

Inhaltsverzeichnis

I	Klassische Aspekte des Software Engineering	1
1	Allgemeine Aspekte	3
1.1	Softwareentwicklung gestern	3
1.2	Softwareentwicklung heute	5
1.3	Was ist Software Engineering?	7
1.4	Eigenschaften von Software	8
1.5	Anwendungssysteme und Geschäftsprozesse	10
2	Technische Aspekte	17
2.1	Kriterien für die Softwarequalität	17
2.2	Konventionelle Softwareentwicklung	19
3	Vorgehensmodelle	27
3.1	Wasserfallmodell und allgemeines V-Modell	28
3.2	Iterative, inkrementelle und evolutionäre Vorgehensweisen	31
3.3	Rational Unified Process	32
3.4	V-Modell XT	35
3.5	Extreme Programming	38
4	Projektmanagement	43
4.1	Problemstellung	43
4.2	Voruntersuchung	45
4.3	Planung	46
4.4	Kontrolle und -steuerung	48
4.5	Änderungsmanagement	50
4.6	Risikomanagement	51
4.7	Teammanagement	52
4.8	Abschluss	54
5	Qualitätsmanagement und Qualitätssicherung	57
5.1	Einleitung	57
5.2	Qualitätsmanagement	59
5.3	Konstruktive Qualitätssicherung	61
5.4	Analytische Qualitätssicherung	61

- 6 Softwareentwicklungswerkzeuge 67**
 - 6.1 Vorteile des Werkzeugeinsatzes 67
 - 6.2 Spezielle Werkzeuge 67
 - 6.3 Softwareentwicklungsumgebungen 69

- 7 Objektorientierte Softwareentwicklung und die UML 71**
 - 7.1 Objektorientierte Softwareentwicklung 71
 - 7.2 UML im Überblick 73
 - 7.3 Modellbasierte Softwareentwicklung 73
 - 7.4 Modellgetriebene Softwareentwicklung 74
 - 7.5 MOOS: Methodische objektorientierte Softwareentwicklung 75

- II UML-Strukturmodelle 77**

- 8 Objektdiagramm 79**
 - 8.1 Einstieg 79
 - 8.2 Zustände 80
 - 8.3 Operationen und Verhalten 82
 - 8.4 Nachrichten und Verbindungen 82
 - 8.5 Identität 84
 - 8.6 Fallbeispiel »Bestellwesen« 84

- 9 Klassendiagramm 87**
 - 9.1 Einleitung 87
 - 9.2 Klassen und Instanzen 87
 - 9.3 Attribute und Operationen 89
 - 9.4 Assoziationen 92
 - 9.5 Aggregationen und Kompositionen 95
 - 9.6 Assoziationsklassen 97
 - 9.7 n-äre Assoziationen 98
 - 9.8 Abgeleitete Elemente 99
 - 9.9 Standardoperationen 100
 - 9.10 Generalisierungsbeziehungen 102
 - 9.11 Klassendiagramm »Bestellwesen« 104

- 10 Fortgeschrittene Konzepte 107**
 - 10.1 Sichtbarkeiten und Schnittstellen 107
 - 10.2 Stereotype 109
 - 10.3 Abhängigkeiten 109
 - 10.4 Abstrakte Klassen und Interfaces 110
 - 10.5 Typen und Attributdeklarationen 113
 - 10.6 Klassenattribute und Klassenoperationen 114
 - 10.7 Parametrisierte Klassen 115
 - 10.8 Pakete 117
 - 10.9 Komponenten und Teilsysteme 119
 - 10.10 Kollaborationstypen und Kollaborationen 121

11	Textuelle Spezifikation mit der OCL	123
11.1	Object Constraint Language (OCL)	123
11.2	Zusicherungen	125
11.3	Textuelle Spezifikationen	127
11.4	Abschließende Bemerkungen zu Strukturmodellen	129
III	UML-Funktions- und -Verhaltensmodelle	131
12	Anwendungsfalldiagramm	133
12.1	Akteure	133
12.2	Anwendungsfälle	134
12.3	Beziehungen zwischen Anwendungsfällen	139
13	Aktivitätsdiagramm	147
13.1	Aktionsknoten und Kontrollflusskanten	147
13.2	Entscheidungs- und Verbindungsknoten	148
13.3	Parallelisierungs- und Synchronisierungsknoten	150
14	Interaktionsdiagramme	153
14.1	Sequenzdiagramm	153
14.2	Synchrone Nachrichten und Aktivierungen	158
14.3	Asynchrone Nachrichten und aktive Objekte	160
14.4	Kommunikationsdiagramm	162
14.5	Diskussion	164
15	Zustandsdiagramm	165
15.1	Zustände	165
15.2	Ereignisse	166
15.3	Zustandsübergänge und Bedingungen	168
15.4	Aktionen	169
15.5	Initialzustand und Terminalzustände	171
15.6	Zusammengesetzte Zustände	172
15.7	Abschließende Bemerkungen zur UML	174
IV	Anforderungsermittlung	177
16	Ziele und Vorgehensweise	179
16.1	Ausgangspunkt: Das Lastenheft	179
16.2	Allgemeine Aspekte	181
16.3	Funktionale und nichtfunktionale Anforderungen	182
16.4	Aktivitäten	183
16.5	Anforderungsermittlung in Moos	184

17	Anwendungsfallmodellierung	189
17.1	Einstieg, (Re-)Konstruktion der Fachsprache, Glossar	189
17.2	Szenarien	192
17.3	Akteure	193
17.4	Anwendungsfälle	194
17.5	Beziehungen zwischen Anwendungsfällen	197
17.6	Pakete	199
17.7	Zusammenfassung der Merkgeln	200
18	Domänenklassenmodellierung	201
18.1	Vorgehensweise	201
18.2	Klassen und Assoziationen	202
18.3	Generalisierungsbeziehungen	203
18.4	Modellbereinigung	203
18.5	Textuelle Spezifikationen	205
18.6	Pakete	206
18.7	Zusammenfassung der Merkgeln	207
19	Benutzungsschnittstelle	209
19.1	Anwendungsfallbezogene Szenen und Navigation	211
19.2	Geschäftsprozessorientierte Szenen und Navigation	212
19.3	Prototyping	216
20	Verhaltensmodellierung	217
20.1	Ablauforientiertes Verhalten von Geschäftsprozessfunktionen	217
20.2	Ablauforientiertes Verhalten von Anwendungsfällen	218
20.3	Zustandsorientiertes Verhalten	223
21	Qualitätssicherung	225
21.1	Validierung	225
21.2	Verifikation	226
21.3	Testfallspezifizierung	227
22	SeminarIS: Anforderungsermittlung	231
22.1	Geschäftsprozesse und Auftragsvergabe	231
22.2	Lastenheft	233
22.3	Glossar	238
22.4	Szenarien	240
22.5	Akteure	242
22.6	Anwendungsfälle	243
22.7	Domänenklassen und Assoziationen	251
22.8	Generalisierungsbeziehungen	255
22.9	Modellbereinigung	256
22.10	Pakete	257
22.11	Textuelle Spezifikationen	258

22.12	Benutzungsschnittstelle	258
22.13	Verhaltensmodellierung	259
22.14	Qualitätssicherung	260
22.15	Abschließende Bemerkungen zur Anforderungsermittlung	262
V	Softwarespezifizierung und Architekturkonzeption	265
23	Ziele und Vorgehensweise	267
23.1	Einführung	267
23.2	Softwarespezifizierung in Moos	268
24	Transformation der Anforderungsspezifikation	271
24.1	Einleitung	271
24.2	Domänenklassen → Entitätsklassen	273
24.3	Anwendungsfälle → AF-Klassen	274
24.4	GF-Anwendungsfälle → GF-Klassen	276
24.5	Abläufe und Interaktionsdiagramme → Operationen	277
25	Überarbeitung des Klassenmodells	281
25.1	1-1-Assoziation oder <i>eine</i> Klasse?	281
25.2	Attribute oder Generalisierung?	283
25.3	Zusammenfassung der Heuristiken	286
25.4	Modellbereinigung	287
25.5	Pakete	287
26	Modellverfeinerung	289
26.1	Benutzungsschnittstelle	289
26.2	AF-E-Kollaborationstyp	291
26.3	Verhaltensmodellierung	291
27	Qualitätssicherung	297
27.1	Validierung	297
27.2	Intra-Modell-Verifikation	297
27.3	Inter-Modell-Verifikation	298
27.4	Testfallspezifizierung	300
28	SeminarIS: Softwarespezifizierung	303
28.1	Entitätsklassen	303
28.2	Anwendungsfälle	304
28.3	Interaktionsdiagramme	305
28.4	Modellaufbereitung	305
28.5	Pakete	306
28.6	Verhaltensmodellierung	307
28.7	Verifikation	310
28.8	Testfallspezifizierung	311
28.9	Abschließende Bemerkungen zur Softwarespezifizierung	312

29	Architekturkonzeption	315
29.1	Ziele und Vorgehensweise	315
29.2	Globale Systemaspekte	317
29.3	Schichtenarchitekturen	320
29.4	Schnittstellen-, Kontroll- und Entitätsklassen	322
29.5	Komponenten	324
29.6	Qualitätssicherung für Architekturen	326
29.7	SeminarIS: Architekturkonzeption	326
29.8	Abschließende Bemerkungen zur Architekturkonzeption	328
VI	Grundlagen des Entwurfs	329
30	Entwurfskonzepte	331
30.1	Geheimnisprinzip	331
30.2	Schwache Kopplung und starke Kohäsion	333
30.3	Generalisierung und Konformität	340
30.4	Entwurf mit Verträgen	345
30.5	Interfaces	348
30.6	Parametrisierte Klassen und Interfaces	351
30.7	Pakete und Teilsysteme	352
30.8	Zusammenfassung der Regeln und Heuristiken	353
31	Entwurfsmuster	355
31.1	Einführung	355
31.2	Erzeugende Muster	359
31.3	Strukturmuster	365
31.4	Verhaltensmuster	368
31.5	Auswahl von Entwurfsmustern	375
31.6	Einsatz von Entwurfsmustern	377
32	Berücksichtigung von Java im Entwurf	379
32.1	Sichtbarkeit	379
32.2	Abstrakte Klassen und Interfaces	379
32.3	Polymorphismus und dynamisches Binden	380
32.4	Generizität	381
32.5	Aufzählungstypen	382
32.6	Komponenten und Teilsysteme	383
32.7	Abschließende Bemerkungen zu den Grundlagen des Entwurfs	384
VII	Grobentwurf	385
33	Ziele und Vorgehensweise	387
33.1	Warum Grob- und Feinentwurf?	387
33.2	Grobentwurf in Moos	388
33.3	Grundsätzliche Entwurfsentscheidungen	389
33.4	Merkmale der Grobentwurfsspezifikation	390

34	Transformation der Softwarespezifikation	391
34.1	Große Systeme: Pakete → Komponenten	391
34.2	Paketbeziehungen → Komponenten-Interfaces	392
34.3	Spezifikationsklassen → Grobentwurfsklassen	395
34.4	Assoziationen	397
34.5	include- und extend-Beziehungen	400
34.6	Kleine Systeme: Pakete und Klassen	403
35	Modellverfeinerung	407
35.1	Benutzungsschnittstelle	407
35.2	Externe Systeme	408
35.3	Präzisierung der Komponentenschnittstellen	410
35.4	Operationen für Schnittstellen-, Kontroll- und Entitätsklassen	412
36	Verhaltensmodellierung	415
36.1	AAS-K-E-Kollaborationstyp	415
36.2	Anwendungsfälle	416
36.3	Komplexe Operationen	418
37	Qualitätssicherung	421
37.1	Validierung	421
37.2	Verifikation	421
37.3	Testfallspezifizierung	422
38	SeminarIS: Grobentwurf	425
38.1	Transformation der Softwarespezifikation	425
38.2	Pakete	429
38.3	Verhaltensmodellierung	430
38.4	Abschließende Bemerkungen zum Grobentwurf	434
VIII	Feinentwurf	437
39	Ziele und Vorgehensweise	439
39.1	Feinentwurf in Moos	439
40	Transformation der Grobentwurfsspezifikation	441
40.1	Einleitung	441
40.2	Elemente der Benutzungsoberfläche	442
40.3	Haupt- und Akteur-Schnittstellenklassen → JFrame	446
40.4	Akteur-Anwendungsfall-Schnittstellenklassen → JPanel	447
40.5	AAS-E-Beziehungen → Sicht-Schnittstellenklassen	448
40.6	Anwendungslogik und Anwendungsdaten	451

41	Transformation zur Zielsprache	453
41.1	Assoziationen: Standardfälle	453
41.2	Spezielle Assoziationen und Assoziationsklassen	462
41.3	Echte Ganzes-Klassen	464
41.4	Zustandsdiagramme	466
41.5	Verhaltensmodellierung	470
41.6	Pakete	472
42	Persistente Datenhaltung	475
42.1	Grundlagen	475
42.2	Abbildung von Klassen auf Tabellen	477
42.3	Objektidentifikatoren	478
42.4	Assoziationen	479
42.5	Generalisierungsbeziehungen	482
42.6	Laden externer Objekte	483
42.7	Transaktionen	483
42.8	Persistenz-Rahmenwerke	485
42.9	Benutzung von Persistenz-Rahmenwerken	486
43	Qualitätssicherung	489
43.1	Validierung	489
43.2	Verifikation	489
43.3	Testfallspezifizierung	490
44	Weitere Aspekte	495
44.1	Redundanzvermeidung	495
44.2	Refaktorisierung	496
44.3	Wiederverwendung	497
44.4	Hard- und Softwareumgebung	498
45	SeminarIS: Feinentwurf	499
45.1	Benutzungsschnittstelle	499
45.2	Modellverfeinerung	504
45.3	Pakete	506
45.4	Abschließende Bemerkungen zum Feinentwurf	507
46	Abschließende Bemerkungen zum Buch	509
46.1	Rückblick	509
46.2	Ausblick	510
	Quellen- und Literaturangaben	513
	Stichwortverzeichnis	525