.3.1	Wiederverwendung	233
7.3.2	Konfigurationsverwaltung	234
7.3.3	Host-Ziel-Entwicklung	236
7.4	Open-Source-Entwicklung	238
7.4.1	Open-Source-Lizenzierung	239
	Zusammenfassung	242
	Ergänzende Literatur	242
	Übungen	243

Kapitel 8	Testen von Software	245
8.1	Entwicklertests	251
8.1.1	Modultests	252
8.1.2	Auswahl der Testfälle für Modultests	254
8.1.3	Testen von Komponenten	257
8.1.4	Testen von Systemen	260
8.2	Testgetriebene Entwicklung	263
8.3	Freigabetests	266
8.3.1	Anforderungsbasiertes Testen	266
8.3.2	Szenariobasiertes Testen	267
8.3.3	Leistungstests	269
8.4	Benutzertests	270
	Zusammenfassung	273
	Ergänzende Literatur	273
	Übungen	274

Kapitel 9	Softwareevolution	275
9.1	Evolutionsprozesse	279
9.2	Dynamik der Programmevolution	282
9.3	Softwarewartung	285
9.3.1	Vorhersagen des Wartungsaufwands	289
9.3.2	Software-Reengineering	291
9.3.3	Präventive Wartung durch Refactoring	294
9.4	Verwaltung von Altsystemen	295
	Zusammenfassung	301
	Ergänzende Literatur	301
	Übungen	302

Teil II	Verlässlichkeit und Informationssicherheit	303
----------------	---	-----

Kapitel 10	Soziotechnische Systeme	305
10.1	Komplexe Systeme	309
10.1.1	Typische Systemeigenschaften	312
10.1.2	Nichtdeterminismus	314
10.1.3	Erfolgskriterien	315
10.2	Systems-Engineering	316
10.3	Systembeschaffung	318
10.4	Systementwicklung	321
10.5	Systembetrieb	325
10.5.1	Menschliches Versagen	326
10.5.2	Systemevolution	328
	Zusammenfassung	330
	Ergänzende Literatur	330
	Übungen	331

Kapitel 11	Verlässlichkeit und Informationssicherheit	333
11.1	Eigenschaften der Verlässlichkeit	336
11.2	Verfügbarkeit und Zuverlässigkeit	340
11.3	Betriebssicherheit	345
11.4	Informationssicherheit	349
	Zusammenfassung.	353
	Ergänzende Literatur.	353
	Übungen	354
Kapitel 12	Spezifikation der Verlässlichkeit und Informationssicherheit	355
12.1	Risikogesteuerte Spezifikation der Anforderungen.	357
12.2	Spezifikation der Betriebssicherheit	359
	12.2.1 Gefahrenerkennung.	360
	12.2.2 Gefahrenbewertung.	361
	12.2.3 Gefahrenanalyse	364
	12.2.4 Risikoreduzierung	366
12.3	Spezifikation der Zuverlässigkeit.	367
	12.3.1 Zuverlässigkeitsmetriken	369
	12.3.2 Nichtfunktionale Zuverlässigkeitsanforderungen	371
	12.3.3 Funktionale Zuverlässigkeitsanforderungen	375
12.4	Spezifikation der Informationssicherheit	376
12.5	Formale Spezifikation	380
	Zusammenfassung.	384
	Ergänzende Literatur.	384
	Übungen	385
Kapitel 13	Entwicklung verlässlicher Systeme	387
13.1	Redundanz und Diversität	390
13.2	Verlässliche Prozesse	392
13.3	Verlässliche Systemarchitekturen	395
	13.3.1 Schutzsysteme.	396
	13.3.2 Selbstüberwachende Architekturen	397
	13.3.3 Diversitäre Programmierung	399
	13.3.4 Softwarediversität	400
13.4	Verlässliche Programmierung.	402
	Zusammenfassung.	412
	Ergänzende Literatur.	412
	Übungen	413

Kapitel 14	Security-Engineering	415
14.1	Management von Sicherheitsrisiken.	419
14.1.1	Bewertung der Risiken während der Entwicklung.	421
14.1.2	Bewertung der Risiken während des Betriebs	424
14.2	Entwerfen unter dem Sicherheitsaspekt.	425
14.2.1	Architekturentwurf	426
14.2.2	Entwurfsrichtlinien	430
14.2.3	Entwerfen unter dem Aspekt der Inbetriebnahme	435
14.3	Überlebensfähigkeit des Systems	437
	Zusammenfassung	441
	Ergänzende Literatur	441
	Übungen	442
Kapitel 15	Gewährleistung der Verlässlichkeit und Informationssicherheit	443
15.1	Statische Analyse.	445
15.1.1	Verifikation und formale Methoden	446
15.1.2	Modellprüfung	448
15.1.3	Automatisierte statische Analyse	449
15.2	Zuverlässigkeit testen	452
15.2.1	Betriebsprofile	454
15.3	Testen der Informationssicherheit	455
15.4	Prozesssicherung	458
15.4.1	Prozesse für die Gewährleistung der Betriebssicherheit	459
15.5	Nachweis der Betriebssicherheit und Verlässlichkeit	462
15.5.1	Strukturierte Argumentationen	464
15.5.2	Strukturierte Betriebssicherheitsargumentationen	466
	Zusammenfassung	469
	Ergänzende Literatur	470
	Übungen	471
Teil III	Software-Engineering für Fortgeschrittene	473
Kapitel 16	Wiederverwendung von Software	475
16.1	Die Wiederverwendungslandschaft	480
16.2	Anwendungsframeworks.	483
16.3	Softwareproduktlinien	486
16.4	Wiederverwendung von COTS-Produkten	492
16.4.1	COTS-Lösungen	493
16.4.2	COTS-Integration	496
	Zusammenfassung	500
	Ergänzende Literatur	501
	Übungen	502

Kapitel 17	Komponentenbasiertes Software-Engineering	503
17.1	Komponenten und Komponentenmodelle	507
17.1.1	Komponentenmodelle	510
17.2	Prozesse des komponentenbasierten Software-Engineerings	513
17.2.1	Komponentenbasiertes Software-Engineering für Wiederverwendung	515
17.2.2	Komponentenbasiertes Software-Engineering mit Wiederverwendung.	517
17.3	Komposition von Komponenten	521
	Zusammenfassung.	528
	Ergänzende Literatur.	529
	Übungen	530
Kapitel 18	Entwicklung verteilter Systeme	531
18.1	Probleme der verteilten Systeme	534
18.1.1	Kommunikationsmodelle	537
18.1.2	Middleware	540
18.2	Client-Server-Systeme.	541
18.3	Architekturmuster für verteilte Systeme	543
18.3.1	Master-Slave-Architekturen	544
18.3.2	Zweischichtige Client-Server-Architekturen	545
18.3.3	Mehrschichtige Client-Server-Architekturen	547
18.3.4	Verteilte Komponentenarchitekturen	548
18.3.5	Peer-to-Peer-Architekturen	551
18.4	Software als Service	554
	Zusammenfassung.	559
	Ergänzende Literatur.	559
	Übungen	560
Kapitel 19	Serviceorientierte Architektur	561
19.1	Services als wiederverwendbare Komponenten	568
19.2	Service-Engineering	572
19.2.1	Ermittlung von Servicekandidaten	573
19.2.2	Entwerfen von Serviceschnittstellen	575
19.2.3	Implementierung und Bereitstellung der Services	578
19.2.4	Services für Altsysteme.	580
19.3	Softwareentwicklung mit Services.	581
19.3.1	Entwurf und Implementierung des Workflows	584
19.3.2	Testen von Services.	587
	Zusammenfassung.	589
	Ergänzende Literatur.	589
	Übungen	590

Kapitel 20	Eingebettete Software	591
20.1	Entwurf eingebetteter Systeme	594
20.1.1	Modellierung von Echtzeitsystemen	599
20.1.2	Programmierung von Echtzeitsystemen	601
20.2	Architekturmuster	602
20.2.1	Beobachten und Reagieren	603
20.2.2	Umgebungssteuerung	605
20.2.3	Pipelineverarbeitung	607
20.3	Analyse des Zeitverhaltens	608
20.4	Echtzeitbetriebssysteme	612
20.4.1	Prozessverwaltung	614
	Zusammenfassung	616
	Ergänzende Literatur	616
	Übungen	617
Kapitel 21	Aspektorientiertes Software-Engineering	619
21.1	Die Trennung von Belangen	622
21.2	Aspekte, Verbindungspunkte und Pointcuts	626
21.3	Software-Engineering mit Aspekten	630
21.3.1	Belangorientiertes Requirements-Engineering	632
21.3.2	Aspektorientiertes Entwerfen und Programmieren	636
21.3.3	Verifikation und Validierung	640
	Zusammenfassung	643
	Ergänzende Literatur	643
	Übungen	644
Teil IV	Softwaremanagement	645
Kapitel 22	Projektmanagement	647
22.1	Risikomanagement	650
22.1.1	Risikoerkennung	652
22.1.2	Risikoanalyse	653
22.1.3	Risikoplanung	656
22.1.4	Risikoüberwachung	656
22.2	Personalmanagement	657
22.2.1	Mitarbeitermotivation	658
22.3	Teamwork	662
22.3.1	Teammitglieder auswählen	664
22.3.2	Organisation der Gruppe	666
22.3.3	Kommunikation in der Gruppe	668
	Zusammenfassung	670
	Ergänzende Literatur	671
	Übungen	672

Kapitel 23	Projektplanung	673
23.1	Preiskalkulation für Software	677
23.2	Plangesteuerte Entwicklung	678
23.2.1	Projektpläne	679
23.2.2	Der Planungsprozess	680
23.3	Zeitplanung	682
23.3.1	Darstellung des Zeitplans	684
23.4	Agile Planung	688
23.5	Schätztechniken	690
23.5.1	Algorithmische Kostenmodellierung	692
23.5.2	Das COCOMO-II-Modell	693
23.5.3	Projektdauer und Personalplanung	702
	Zusammenfassung	704
	Ergänzende Literatur	704
	Übungen	705
Kapitel 24	Qualitätsmanagement	707
24.1	Softwarequalität	711
24.2	Softwarestandards	714
24.2.1	Der Rahmenstandard ISO 9001	717
24.3	Reviews und Inspektionen	719
24.3.1	Der Review-Prozess	720
24.3.2	Programminspektionen	722
24.4	Softwaremessung und -metriken	724
24.4.1	Produktmetriken	728
24.4.2	Softwarekomponentenanalyse	731
24.4.3	Mehrdeutigkeit von Messungen	732
	Zusammenfassung	735
	Ergänzende Literatur	736
	Übungen	737
Kapitel 25	Konfigurationsmanagement	739
25.1	Änderungsmanagement	743
25.2	Versionsmanagement	749
25.3	Systemerstellung	753
25.4	Release-Management	758
	Zusammenfassung	761
	Ergänzende Literatur	761
	Übungen	762
Kapitel 26	Prozessverbesserung	763
26.1	Der Prozessverbesserungsprozess	767
26.2	Prozessmessung	770
26.3	Prozessanalyse	773
26.3.1	Prozessausnahmen	776
26.4	Prozessänderung	777

26.5	Das CMMI-Rahmenwerk zur Prozessverbesserung	780
26.5.1	Das gestufte CMMI-Modell	783
26.5.2	Das kontinuierliche CMMI-Modell	785
	Zusammenfassung	787
	Ergänzende Literatur	787
	Übungen	788
	Glossar	789
	Literatur A-Z	811
	Register	831