

Inhaltsübersicht

Vorwort		15
Teil I	Einführung in Software-Engineering	21
Kapitel 1	Einführung	23
Kapitel 2	Softwareprozesse	53
Kapitel 3	Agile Softwareentwicklung	85
Kapitel 4	Requirements-Engineering	119
Kapitel 5	Systemmodellierung	161
Kapitel 6	Entwurf der Architektur	191
Kapitel 7	Entwurf und Implementierung	223
Kapitel 8	Testen von Software	255
Kapitel 9	Softwareevolution	287
Teil II	Systemverlässlichkeit und Informationssicherheit	319
Kapitel 10	Verlässliche Systeme	321
Kapitel 11	Reliability-Engineering	343
Kapitel 12	Safety-Engineering	379
Kapitel 13	Security-Engineering	417
Kapitel 14	Resilienz-Engineering	457

Teil III	Software-Engineering für Fortgeschrittene	489
Kapitel 15	Wiederverwendung von Software	491
Kapitel 16	Komponentenbasiertes Software-Engineering	521
Kapitel 17	Entwicklung verteilter Systeme	551
Kapitel 18	Serviceorientiertes Software-Engineering	585
Kapitel 19	Systems-Engineering	619
Kapitel 20	Systeme von Systemen	651
Kapitel 21	Entwicklung von Echtzeitsoftware	683
Teil IV	Softwaremanagement	713
Kapitel 22	Projektmanagement	715
Kapitel 23	Projektplanung	743
Kapitel 24	Qualitätsmanagement	779
Kapitel 25	Konfigurationsmanagement	813
Glossar		843
Literatur A-Z		867
Register		885

Inhaltsverzeichnis

Vorwort	15
Teil I Einführung in Software-Engineering	21
Kapitel 1 Einführung	23
1.1 Professionelle Softwareentwicklung	26
1.1.1 Software-Engineering	30
1.1.2 Vielfalt des Software-Engineerings	32
1.1.3 Software-Engineering und das Internet	35
1.2 Ethik des Software-Engineerings	36
1.3 Fallstudien	40
1.3.1 Ein Steuerungssystem für Insulinpumpen	41
1.3.2 Ein Patienteninformationssystem für die psychiatrische Ambulanz	43
1.3.3 Eine Wetterstation in Wildnisgebieten	45
1.3.4 Eine digitale Lernumgebung für Schulen	47
Kapitel 2 Softwareprozesse	53
2.1 Vorgehensmodelle	56
2.1.1 Das Wasserfallmodell	57
2.1.2 Inkrementelle Entwicklung	60
2.1.3 Integration und Konfiguration	63
2.2 Prozessaktivitäten	64
2.2.1 Softwarespezifikation	65
2.2.2 Softwareentwurf und -implementierung	67
2.2.3 Softwarevalidierung	70
2.2.4 Weiterentwicklung von Software	72
2.3 Umgang mit Änderungen	73
2.3.1 Softwareprototypen	74
2.3.2 Inkrementelle Auslieferung	75
2.4 Prozessverbesserung	77
Kapitel 3 Agile Softwareentwicklung	85
3.1 Agile Methoden	89
3.2 Techniken der agilen Entwicklung	91
3.2.1 User-Stories	93
3.2.2 Refactoring	95
3.2.3 Test-First-Entwicklung	96
3.2.4 Pair Programming	98

3.3	Agiles Projektmanagement	100
3.4	Skalieren von agilen Methoden	104
3.4.1	Praktische Probleme mit agilen Methoden	105
3.4.2	Agile und plangesteuerte Methoden	107
3.4.3	Agile Methoden für große Systeme	110
3.4.4	Organisationsübergreifende agile Methoden	113
Kapitel 4	Requirements-Engineering	119
4.1	Funktionale und nichtfunktionale Anforderungen	124
4.1.1	Funktionale Anforderungen	124
4.1.2	Nichtfunktionale Anforderungen	126
4.2	Prozesse des Requirements-Engineerings	131
4.3	Anforderungserhebung und -analyse	132
4.3.1	Techniken zur Anforderungserhebung	135
4.3.2	User-Stories und Szenarios	139
4.4	Anforderungsspezifikation	141
4.4.1	Spezifikation in natürlicher Sprache	143
4.4.2	Strukturierte Spezifikationen	144
4.4.3	Anwendungsfälle	146
4.4.4	Die Gesamtsystemspezifikation	148
4.5	Validierung von Anforderungen	151
4.6	Anforderungsänderung	153
4.6.1	Planung des Anforderungsmanagements	155
4.6.2	Anforderungsänderungsmanagement	156
Kapitel 5	Systemmodellierung	161
5.1	Kontextmodelle	165
5.2	Interaktionsmodelle	168
5.2.1	Anwendungsfallmodellierung	168
5.2.2	Sequenzdiagramme	170
5.3	Strukturelle Modelle	173
5.3.1	Klassendiagramme	173
5.3.2	Generalisierung	176
5.3.3	Aggregation	177
5.4	Verhaltensmodelle	178
5.4.1	Datenorientierte Modellierung	179
5.4.2	Ereignisgesteuerte Modellierung	180
5.4.3	Modellgetriebene Softwareentwicklung	183
5.5	Modellgetriebene Architektur	184
Kapitel 6	Entwurf der Architektur	191
6.1	Architektonische Entwurfsentscheidungen	195
6.2	Architektursichten	198
6.3	Architekturmuster	200
6.3.1	Schichtenarchitektur	202
6.3.2	Repository-Architektur	204

6.3.3	Client-Server-Architektur	206
6.3.4	Pipes-und-Filter-Architektur	208
6.4	Anwendungsarchitekturen	210
6.4.1	Transaktionsverarbeitende Systeme	212
6.4.2	Informationssysteme	213
6.4.3	Sprachverarbeitende Systeme	216
Kapitel 7	Entwurf und Implementierung	223
7.1	Objektorientierter Entwurf mit UML	226
7.1.1	Systemkontext und Interaktionen	226
7.1.2	Entwurf der Architektur	229
7.1.3	Bestimmung der Objektklassen	230
7.1.4	Entwurfsmodelle	232
7.1.5	Schnittstellenspezifikation	236
7.2	Entwurfsmuster	237
7.3	Implementierungsaspekte	241
7.3.1	Wiederverwendung	242
7.3.2	Konfigurationsverwaltung	244
7.3.3	Host-Ziel-Entwicklung	245
7.4	Open-Source-Entwicklung	248
7.4.1	Open-Source-Lizenzierung	250
Kapitel 8	Testen von Software	255
8.1	Entwicklertests	261
8.1.1	Modultests	262
8.1.2	Auswahl der Testfälle für Modultests	264
8.1.3	Testen von Komponenten	267
8.1.4	Testen von Systemen	270
8.2	Testgetriebene Entwicklung	273
8.3	Freigabetests	275
8.3.1	Anforderungsbasiertes Testen	276
8.3.2	Szenariobasiertes Testen	277
8.3.3	Leistungstests	279
8.4	Benutzertests	280
Kapitel 9	Softwareevolution	287
9.1	Evolutionsprozesse	292
9.2	Altsysteme	295
9.2.1	Verwaltung von Altsystemen	300
9.3	Softwarewartung	305
9.3.1	Vorhersagen des Wartungsaufwands	309
9.3.2	Software-Reengineering	311
9.3.3	Refactoring	314

Teil II Systemverlässlichkeit und Informationssicherheit

319

Kapitel 10	Verlässliche Systeme	321
10.1	Eigenschaften der Verlässlichkeit	324
10.2	Soziotechnische Systeme	328
10.2.1	Regulierung und Einhaltung der Vorschriften	331
10.3	Redundanz und Diversität	332
10.4	Verlässliche Prozesse	334
10.5	Formale Methoden und Verlässlichkeit	337
Kapitel 11	Reliability-Engineering	343
11.1	Verfügbarkeit und Zuverlässigkeit	347
11.2	Zuverlässigkeitsanforderungen	350
11.2.1	Zuverlässigkeitsmetriken	351
11.2.2	Nichtfunktionale Zuverlässigkeitsanforderungen	352
11.2.3	Funktionale Zuverlässigkeitsanforderungen	355
11.3	Fehlertolerante Architekturen	356
11.3.1	Schutzsysteme	357
11.3.2	Selbstüberwachende Architekturen	359
11.3.3	Diversitäre Programmierung	361
11.3.4	Softwarediversität	362
11.4	Zuverlässige Programmierung	364
11.5	Zuverlässigkeitsmessung	371
11.5.1	Betriebsprofile	373
Kapitel 12	Safety-Engineering	379
12.1	Betriebssicherheitskritische Systeme	382
12.2	Spezifikation der Betriebssicherheit	385
12.2.1	Gefahrenerkennung	387
12.2.2	Gefahrenbewertung	388
12.2.3	Gefahrenanalyse	391
12.2.4	Risikoreduzierung	393
12.3	Prozesse zur Entwicklung betriebssicherer Systeme	394
12.3.1	Prozesse für die Gewährleistung der Betriebssicherheit	396
12.3.2	Formale Verifikation	399
12.3.3	Modellprüfung	400
12.3.4	Statische Programmanalyse	402
12.4	Nachweis der Betriebssicherheit	404
12.4.1	Strukturierte Argumentationen	406
12.4.2	Betriebssicherheitsargumentationen für Software	408

Kapitel 13	Security-Engineering	417
13.1	Informationssicherheit und Verlässlichkeit	421
13.2	Informationssicherheit in Unternehmen und Organisationen	425
13.2.1	Bewertung von Informationssicherheitsrisiken	427
13.3	Anforderungen an die Informationssicherheit	428
13.3.1	Missbrauchsfälle	432
13.4	Entwerfen sicherer Systeme	434
13.4.1	Bewertung der Entwurfsrisiken	436
13.4.2	Architekturentwurf	440
13.4.3	Entwurfsrichtlinien	444
13.4.4	Informationssichere Systemprogrammierung	449
13.5	Testen und Gewährleistung der Informationssicherheit	451
Kapitel 14	Resilienz-Engineering	457
14.1	Cybersicherheit	462
14.2	Soziotechnische Resilienz	466
14.2.1	Menschliches Versagen	469
14.2.2	Betriebs- und Managementprozesse	473
14.3	Entwerfen resilienter Systeme	476
Teil III	Software-Engineering für Fortgeschrittene	489
Kapitel 15	Wiederverwendung von Software	491
15.1	Die Wiederverwendungslandschaft	495
15.2	Anwendungsframeworks	499
15.3	Softwareproduktlinien	502
15.4	Wiederverwendung von Anwendungssystemen	509
15.4.1	Konfigurierbare Anwendungssysteme	511
15.4.2	Integrierte Anwendungssysteme	514
Kapitel 16	Komponentenbasiertes Software-Engineering	521
16.1	Komponenten und Komponentenmodelle	525
16.1.1	Komponentenmodelle	528
16.2	Prozesse des komponentenbasierten Software-Engineerings	531
16.2.1	Komponentenbasiertes Software-Engineering für Wiederverwendung	533
16.2.2	Komponentenbasiertes Software-Engineering mit Wiederverwendung	536
16.3	Komposition von Komponenten	539

Kapitel 17	Entwicklung verteilter Systeme	551
17.1	Verteilte Systeme	554
17.1.1	Kommunikationsmodelle	558
17.1.2	Middleware	560
17.2	Client-Server-Systeme	562
17.3	Architekturmuster für verteilte Systeme	564
17.3.1	Master-Slave-Architekturen.	565
17.3.2	Zweischichtige Client-Server-Architekturen	566
17.3.3	Mehrschichtige Client-Server-Architekturen	568
17.3.4	Verteilte Komponentenarchitekturen	570
17.3.5	Peer-to-Peer-Architekturen	573
17.4	Software als Service.	576
Kapitel 18	Serviceorientiertes Software-Engineering	585
18.1	Serviceorientierte Architektur	591
18.1.1	Servicekomponenten in einer serviceorientierten Architektur . . .	593
18.2	REST-konforme Services	596
18.3	Service-Engineering.	599
18.3.1	Ermittlung von Servicekandidaten	600
18.3.2	Entwerfen von Serviceschnittstellen	603
18.3.3	Implementierung und Bereitstellung der Services	607
18.4	Kombinationen von Services	608
18.4.1	Entwurf und Implementierung des Workflows	611
18.4.2	Testen von Services	614
Kapitel 19	Systems-Engineering	619
19.1	Soziotechnische Systeme	625
19.1.1	Typische Systemeigenschaften	628
19.1.2	Nichtdeterminismus	631
19.1.3	Erfolgskriterien.	632
19.2	Entwurfsplanung	633
19.3	Systembeschaffung	636
19.4	Systementwicklung	641
19.5	Systembetrieb und Systemevolution	645
19.5.1	Systemevolution.	646
Kapitel 20	Systeme von Systemen	651
20.1	Systemkomplexität	656
20.2	Klassifikation von Systemen von Systemen.	660
20.3	Reduktionismus und komplexe Systeme	663
20.4	SvS-Engineering.	666
20.4.1	Schnittstellenentwicklung.	668
20.4.2	Integration und Inbetriebnahme	670
20.5	Architektur von Systemen von Systemen	672
20.5.1	Architekturmuster für Systeme von Systemen.	675

Kapitel 21	Entwicklung von Echtzeitsoftware	683
21.1	Entwurf eingebetteter Systeme	686
21.1.1	Modellierung von Echtzeitsystemen	691
21.1.2	Programmierung von Echtzeitsystemen	693
21.2	Architekturmuster für Echtzeitsoftware.	694
21.2.1	Beobachten und Reagieren	695
21.2.2	Umgebungssteuerung	697
21.2.3	Pipelineverarbeitung	699
21.3	Analyse des Zeitverhaltens.	701
21.4	Echtzeitbetriebssysteme	706
21.4.1	Prozessverwaltung.	707
Teil IV	Softwaremanagement	713
Kapitel 22	Projektmanagement	715
22.1	Risikomanagement	719
22.1.1	Risikoerkennung	721
22.1.2	Risikoanalyse.	722
22.1.3	Risikoplanung	725
22.1.4	Risikoüberwachung.	726
22.2	Personalmanagement	726
22.2.1	Mitarbeitermotivation	727
22.3	Teamwork	731
22.3.1	Teammitglieder auswählen.	733
22.3.2	Organisation der Gruppe.	735
22.3.3	Kommunikation in der Gruppe.	737
Kapitel 23	Projektplanung	743
23.1	Preiskalkulation für Software.	747
23.2	Plangesteuerte Entwicklung	748
23.2.1	Projektpläne.	749
23.2.2	Der Planungsprozess	750
23.3	Zeitplanung.	752
23.3.1	Darstellung des Zeitplans	753
23.4	Agile Planung	758
23.5	Schätztechniken	760
23.5.1	Algorithmische Kostenmodellierung	762
23.6	Das COCOMO-II-Modell.	764
23.6.1	Das Application-Composition-Modell	766
23.6.2	Das Early-Design-Modell.	767
23.6.3	Das Reuse-Modell	768
23.6.4	Das Post-Architecture-Modell.	769
23.6.5	Projektdauer und Personalplanung	772

Kapitel 24	Qualitätsmanagement	779
24.1	Softwarequalität	783
24.2	Softwarestandards	786
24.2.1	Der Rahmenstandard ISO 9001	788
24.3	Reviews und Inspektionen	790
24.3.1	Der Review-Prozess	792
24.3.2	Programminspektionen	793
24.4	Qualitätsmanagement und agile Entwicklung	795
24.5	Softwaremessung	797
24.5.1	Produktmetriken	801
24.5.2	Softwarekomponentenanalyse	804
24.5.3	Mehrdeutigkeit von Messungen	805
24.5.4	Softwareanalytik	807
Kapitel 25	Konfigurationsmanagement	813
25.1	Versionsmanagement	819
25.2	Systemerstellung	825
25.3	Änderungsmanagement	831
25.4	Release-Management	837
Glossar		843
Literatur A-Z		867
Register		885