

Inhalt

Vorwort	XIII
Vorwort zur siebten Auflage	XIV
1 Einleitung	1
1.1 Softwarearchitekten	5
1.2 Effektiv, agil und pragmatisch	6
1.3 Wer sollte dieses Buch lesen?	9
1.4 Wegweiser durch das Buch	10
1.5 Webseite zum Buch	12
1.6 Weiterführende Literatur	12
1.7 Danksagung	13
2 Architektur und Architekten	15
2.1 Was ist Softwarearchitektur?	16
2.1.1 Darum Softwarearchitektur!	20
2.2 Die Aufgaben von Softwarearchitekten	21
2.3 Wie entstehen Architekturen?	26
2.4 In welchem Kontext steht Architektur?	29
2.5 Weiterführende Literatur	32
3 Vorgehen bei der Architekturentwicklung	33
3.1 Informationen sammeln	37
3.2 Anforderungen klären	38
3.2.1 Was ist die Kernaufgabe des Systems?	38
3.2.2 Welche Kategorie von System?	39
3.2.3 Wesentliche Qualitätsanforderungen ermitteln	39
3.2.4 Relevante Stakeholder ermitteln	44
3.2.5 Fachlichen und technischen Kontext ermitteln	45
3.3 Einflussfaktoren und Randbedingungen ermitteln	47
3.4 Entwerfen und kommunizieren	53
3.5 Umsetzung begleiten	54
3.6 Lösungsstrategien entwickeln	55
3.7 Weiterführende Literatur	57

4	Entwurf: Grundlagen, Methoden und Muster	59
4.1	Grundlagen	61
4.1.1	Grundsätze des Entwurfs (Maxime)	62
4.1.2	Prinzipien	65
4.1.3	SOLID-Prinzipien des objektorientierten Entwurfs	71
4.1.3.1	Offen-Geschlossen-Prinzip	71
4.1.3.2	Liskov-Substitutionsprinzip (LSP)	73
4.1.3.3	Interface Segregation Principle (ISP)	74
4.1.3.4	Dependency Inversion Principle (DIP)	76
4.2	Heuristiken	79
4.3	Entwurfsmethoden	84
4.3.1	Fachlich motivierter Entwurf: Domain-Driven Design	84
4.3.2	Der WAM-Ansatz	87
4.3.3	Quality-Driven Software Architecture	89
4.3.4	Top-down und Bottom-up	97
4.4	Schnittstellen entwerfen	98
4.4.1	Anforderungen an Schnittstellen	99
4.4.2	Worauf müssen Sie achten?	100
4.4.3	Tipps zum Entwurf von Schnittstellen	101
4.5	Architekturstile und -muster	102
4.5.1	Datenflussarchitekturstil	103
4.5.1.1	Architekturstil Batch-Sequentiell	103
4.5.1.2	Architekturstil Pipes und Filter	104
4.5.2	Datenzentrierter Architekturstil	107
4.5.2.1	Repository	107
4.5.2.2	Blackboard	108
4.5.3	Hierarchische Architekturstile	109
4.5.3.1	Master-Slave	109
4.5.3.2	Schichten (Layer)	110
4.5.3.3	Architekturstil Ports-und-Adapter	113
4.5.4	Architekturstile verteilter Systeme	116
4.5.4.1	Client-Server	117
4.5.4.2	Command Query Responsibility Segregation	117
4.5.4.3	Broker	119
4.5.4.4	Peer-to-Peer	120
4.5.5	Ereignisbasierte Systeme – Event Systems	121
4.5.5.1	Ungepufferte Event-Kommunikation	122
4.5.5.2	Message- oder Event-Queue-Architekturen	122
4.5.5.3	Message-Service-Architekturen	123
4.5.6	Interaktionsorientierte Systeme	124
4.5.6.1	Model-View-Controller	124
4.5.6.2	Presentation Model	125
4.5.7	Weitere Architekturstile und -muster	128
4.6	Entwurfsmuster	130

4.6.1	Entwurf mit Mustern	130
4.6.2	Adapter.	131
4.6.3	Beobachter (Observer)	132
4.6.4	Dekorierer (Decorator)	133
4.6.5	Stellvertreter (Proxy)	134
4.6.6	Fassade.	135
4.6.7	Zustand (State)	136
4.7	Weiterführende Literatur	137
5	Kommunikation und Dokumentation von Architekturen	139
5.1	Architekten müssen kommunizieren und dokumentieren.	140
5.2	Effektive Architekturdokumentation	142
5.2.1	Anforderungen an Architekturdokumentation	142
5.2.2	Regeln für gute Architekturdokumentation	145
5.3	Typische Architekturdokumente	148
5.3.1	Zentrale Architekturbeschreibung	149
5.3.2	Architekturüberblick	152
5.3.3	Dokumentationsübersicht	152
5.3.4	Übersichtspräsentation der Architektur	152
5.3.5	Architekturtapete	153
5.4	Sichten	153
5.4.1	Sichten in der Softwarearchitektur	154
5.4.2	Vier Arten von Sichten	156
5.4.3	Entwurf der Sichten	158
5.5	Kontextabgrenzung	160
5.5.1	Elemente der Kontextabgrenzung	160
5.5.2	Notation der Kontextabgrenzung	161
5.5.3	Entwurf der Kontextabgrenzung	161
5.6	Bausteinsicht	162
5.6.1	Elemente der Bausteinsicht	166
5.6.2	Notation der Bausteinsicht	167
5.6.3	Entwurf der Bausteinsicht	168
5.7	Laufzeitsicht	169
5.7.1	Elemente der Laufzeitsicht	170
5.7.2	Notation der Laufzeitsicht	171
5.7.3	Entwurf der Laufzeitsicht	172
5.8	Verteilungssicht	172
5.8.1	Elemente der Verteilungssicht	173
5.8.2	Notation der Verteilungssicht	173
5.8.3	Entwurf der Verteilungssicht	174
5.9	Dokumentation von Schnittstellen	175
5.10	Dokumentation technischer Konzepte	178
5.11	Werkzeuge zur Dokumentation	178
5.12	TOGAF zur Architekturdokumentation	180

5.13	Weiterführende Literatur	181
6	Modellierung für Softwarearchitekten.....	183
6.1	Modelle als Arbeitsmittel	183
6.1.1	Grafische oder textuelle Modellierung	185
6.2	UML 2 für Softwarearchitekten	186
6.2.1	Die Diagrammarten der UML 2	187
6.2.2	Die Bausteine von Architekturen	189
6.2.3	Schnittstellen	190
6.2.4	Die Bausteinsicht	191
6.2.5	Die Verteilungssicht	193
6.2.6	Die Laufzeitsicht	195
6.2.7	Darum UML	198
6.2.8	Darum nicht UML	199
6.3	Tipps zur Modellierung	199
6.4	Weiterführende Literatur	200
7	Technische Konzepte und typische Architekturaspekte	201
7.1	Persistenz	205
7.1.1	Motivation	205
7.1.2	Einflussfaktoren und Entscheidungskriterien	208
7.1.2.1	Art der zu speichernden Daten	209
7.1.2.2	Konsistenz und Verfügbarkeit (ACID, BASE oder CAP)	210
7.1.2.3	Zugriff und Navigation	212
7.1.2.4	Deployment und Betrieb	212
7.1.3	Lösungsmuster	213
7.1.3.1	Persistenzschicht	213
7.1.3.2	DAO: Eine Miniatur-Persistenzschicht	217
7.1.4	Bekannte Risiken und Probleme	218
7.1.5	Weitere Themen zu Persistenz	219
7.1.6	Zusammenhang zu anderen Aspekten	223
7.1.7	Praktische Vertiefung	224
7.1.8	Weiterführende Literatur	225
7.2	Geschäftsregeln	226
7.2.1	Motivation	226
7.2.2	Funktionsweise von Regelmaschinen	229
7.2.3	Kriterien pro & kontra Regelmaschinen	231
7.2.4	Mögliche Probleme	232
7.2.5	Weiterführende Literatur	233
7.3	Integration	233
7.3.1	Motivation	233
7.3.2	Typische Probleme	235
7.3.3	Lösungskonzepte	236
7.3.4	Entwurfsmuster zur Integration	240

7.3.5	Konsequenzen und Risiken	241
7.3.6	Zusammenhang mit anderen Aspekten	243
7.3.7	Weiterführende Literatur	244
7.4	Verteilung	244
7.4.1	Motivation	244
7.4.2	Typische Probleme	245
7.4.3	Lösungskonzept	245
7.4.4	Konsequenzen und Risiken	247
7.4.5	Zusammenhang mit anderen Aspekten	247
7.4.6	Weiterführende Literatur	247
7.5	Kommunikation	248
7.5.1	Motivation	248
7.5.2	Entscheidungsalternativen	248
7.5.3	Grundbegriffe der Kommunikation	248
7.5.4	Weiterführende Literatur	252
7.6	Grafische Oberflächen (GUI)	254
7.6.1	Motivation	254
7.6.2	Einflussfaktoren und Entscheidungskriterien	254
7.6.3	GUI-relevante Architekturmuster	256
7.6.4	Struktur und Ergonomie von Benutzeroberflächen	257
7.6.5	Bekannte Risiken und Probleme	258
7.6.6	Zusammenhang zu anderen Aspekten	260
7.7	Geschäftsprozess-Management: Ablaufsteuerung im Großen	261
7.7.1	Workflow-Sprachen	263
7.7.2	Vorhersagbarkeit	266
7.7.3	Zweck der Ablaufsteuerung	267
7.7.4	Lösungsansätze	269
7.7.5	Integration von Workflow-Systemen	272
7.7.6	Mächtigkeit von WfMS	273
7.7.7	Weiterführende Literatur	273
7.8	Sicherheit	274
7.8.1	Motivation – Was ist IT-Sicherheit?	274
7.8.2	Sicherheitsziele	275
7.8.3	Lösungskonzepte	277
7.8.4	Security Engineering mit Patterns	284
7.8.5	Weiterführende Literatur	285
7.9	Protokollierung	286
7.9.1	Typische Probleme	286
7.9.2	Lösungskonzept	287
7.9.3	Zusammenhang mit anderen Aspekten	288
7.9.4	Weiterführende Literatur	288
7.10	Ausnahme- und Fehlerbehandlung	289
7.10.1	Motivation	289
7.10.2	Fehlerkategorien schaffen Klarheit	291

7.10.3	Muster zur Fehlerbehandlung	293
7.10.4	Mögliche Probleme	294
7.10.5	Zusammenhang mit anderen Aspekten	295
7.10.6	Weiterführende Literatur	296
7.11	Skalierbarkeit	296
7.11.1	Was bedeutet Skalierbarkeit?	296
7.11.2	Skalierungsstrategien	297
7.11.3	Elastizität	297
7.11.4	Scale-Up-Strategie	297
7.11.5	Vertikale Scale-Out-Strategie	298
7.11.6	Horizontale Scale-Out-Strategie	298
7.11.7	Der Strategiemix	298
7.11.8	Allgemeine Daumenregeln	299
7.11.9	CPU-Power	300
7.11.10	GPU-Power	300
7.11.11	RAIDs, SANs und andere Speichersysteme	300
7.11.12	Bussysteme für die Speicheranbindung	301
7.11.13	Geringere Bandbreite im Netz	301
8	Bewertung von Softwarearchitekturen	303
8.1	Qualitative Architekturbewertung	306
8.2	Quantitative Bewertung durch Metriken	313
8.3	Werkzeuge zur Bewertung	315
8.4	Weiterführende Literatur	316
9	Systematische Verbesserung und Evolution	317
9.1	Wege in den Abgrund	319
9.2	Systematisch verbessern	320
9.3	Bewährte Praktiken und Muster	323
9.4	Analyse: Probleme identifizieren	325
9.5	Evaluate: Probleme und Maßnahmen bewerten	327
9.6	Improve: Verbesserungsmaßnahmen planen und durchführen	328
9.6.1	Maxime für Verbesserungsprojekte	328
9.6.2	Kategorien von Verbesserungsmaßnahmen	328
9.7	Crosscutting: phasenübergreifende Praktiken	332
9.8	Mehr zu AIM ⁴²	333
9.9	Weiterführende Literatur	333
10	Serviceorientierte Architekturen (SOA) und Microservices	335
10.1	Was ist SOA?	336
10.2	So funktionieren Services	341
10.3	SOA in der Praxis	342
10.4	SOA und Softwarearchitektur	345
10.5	Microservices	345

10.6	Weiterführende Literatur	350
11	Enterprise-IT-Architektur	351
11.1	Wozu Architekturebenen?	352
11.2	Aufgaben von Enterprise-Architekten	353
11.2.1	Management der Infrastrukturkosten	353
11.2.2	Management des IS-Portfolios	354
11.2.3	Definition von Referenzarchitekturen	355
11.2.4	Weitere Aufgaben	357
11.3	Weiterführende Literatur	359
12	Beispiele von Softwarearchitekturen	361
12.1	Beispiel: Datenmigration im Finanzwesen	362
12.2	Beispiel: Kampagnenmanagement im CRM	379
13	Werkzeuge für Softwarearchitekten	411
13.1	Kategorien von Werkzeugen	411
13.2	Typische Auswahlkriterien	414
14	iSAQB Curriculum	417
14.1	Standardisierte Lehrpläne für Softwarearchitekten	418
14.1.1	Grundlagenausbildung und Zertifizierung <i>Foundation-Level</i>	418
14.1.2	Fortgeschrittene Aus- und Weiterbildung (<i>Advanced-Level</i>)	419
14.2	Können, Wissen und Verstehen	420
14.3	Voraussetzungen und Abgrenzungen	420
14.4	Struktur des iSAQB-Foundation-Level-Lehrplans	421
14.5	Zertifizierung nach dem iSAQB-Lehrplan	424
15	Nachwort: Architektonien	425
15.1	In sechs Stationen um die (IT-)Welt	425
15.2	Ratschläge aus dem architektonischen Manifest	428
16	Literatur	433
	Stichwortverzeichnis	437