

Effektive Softwarearchitekturen

Ein praktischer Leitfaden

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhalt

1	Einleitung	1
1.1	Softwarearchitekt:innen	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	11
1.7	Danksagung	12
2	Softwarearchitektur: Grundlagen und Aufgaben	13
2.1	Was ist Softwarearchitektur?	14
2.1.1	System	14
2.1.2	Komponenten	15
2.1.3	Beziehungen	15
2.1.4	Umgebung	16
2.1.5	Komponenten + Beziehungen = Strukturen	16
2.1.6	Prinzipien (synonym: Konzepte)	17
2.1.7	Entwurf und Evolution	18
2.2	Architekturentscheidungen	19
2.3	Die Aufgaben von Softwarearchitekt:innen	22
2.3.1	Anforderungen klären	23
2.3.2	Drei Kategorien von Entwurfsentscheidungen	24
2.3.3	Architektur kommunizieren und dokumentieren	24
2.3.4	Umsetzung begleiten: Von Goldstücken und Missverständnissen	25
2.3.5	Architektur analysieren und bewerten	25
2.4	Rolle von Softwarearchitekt:innen: Wer macht's?	26
2.4.1	Monarchie	27
2.4.2	Architekt:in im Team	28
2.4.3	Architekturagent:innen	29
2.4.4	Demokratie oder: Team-Architektur	31
2.5	Architekturen entstehen (meist) iterativ	32
2.6	Weiterführende Literatur	34

3	Anforderungen klären	35
3.1	Was ist Kernaufgabe oder Ziel des Systems?	35
3.2	Relevante Stakeholder ermitteln	36
3.3	Welche Kategorie von System?	37
3.4	Fachdomäne klären	38
3.5	Qualitätsanforderungen klären	39
3.6	Externe Nachbarsysteme: Kontextabgrenzung	44
3.7	Einflussfaktoren und Randbedingungen ermitteln	45
4	Entwurf: Grundlagen, Methoden und Muster	47
4.1	Grundlagen, Prinzipien und Heuristiken	48
4.1.1	Grundlagen des Entwurfs	48
4.1.2	Prinzipien	51
4.1.2.1	Lose (geringe) Kopplung	51
4.1.2.2	Hohe Kohäsion	53
4.1.2.3	Trenne Verantwortlichkeiten/Belange	54
4.1.2.4	Modularisierung	55
4.1.2.5	Abstraktion, Kapselung und das Geheimnisprinzip	55
4.1.2.6	Hohe Konsistenz	55
4.1.2.7	Keine zyklischen Abhängigkeiten	56
4.1.2.8	SOLID-Prinzipien des objektorientierten Entwurfs	57
4.1.3	Heuristiken	62
4.2	Entwurfsmethoden	66
4.2.1	Domain-Driven Design (Entwurf nach Fachlichkeit)	67
4.2.2	Quality-Driven Software Architecture	72
4.2.3	Top-down und Bottom-up	79
4.2.4	Sichtenbasierter Entwurf	80
4.2.4.1	Sichten in der Softwarearchitektur	81
4.2.4.2	Entwurf der Kontextabgrenzung	82
4.2.4.3	Entwurf der Bausteinsicht	83
4.2.4.4	Entwurf der Laufzeitsicht	83
4.2.4.5	Entwurf der Verteilungssicht	84
4.3	Schnittstellen entwerfen	84
4.3.1	Anforderungen an Schnittstellen	86
4.3.2	Worauf müssen Sie achten?	86
4.3.3	Tipps zum Entwurf von Schnittstellen	87
4.4	Architekturmuster (Patterns)	88
4.4.1	Schichten (Layer)	88
4.4.2	Ports-und-Adapter	91
4.4.3	Client-Server	94
4.4.4	Microservices	94
4.4.5	Pipes und Filter	102
4.4.6	Batch-Pattern	104

4.4.7	Repository	105
4.4.8	Blackboard.....	106
4.4.9	Command Query Responsibility Segregation (CQRS).....	107
4.4.10	Broker	109
4.4.11	Peer-to-Peer	110
4.4.12	Ereignisbasierte Systeme – Event Systems	111
4.4.12.1	Ungepufferte Event Systems	112
4.4.12.2	Message- oder Event-Queues	112
4.4.12.3	Message-Service	113
4.4.13	Model-View-Controller	114
4.4.14	Presentation Model.....	115
4.4.15	REST-Architektur	117
4.4.16	Adapter	119
4.4.17	Stellvertreter (Proxy)	120
4.4.18	Fassade	121
4.4.19	Beobachter (Observer)	122
4.5	Weiterführende Literatur	123
5	Architekturen kommunizieren, dokumentieren und modellieren .	125
5.1	Warum kommunizieren und dokumentieren.....	126
5.2	Anforderungen an Architekturdokumentation	128
5.3	Effektiv dokumentieren	129
5.3.1	Tipps für bessere Architekturdigramme	131
5.4	Bestandteile von Architekturdokumentation.....	138
5.4.1	Kontextabgrenzung: Vogelperspektive.....	138
5.4.2	Bausteinsicht: Code-im-Großen.....	140
5.4.3	Schnittstellen: Die Brücken zwischen Welten	143
5.4.4	Laufzeitsicht: Was geschieht wann?	144
5.4.5	Verteilungssicht: Zusammenhang zur technischen Infrastruktur	145
5.4.6	Querschnittliche Konzepte.....	146
5.4.7	Entscheidungen	148
5.5	Architekturdokumentation mit arc42.....	150
5.5.1	Aufbau von arc42	150
5.5.2	arc42 Canvas: Dokumentation kompakt	152
5.6	Notationen zur Modellierung: UML, C4 und andere.....	154
5.6.1	UML Kurzeinführung.....	155
5.6.2	C4 Kurzeinführung.....	159
5.6.3	Weitere Notationen für Softwarearchitektur	164
5.7	Werkzeuge zur Dokumentation.....	165
5.8	Weiterführende Literatur	168
6	Analyse und Bewertung von Softwarearchitekturen.....	169
6.1	Qualitative Architekturbewertung	172

6.2	Quantitative Bewertung durch Metriken	179
6.3	Werkzeuge zur Bewertung	181
7	Technische und querschnittliche Konzepte	183
7.1	Persistenz	186
7.1.1	Motivation	186
7.1.2	Einflussfaktoren und Entscheidungskriterien	189
7.1.2.1	Art der zu speichernden Daten	190
7.1.2.2	Konsistenz und Verfügbarkeit (ACID, BASE oder CAP)	191
7.1.2.3	Zugriff und Navigation	192
7.1.2.4	Deployment und Betrieb	193
7.1.3	Lösungsmuster	193
7.1.3.1	Persistenzschicht	193
7.1.3.2	DAO: Eine Miniatur-Persistenzschicht	196
7.1.4	Bekannte Risiken und Probleme	197
7.1.5	Weitere Themen zu Persistenz	198
7.1.6	Data Contracts: Daten als Schnittstelle	202
7.1.7	Zusammenhang zu anderen Themen	204
7.1.8	Praktische Vertiefung	206
7.1.9	Weiterführende Literatur	207
7.2	Geschäftsregeln	207
7.2.1	Motivation	207
7.2.2	Funktionsweise von Regelmaschinen	210
7.2.3	Kriterien pro & kontra Regelmaschinen	212
7.2.4	Mögliche Probleme	213
7.2.5	Weiterführende Literatur	214
7.3	Integration	215
7.3.1	Motivation	215
7.3.2	Typische Probleme	217
7.3.3	Lösungskonzepte	218
7.3.4	Entwurfsmuster zur Integration	222
7.3.5	Zusammenhang mit anderen Themen	222
7.3.6	Weiterführende Literatur	223
7.4	Verteilung	224
7.4.1	Motivation	224
7.4.2	Typische Probleme	224
7.4.3	Lösungskonzept	224
7.4.4	Konsequenzen und Risiken	225
7.4.5	Zusammenhang mit anderen Themen	226
7.4.6	Weiterführende Literatur	226
7.5	Kommunikation	226
7.5.1	Motivation	227
7.5.2	Entscheidungsalternativen	227

7.5.3	Grundbegriffe der Kommunikation	227
7.5.4	Weiterführende Literatur	231
7.6	Grafische Oberflächen (GUI)	231
7.6.1	Motivation	231
7.6.2	Einflussfaktoren und Entscheidungskriterien	231
7.6.3	GUI-relevante Architekturmuster	234
7.6.4	Struktur und Ergonomie von Benutzeroberflächen	234
7.6.5	Bekannte Risiken und Probleme	236
7.6.6	Zusammenhang zu anderen Themen	238
7.7	Sicherheit	238
7.7.1	Motivation – Was ist IT-Sicherheit?	238
7.7.2	Sicherheitsziele	239
7.7.3	Lösungskonzepte	241
7.7.4	Security Engineering mit Patterns	248
7.7.5	Weiterführende Literatur	249
7.8	Protokollierung	249
7.8.1	Typische Probleme	250
7.8.2	Lösungskonzept	251
7.8.3	Zusammenhang mit anderen Themen	251
7.9	Ausnahme- und Fehlerbehandlung	252
7.9.1	Motivation	252
7.9.2	Fehlerkategorien schaffen Klarheit	254
7.9.3	Muster zur Fehlerbehandlung	256
7.9.4	Mögliche Probleme	257
7.9.5	Zusammenhang mit anderen Themen	258
7.9.6	Weiterführende Literatur	259
7.10	Skalierbarkeit	259
7.10.1	Skalierungsstrategien	259
7.10.2	Elastizität	260
7.10.3	Scale-Up-Strategie	260
7.10.4	Vertikale Scale-Out-Strategie	260
7.10.5	Horizontale Scale-Out-Strategie	261
7.10.6	Der Strategiemix	261
7.10.7	Allgemeine Daumenregeln	262
7.10.8	CPU-Power	262
7.10.9	GPU-Power	262
7.10.10	RAIDs, SANs und andere Speichersysteme	263
7.10.11	Bussysteme für die Speicheranbindung	263
7.10.12	Geringere Bandbreite im Netz	264
7.11	Container und die Cloud	264
7.11.1	Was bedeutet „Cloud“?	265
7.11.2	Virtuelle Maschinen (VMs) und Container	267
7.11.3	Von Monolithen in die Cloud	269

7.11.4	Was Sie noch über die Cloud wissen sollten.....	272
7.11.5	Weiterführende Literatur.....	274
7.12	Weitere spannende Themen.....	274
8	Systematische Verbesserung und Evolution	277
8.1	Wege in den Abgrund.....	279
8.2	Systematisch verbessern	280
8.3	Bewährte Praktiken und Muster	284
8.4	Analyse: Probleme identifizieren	285
8.5	Evaluate: Probleme und Maßnahmen bewerten	287
8.6	Improve: Verbesserungsmaßnahmen planen und durchführen	288
8.6.1	Maxime für Verbesserungsprojekte	288
8.6.2	Kategorien von Verbesserungsmaßnahmen	288
8.7	Crosscutting: phasenübergreifende Praktiken	292
8.8	Mehr zu aim ⁴²	293
8.9	Weiterführende Literatur	293
9	Beispiele von Softwarearchitekturen	295
9.1	Beispiel: Datenmigration im Finanzwesen.....	296
9.2	Beispiel: Kampagnenmanagement im CRM.....	313
10	iSAQB Curriculum.....	339
10.1	Standardisierte Lehrpläne für Softwarearchitektur	340
10.1.1	Grundlagenausbildung und Zertifizierung Foundation-Level	340
10.1.2	Fortgeschrittene Aus- und Weiterbildung (Advanced-Level)	341
10.2	iSAQB-Foundation-Level-Lehrplan.....	341
10.2.1	Können, Wissen und Verstehen	342
10.2.2	Voraussetzungen und Abgrenzungen.....	342
10.2.3	Struktur des iSAQB-Foundation-Lehrplans	343
10.2.4	Zertifizierung gemäß iSAQB	343
10.3	Beispielfragen zur Foundation-Level-Prüfung	344
	Literatur	349
	Stichwortverzeichnis.....	353