

<b>Vorwort</b> .....	<b>XIII</b>
Vorwort zur neunten Auflage .....	XIV
<b>1 Einleitung</b> .....	<b>1</b>
1.1 Softwarearchitekt(inn)en .....	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
<b>2 Architektur und Architekten</b> .....	<b>15</b>
2.1 Was ist Softwarearchitektur? .....	16
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
<b>3 Vorgehen bei der Architekturentwicklung</b> .....	<b>33</b>
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

<b>4</b>	<b>Entwurf: Grundlagen, Methoden und Muster</b> .....	<b>59</b>
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	Domain-Driven Design (Entwurf nach Fachlichkeit) .....	84
4.3.2	Quality-Driven Software Architecture .....	89
4.3.3	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
<b>5</b>	<b>Kommunikation und Dokumentation von Architekturen</b>	<b>139</b>
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	144
5.3	Typische Architekturdokumente	147
5.3.1	Zentrale Architekturbeschreibung	148
5.3.2	Architekturüberblick	151
5.3.3	Dokumentationsübersicht	151
5.3.4	Übersichtspräsentation der Architektur	151
5.3.5	Architekturtapete	152
5.4	Sichten	152
5.4.1	Sichten in der Softwarearchitektur	153
5.4.2	Vier Arten von Sichten	155
5.4.3	Entwurf der Sichten	157
5.5	Kontextabgrenzung	159
5.5.1	Elemente der Kontextabgrenzung	159
5.5.2	Notation der Kontextabgrenzung	160
5.5.3	Entwurf der Kontextabgrenzung	160
5.6	Bausteinsicht	161
5.6.1	Elemente der Bausteinsicht	165
5.6.2	Notation der Bausteinsicht	166
5.6.3	Entwurf der Bausteinsicht	167
5.7	Laufzeitsicht	168
5.7.1	Elemente der Laufzeitsicht	169
5.7.2	Notation der Laufzeitsicht	170
5.7.3	Entwurf der Laufzeitsicht	171
5.8	Verteilungssicht	171
5.8.1	Elemente der Verteilungssicht	172
5.8.2	Notation der Verteilungssicht	172
5.8.3	Entwurf der Verteilungssicht	173
5.9	Dokumentation von Schnittstellen	174
5.10	Dokumentation technischer Konzepte	177
5.11	Werkzeuge zur Dokumentation	177
5.12	TOGAF zur Architekturdokumentation	179
5.13	Weiterführende Literatur	181

<b>6</b>	<b>Modellierung für Softwarearchitekten</b>	<b>183</b>
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
<b>7</b>	<b>Technische Konzepte und typische Architekturaspekte</b>	<b>201</b>
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	Zusammenhang mit anderen Aspekten	241
7.3.6	Weiterführende Literatur	242

7.4	Verteilung .....	242
7.4.1	Motivation .....	242
7.4.2	Typische Probleme .....	243
7.4.3	Lösungskonzept .....	243
7.4.4	Konsequenzen und Risiken .....	245
7.4.5	Zusammenhang mit anderen Aspekten .....	245
7.4.6	Weiterführende Literatur .....	245
7.5	Kommunikation .....	246
7.5.1	Motivation .....	246
7.5.2	Entscheidungsalternativen .....	246
7.5.3	Grundbegriffe der Kommunikation .....	246
7.5.4	Weiterführende Literatur .....	250
7.6	Grafische Oberflächen (GUI) .....	251
7.6.1	Motivation .....	251
7.6.2	Einflussfaktoren und Entscheidungskriterien .....	251
7.6.3	GUI-relevante Architekturmuster .....	253
7.6.4	Struktur und Ergonomie von Benutzeroberflächen .....	254
7.6.5	Bekannte Risiken und Probleme .....	255
7.6.6	Zusammenhang zu anderen Aspekten .....	257
7.7	Geschäftsprozess-Management: Ablaufsteuerung im Großen .....	258
7.7.1	Workflow-Sprachen .....	260
7.7.2	Vorhersagbarkeit .....	263
7.7.3	Zweck der Ablaufsteuerung .....	264
7.7.4	Lösungsansätze .....	266
7.7.5	Integration von Workflow-Systemen .....	269
7.7.6	Mächtigkeit von WfMS .....	270
7.7.7	Weiterführende Literatur .....	270
7.8	Sicherheit .....	271
7.8.1	Motivation – Was ist IT-Sicherheit? .....	271
7.8.2	Sicherheitsziele .....	272
7.8.3	Lösungskonzepte .....	274
7.8.4	Security Engineering mit Patterns .....	281
7.8.5	Weiterführende Literatur .....	282
7.9	Protokollierung .....	283
7.9.1	Typische Probleme .....	283
7.9.2	Lösungskonzept .....	284
7.9.3	Zusammenhang mit anderen Aspekten .....	285
7.9.4	Weiterführende Literatur .....	285
7.10	Ausnahme- und Fehlerbehandlung .....	286
7.10.1	Motivation .....	286
7.10.2	Fehlerkategorien schaffen Klarheit .....	288
7.10.3	Muster zur Fehlerbehandlung .....	290
7.10.4	Mögliche Probleme .....	291
7.10.5	Zusammenhang mit anderen Aspekten .....	292

7.10.6	Weiterführende Literatur	293
7.11	Skalierbarkeit	293
7.11.1	Was bedeutet Skalierbarkeit?	293
7.11.2	Skalierungsstrategien	294
7.11.3	Elastizität	294
7.11.4	Scale-Up-Strategie	294
7.11.5	Vertikale Scale-Out-Strategie	295
7.11.6	Horizontale Scale-Out-Strategie	295
7.11.7	Der Strategiemix	295
7.11.8	Allgemeine Daumenregeln	296
7.11.9	CPU-Power	297
7.11.10	GPU-Power	297
7.11.11	RAIDs, SANs und andere Speichersysteme	297
7.11.12	Bussysteme für die Speicheranbindung	298
7.11.13	Geringere Bandbreite im Netz	298
7.12	Blockchain und dezentralisierte Architektur	299
7.12.1	Definition und Abgrenzung	299
7.12.2	Technische Grundlagen von Blockchains	301
7.12.3	Smart Contracts	302
7.12.4	Blockchains in nichtöffentlichen Kontexten	302
7.12.5	Architekturabwägungen	304
7.12.6	Architekturmuster	305
<b>8</b>	<b>Analyse und Bewertung von Softwarearchitekturen</b>	<b>307</b>
8.1	Qualitative Architekturbewertung	310
8.2	Quantitative Bewertung durch Metriken	317
8.3	Werkzeuge zur Bewertung	319
8.4	Weiterführende Literatur	320
<b>9</b>	<b>Systematische Verbesserung und Evolution</b>	<b>321</b>
9.1	Wege in den Abgrund	323
9.2	Systematisch verbessern	324
9.3	Bewährte Praktiken und Muster	327
9.4	Analyse: Probleme identifizieren	329
9.5	Evaluate: Probleme und Maßnahmen bewerten	331
9.6	Improve: Verbesserungsmaßnahmen planen und durchführen	332
9.6.1	Maxime für Verbesserungsprojekte	332
9.6.2	Kategorien von Verbesserungsmaßnahmen	332
9.7	Crosscutting: phasenübergreifende Praktiken	336
9.8	Mehr zu AIM <sup>42</sup>	337
9.9	Weiterführende Literatur	337

<b>10</b>	<b>Microservices</b> .....	<b>339</b>
10.1	Was sind Microservices? .....	340
10.2	Warum Microservices? .....	340
10.3	Eigenschaften von Microservices .....	341
10.4	Microservices und die Organisation .....	343
10.5	Für welche Systeme eignen sich Microservices? .....	344
10.6	Herausforderungen bei Microservices .....	344
10.6.1	Überblick über viele Services behalten .....	345
10.6.2	Microservices effektiv entwickeln .....	345
10.6.3	Service Discovery .....	346
10.6.4	UI-Integration .....	347
10.6.5	Dezentralisierte Daten .....	347
10.6.6	Versionierung von Microservices .....	348
10.6.7	Laufzeitumgebungen und Infrastruktur verwalten .....	349
10.7	Beispiele für Microservices .....	349
10.8	Weiterführende Literatur .....	349
<b>11</b>	<b>Enterprise-IT-Architektur</b> .....	<b>351</b>
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
<b>12</b>	<b>Beispiele von Softwarearchitekturen</b> .....	<b>361</b>
12.1	Beispiel: Datenmigration im Finanzwesen .....	362
12.2	Beispiel: Kampagnenmanagement im CRM .....	379
<b>13</b>	<b>Werkzeuge für Softwarearchitekten</b> .....	<b>409</b>
13.1	Kategorien von Werkzeugen .....	409
13.2	Typische Auswahlkriterien .....	413
<b>14</b>	<b>iSAQB Curriculum</b> .....	<b>415</b>
14.1	Standardisierte Lehrpläne für Softwarearchitekten .....	416
14.1.1	Grundlagenausbildung und Zertifizierung Foundation-Level .....	416
14.1.2	Fortgeschrittene Aus- und Weiterbildung (Advanced-Level) .....	417
14.2	iSAQB-Foundation-Level-Lehrplan .....	418
14.2.1	Können, Wissen und Verstehen .....	418
14.2.2	Voraussetzungen und Abgrenzungen .....	419
14.2.3	Lernziele des iSAQB Foundation Level .....	420
14.2.4	Zertifizierung gemäß iSAQB .....	422
14.3	Beispielfragen zur Foundation Level Prüfung .....	423

<b>15</b>	<b>Nachwort: Architektonien</b> .....	<b>429</b>
15.1	In sechs Stationen um die (IT-)Welt .....	429
15.2	Ratschläge aus dem architektonischen Manifest .....	432
<b>16</b>	<b>Literatur</b> .....	<b>437</b>
	<b>Stichwortverzeichnis</b> .....	<b>441</b>