

HANSER



Inhaltsverzeichnis

zu

„Effektive Softwarearchitekturen“ (6. Auflage)

von Gernot Starke

ISBN (Buch): 978-3-446-43614-5

ISBN (E-Book): 978-3-446-43653-4

Weitere Informationen und Bestellungen unter
<http://www.hanser-fachbuch.de/978-3-446-43614-5>

sowie im Buchhandel

© Carl Hanser Verlag München

Inhalt

Vorwort	XIII
Vorwort zur sechsten 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	11
1.6 Weiterführende Literatur	12
1.7 Danksagung	12
2 Architektur und Architekten	13
2.1 Was ist Softwarearchitektur?	14
2.1.1 Warum Softwarearchitektur?	18
2.2 Die Aufgaben von Softwarearchitekten	19
2.3 Wie entstehen Architekturen?	24
2.4 In welchem Kontext steht Architektur?	27
2.5 Weiterführende Literatur	30
3 Vorgehen bei der Architekturentwicklung	31
3.1 Informationen sammeln	35
3.2 Anforderungen klären	36
3.2.1 Was ist die Kernaufgabe des Systems?	36
3.2.2 Welche Kategorie von System?	37
3.2.3 Wesentliche Qualitätsanforderungen ermitteln	37
3.2.4 Relevante Stakeholder ermitteln	42
3.2.5 Fachlichen und technischen Kontext ermitteln	43
3.3 Einflussfaktoren und Randbedingungen ermitteln	45
3.4 Entwerfen und kommunizieren	51
3.5 Umsetzung begleiten	52
3.6 Lösungsstrategien entwickeln	53
3.6.1 Strategien gegen organisatorische Risiken	53
3.6.2 Strategien für hohe Performance	55

3.6.3	Strategien für Anpassbarkeit und Flexibilität	56
3.6.4	Strategien für hohe Verfügbarkeit	59
3.7	Wartung und Änderung von Software	60
3.8	Weiterführende Literatur	61
4	Kommunikation und Dokumentation von Architekturen	63
4.1	Architekten müssen kommunizieren und dokumentieren	64
4.2	Effektive Architekturdokumentation	66
4.2.1	Anforderungen an Architekturdokumentation	66
4.2.2	Regeln für gute Architekturdokumentation	69
4.3	Typische Architekturdokumente	72
4.3.1	Zentrale Architekturbeschreibung	73
4.3.2	Architekturüberblick	76
4.3.3	Dokumentationsübersicht	76
4.3.4	Übersichtspräsentation der Architektur	76
4.3.5	Architekturtapete	77
4.4	Sichten	77
4.4.1	Sichten in der Softwarearchitektur	78
4.4.2	Vier Arten von Sichten	80
4.4.3	Entwurf der Sichten	82
4.5	Kontextabgrenzung	84
4.5.1	Elemente der Kontextabgrenzung	84
4.5.2	Notation der Kontextabgrenzung	85
4.5.3	Entwurf der Kontextabgrenzung	85
4.6	Bausteinsicht	86
4.6.1	Elemente der Bausteinsicht	90
4.6.2	Notation der Bausteinsicht	91
4.6.3	Entwurf der Bausteinsicht	92
4.7	Laufzeitsicht	93
4.7.1	Elemente der Laufzeitsicht	94
4.7.2	Notation der Laufzeitsicht	95
4.7.3	Entwurf der Laufzeitsicht	96
4.8	Verteilungssicht	96
4.8.1	Elemente der Verteilungssicht	97
4.8.2	Notation der Verteilungssicht	97
4.8.3	Entwurf der Verteilungssicht	98
4.9	Dokumentation von Schnittstellen	99
4.10	Dokumentation technischer Konzepte	102
4.11	Werkzeuge zur Dokumentation	102
4.12	TOGAF zur Architekturdokumentation	104
4.13	Weiterführende Literatur	105

5	Modellierung für Softwarearchitekten	107
5.1	Modelle als Arbeitsmittel	107
5.1.1	Grafische oder textuelle Modellierung	109
5.2	UML 2 für Softwarearchitekten	110
5.2.1	Die Diagrammart der UML 2	111
5.2.2	Die Bausteine von Architekturen	113
5.2.3	Schnittstellen	114
5.2.4	Die Bausteinsicht	115
5.2.5	Die Verteilungssicht	117
5.2.6	Die Laufzeitsicht	119
5.2.7	Darum UML	123
5.2.8	Darum nicht UML	123
5.3	Tipps zur Modellierung	124
5.4	Weiterführende Literatur	124
6	Strukturentwurf, Architektur- und Designmuster	125
6.1	Von der Idee zur Struktur	127
6.1.1	Komplexität beherrschen	127
6.1.2	Zerlegen – aber wie?	128
6.1.3	Fachmodelle als Basis der Entwürfe	129
6.1.4	Die Fachdomäne strukturieren	132
6.2	Architekturstile und -muster	133
6.2.1	Datenfluss-Architekturstil	135
6.2.1.1	Architekturstil „Batch-sequenziell“	135
6.2.1.2	Architekturstil Pipes und Filter	136
6.2.2	Datenzentrierter Architekturstil	138
6.2.2.1	Repository	138
6.2.2.2	Blackboard	139
6.2.3	Hierarchische Architekturstile	140
6.2.3.1	Master-Slave	141
6.2.3.2	Schichten (Layer)	142
6.2.3.3	Architekturstil „Ports und Adapter“	145
6.2.4	Architekturstile verteilter Systeme	148
6.2.4.1	Client-Server	148
6.2.4.2	Command-Query-Responsibility-Segregation	149
6.2.4.3	Broker	151
6.2.4.4	Peer-to-Peer	152
6.2.5	Ereignisbasierte Systeme – Event Systems	153
6.2.5.1	Ungepufferte Event-Kommunikation	153
6.2.5.2	Message- oder Event-Queue-Architekturen	154
6.2.5.3	Message-Service-Architekturen	155
6.2.6	Interaktionsorientierte Systeme	156
6.2.6.1	Model-View-Controller	156
6.2.6.2	Presentation-Model	157

6.2.7	Weitere Architekturstile und -muster	160
6.3	Heuristiken zum Entwurf	162
6.3.1	Das So-einfach-wie-möglich-Prinzip	162
6.3.2	Entwerfen Sie nach Verantwortlichkeiten	164
6.3.3	Konzentrieren Sie sich auf Schnittstellen	165
6.3.4	Berücksichtigen Sie Fehler	165
6.4	Optimieren von Abhängigkeiten	166
6.4.1	Streben Sie nach loser Kopplung	168
6.4.2	Hohe Kohäsion	169
6.4.3	Offen für Erweiterungen, geschlossen für Änderungen	169
6.4.4	Abhängigkeit nur von Abstraktionen	171
6.4.5	Abtrennung von Schnittstellen	172
6.4.6	Zyklische Abhängigkeiten vermeiden	174
6.4.7	Liskov-Substitutionsprinzip (LSP)	175
6.4.8	Dependency Injection (DI)	176
6.5	Entwurfsmuster	178
6.5.1	Entwurf mit Mustern	178
6.5.2	Adapter	179
6.5.3	Beobachter (Observer)	179
6.5.4	Dekorierer (Decorator)	181
6.5.5	Stellvertreter (Proxy)	181
6.5.6	Fassade	182
6.5.7	Zustand (State)	183
6.6	Entwurf, Test, Qualitätssicherung	184
6.7	Weiterführende Literatur	185
7	Technische Konzepte und typische Architektur Aspekte	187
7.1	Persistenz	191
7.1.1	Motivation	191
7.1.2	Einflussfaktoren und Entscheidungskriterien	194
7.1.2.1	Art der zu speichernden Daten	195
7.1.2.2	Konsistenz und Verfügbarkeit (ACID, BASE oder CAP)	196
7.1.2.3	Zugriff und Navigation	198
7.1.2.4	Deployment und Betrieb	198
7.1.3	Lösungsmuster	199
7.1.3.1	Persistenzschicht	199
7.1.3.2	DAO: Eine Miniatur-Persistenzschicht	203
7.1.4	Bekannte Risiken und Probleme	204
7.1.5	Weitere Themen zu Persistenz	205
7.1.6	Zusammenhang zu anderen Aspekten	209
7.1.7	Praktische Vertiefung	210
7.1.8	Weiterführende Literatur	211
7.2	Geschäftsregeln	212
7.2.1	Motivation	212

7.2.2	Funktionsweise von Regelmaschinen.	215
7.2.3	Kriterien pro & kontra Regelmaschinen.	217
7.2.4	Mögliche Probleme.	218
7.2.5	Weiterführende Literatur.	219
7.3	Integration	219
7.3.1	Motivation	219
7.3.2	Typische Probleme	221
7.3.3	Lösungskonzepte	222
7.3.4	Entwurfsmuster zur Integration	226
7.3.5	Konsequenzen und Risiken	227
7.3.6	Zusammenhang mit anderen Aspekten	229
7.3.7	Weiterführende Literatur.	230
7.4	Verteilung	230
7.4.1	Motivation	230
7.4.2	Typische Probleme	231
7.4.3	Lösungskonzept	231
7.4.4	Konsequenzen und Risiken	233
7.4.5	Zusammenhang mit anderen Aspekten	233
7.4.6	Weiterführende Literatur.	233
7.5	Kommunikation.	234
7.5.1	Motivation	234
7.5.2	Entscheidungsalternativen	234
7.5.3	Grundbegriffe der Kommunikation	234
7.5.4	Weiterführende Literatur.	238
7.6	Grafische Oberflächen(GUI).	240
7.6.1	Motivation	240
7.6.2	Einflussfaktoren und Entscheidungskriterien	240
7.6.3	GUI-relevante Architekturmuster	242
7.6.4	Struktur und Ergonomie von Benutzeroberflächen	243
7.6.5	Bekannte Risiken und Probleme.	244
7.6.6	Zusammenhang zu anderen Aspekten.	246
7.7	Workflow-Management: Ablaufsteuerung im Großen.	247
7.7.1	Zweck der Ablaufsteuerung.	249
7.7.2	Lösungsansätze.	251
7.7.3	Integration von Workflow-Systemen	254
7.7.4	Mächtigkeit von WMS	254
7.7.5	Weiterführende Literatur.	255
7.8	Sicherheit.	255
7.8.1	Motivation – Was ist IT-Sicherheit?.	255
7.8.2	Sicherheitsziele.	256
7.8.3	Lösungskonzepte	258
7.8.4	Security Engineering mit Patterns	265
7.8.5	Weiterführende Literatur.	266
7.9	Protokollierung	267

7.9.1	Typische Probleme	267
7.9.2	Lösungskonzept	268
7.9.3	Zusammenhang mit anderen Aspekten	269
7.9.4	Weiterführende Literatur	269
7.10	Ausnahme- und Fehlerbehandlung	270
7.10.1	Motivation	270
7.10.2	Fehlerkategorien schaffen Klarheit	272
7.10.3	Muster zur Fehlerbehandlung	274
7.10.4	Mögliche Probleme	275
7.10.5	Zusammenhang mit anderen Aspekten	276
7.10.6	Weiterführende Literatur	277
8	Bewertung von Softwarearchitekturen	279
8.1	Qualitative Architekturbewertung	282
8.2	Quantitative Bewertung durch Metriken	289
8.3	Werkzeuge zur Bewertung	291
8.4	Weiterführende Literatur	292
9	Service-orientierte Architektur (SOA)	293
9.1	Was ist SOA?	294
9.2	So funktionieren Services	299
9.3	Was gehört (noch) zu SOA?	300
9.4	SOA und Softwarearchitektur	303
9.5	Weiterführende Literatur	303
10	Enterprise-IT-Architektur	305
10.1	Wozu Architekturebenen?	306
10.2	Aufgaben von Enterprise-Architekten	307
10.2.1	Management der Infrastrukturkosten	307
10.2.2	Management des IS-Portfolios	308
10.2.3	Definition von Referenzarchitekturen	309
10.2.4	Weitere Aufgaben	311
10.3	Weiterführende Literatur	313
11	Beispiele von Softwarearchitekturen	315
11.1	Beispiel: Datenmigration im Finanzwesen	316
11.2	Beispiel: Kampagnenmanagement im CRM	333
12	Werkzeuge für Softwarearchitekten	365
12.1	Kategorien von Werkzeugen	365
12.2	Typische Auswahlkriterien	368

13	iSAQB Curriculum	371
13.1	Standardisierte Lehrpläne für Softwarearchitekten	372
	13.1.1 Grundlagenausbildung und Zertifizierung <i>Foundation-Level</i>	372
	13.1.2 Fortgeschrittene Aus- und Weiterbildung (<i>Advanced-Level</i>)	373
13.2	Können, Wissen und Verstehen	374
13.3	Voraussetzungen und Abgrenzungen	374
13.4	Struktur des iSAQB-Foundation-Level-Lehrplans	375
13.5	Zertifizierung nach dem iSAQB-Lehrplan	378
14	Nachwort: Architektonien	379
14.1	In sechs Stationen um die (IT-)Welt	379
14.2	Ratschläge aus dem architektonischen Manifest	382
15	Literatur	387
	Stichwortverzeichnis	391