

Vorwort: Axiome infrage stellen	XV
1 Einleitung	1
Softwarearchitektur definieren	3
Erwartungen an Architekten	7
Architekturentscheidungen treffen	8
Kontinuierliche Analyse der Architektur	9
Bei aktuellen Trends auf dem Laufenden bleiben	9
Sicherstellen, dass Entscheidungen eingehalten werden	10
Vielfältige Kenntnisse und Erfahrungen	10
Wissen in der Fachdomäne des Problems	11
Fähigkeiten im zwischenmenschlichen Umgang	11
Politik verstehen und sich in dieser Sphäre bewegen können	12
Überschneidungen von Architektur und	13
Engineering-Praktiken	14
Technischer Betrieb/DevOps	17
Prozess	18
Daten	19
Gesetze der Softwarearchitektur	19

Teil I Grundlagen

2 Architektonisches Denken	23
Architektur und Design im Vergleich	24
Technische Breite	26
Vor- und Nachteile analysieren	30
Geschäftliche Faktoren verstehen	33
Die Balance zwischen Architektur und tatsächlichem Programmieren ..	34

3	Modularität	37
	Definition	38
	Modularität messen	40
	Kohäsion	40
	Kopplung	44
	Abstraktheit, Instabilität und Entfernung von der Hauptsequenz	45
	Entfernung von der Hauptsequenz	46
	Konnasenz	48
	Kopplungs- und Konnasenzmetriken vereinheitlichen	52
	Von Modulen zu Komponenten	54
4	Definition architektonischer Eigenschaften	55
	Architektonische Eigenschaften, eine (unvollständige) Liste	58
	Betriebsrelevante architektonische Eigenschaften	58
	Strukturelle architektonische Eigenschaften	59
	Bereichsübergreifende architektonische Eigenschaften	60
	Kompromisse und am wenigsten schlechte Architektur.	64
5	Architektonische Eigenschaften ermitteln	67
	Architektonische Eigenschaften aus domänenspezifischen Anforderungen ableiten	67
	Architektonische Eigenschaften aus funktionalen Anforderungen ableiten	70
	Fallstudie: Silicon Sandwiches	71
	Explizite Eigenschaften	72
	Implizite Eigenschaften	76
6	Messung und Governance von architektonischen Eigenschaften	79
	Architektonische Eigenschaften messen	79
	Betriebsrelevante Metriken	80
	Strukturelle Metriken	81
	Prozessbasierte Metriken	83
	Governance und Fitnessfunktionen	84
	Governance für architektonische Eigenschaften	84
	Fitnessfunktionen	85
7	Anwendungsbereich architektonischer Eigenschaften	93
	Kopplung und Konnasenz	94
	Architektonische Quanten und Granularität	94
	Fallstudie: Going, Going, Gone (»Zum Ersten, zum Zweiten und zum Dritten«)	97

8	Komponentenbasiertes Denken	103
	Anwendungsbereiche für Komponenten	103
	Die Rolle des Architekten	105
	Architektonische Partitionierung	105
	Fallstudie: Partitionierung für Silicon Sandwiches	109
	Die Rolle des Entwicklers	112
	Arbeitsablauf zur Ermittlung der Komponenten	112
	Anfängliche Komponenten ermitteln	112
	Anforderungen auf Komponenten abbilden	113
	Rollen und Verantwortlichkeiten analysieren	113
	Architektonische Eigenschaften analysieren	113
	Komponenten restrukturieren	114
	Komponentengranularität	114
	Komponentendesign	114
	Sinnvolle Komponentenaufteilung ermitteln	114
	Fallstudie: Going, Going, Gone: Komponenten ermitteln	117
	Rückblick auf das architektonische Quantum: Die Wahl zwischen monolithischen und verteilten Architekturen	120

Teil II Architekturstile

9	Architekturstile	125
	Grundmuster	125
	Big Ball of Mud (Der »große Matschkumpen«)	126
	Eingliedrige Architektur	127
	Client/Server	127
	Monolithische und verteilte Architekturen	129
	Irrtum Nr. 1: Das Netzwerk ist verlässlich	130
	Irrtum Nr. 2: Die Latenz ist gleich null	130
	Irrtum Nr. 3: Die Bandbreite ist unendlich	131
	Irrtum Nr. 4: Das Netzwerk ist sicher	133
	Irrtum Nr. 5: Die Topologie ändert sich nie	133
	Irrtum Nr. 6: Es gibt nur einen Administrator	134
	Irrtum Nr. 7: Die Transportkosten sind gleich null	135
	Irrtum Nr. 8: Das Netzwerk ist homogen	135
	Weitere Überlegungen zum verteilten Rechnen	136
10	Der schichtbasierte Architekturstil	139
	Topologie	139
	Voneinander isolierte Schichten	141
	Schichten hinzufügen	143

Zusätzliche Überlegungen	144
Gründe für den schichtbasierten Architekturstil	145
Bewertung der architektonischen Eigenschaften	146
11 Pipeline-Architekturstil	149
Topologie	149
Pipes	150
Filter	150
Beispiel	151
Bewertung der architektonischen Eigenschaften	152
12 Mikrokernel-Architekturstil	155
Topologie	155
Kernsystem	156
Plug-in-Komponenten	159
Registry	162
Kontrakte	163
Beispiele und Anwendungsfälle	164
Bewertung der architektonischen Eigenschaften	165
13 Servicebasierter Architekturstil	167
Topologie	167
Topologische Varianten	168
Servicedesign und Granularität	171
Datenbank-Partitionierung	173
Beispielarchitektur	176
Bewertung der architektonischen Eigenschaften	177
Wann man diesen Architekturstil verwenden sollte	180
14 Eventbasierter Architekturstil	183
Topologie	184
Broker-Topologie	184
Mediator-Topologie	189
Asynchrone Fähigkeiten	200
Fehlerbehandlung	202
Datenverlust verhindern	205
Broadcast-Fähigkeiten	207
Request-Reply	208
Request- oder eventbasiert?	211
Hybride eventbasierte Architekturen	211
Bewertung der architektonischen Eigenschaften	212

15 »Space-based«-Architekturstil	215
Allgemeine Topologie	216
Verarbeitungseinheit	217
Virtualisierte Middleware	218
Data Pumps	223
Data Writer	225
Data Reader	226
Datenkollisionen	228
Cloudbasierte und On-Premises-Implementierungen im Vergleich	231
Repliziertes Caching im Vergleich mit verteiltem Caching	232
Überlegungen zu Near-Cache	235
Implementierungsbeispiele	236
System zum Verkauf von Veranstaltungstickets	236
Online-Auktionssysteme	237
Bewertung der architektonischen Eigenschaften	237
16 Orchestrierter serviceorientierter Architekturstil (SOA)	241
Geschichte und Philosophie	241
Topologie	242
Taxonomie	242
Dienste für die Geschäftslogik	243
Unternehmensdienste	243
Applikationsdienste	243
Infrastrukturdienste	244
Orchestrierungs-Engine	244
Nachrichtenfluss	244
Wiederverwendbarkeit und Kopplung	245
Bewertung der architektonische Eigenschaften	247
17 Microservices-Architekturstil	251
Geschichte	251
Topologie	252
Verteilt	253
Bounded Context	253
Granularität	254
Datenisolation	255
API-Schicht	255
Betriebliche Wiederverwendung (Operational Reuse)	256
Frontends	258
Kommunikation	260
Choreografie und Orchestrierung	261
Transaktionen und Sagas	265

Bewertung der architektonischen Eigenschaften	267
Weiterführende Referenzen	269
18 Den richtigen Architekturstil auswählen	271
Architekturstile als »Modeerscheinungen«	271
Entscheidungskriterien	273
Monolith-Fallstudie: Silicon Sandwiches	275
Modularer Monolith	275
Microkernel	277
Verteilte Architektur, Fallstudie: Going, Going, Gone	278

Teil III Techniken und Soft Skills

19 Architekturentscheidungen	285
Antipatterns für Architekturentscheidungen	285
Antipattern: Covering your Assets	285
Antipattern: Groundhog Day	286
Antipattern: Email-Driven Architecture	287
Architektonisch wichtig	288
Architecture Decision Records (ADR)	289
Grundstruktur	290
ADRs speichern	296
ADRs als Dokumentation	298
ADRs für Standards verwenden	298
Beispiel	299
20 Architektonische Risiken analysieren	301
Matrix zur Risikobewertung	301
Risikobewertung	302
Risk Storming	305
Identifizierung	307
Konsens	308
Risikoanalyse von Agile Stories	311
Risk-Storming-Beispiele	312
Verfügbarkeit	313
Elastizität	315
Sicherheit	317

21	Architektur in Diagrammen und Präsentationen visualisieren	319
	Diagramme	320
	Werkzeuge	320
	Standards für Diagramme: UML, C4 und ArchiMate	322
	Richtlinien für die Erstellung von Diagrammen	323
	Präsentieren	325
	Zeit manipulieren	326
	Inkrementeller Aufbau	326
	Infodecks im Vergleich mit Präsentationen	329
	Folien sind nur die Hälfte des Vortrags	329
	Unsichtbarkeit	329
22	Effektive Teams schaffen	331
	Teams den richtigen Rahmen vorgeben	331
	Architekten-Persönlichkeiten	332
	Der Kontrollfreak	332
	Der Sofa-Architekt	334
	Der effektive Architekt	336
	Wie viel Kontrolle?	336
	Warnsignale des Teams	341
	Checklisten einsetzen	343
	Entwickler-Checkliste für die Codefertigstellung	346
	Checkliste für Unit- und funktionales Testing	347
	Software-Release-Checkliste	348
	Orientierung bieten	348
	Zusammenfassung	351
23	Verhandlungsgeschick und Führungsqualitäten	353
	Verhandlung und Moderation	353
	Mit geschäftlichen Entscheidungsträgern verhandeln	354
	Mit anderen Architekten verhandeln	356
	Mit Entwicklern verhandeln	357
	Der Softwarearchitekt als Führungskraft	359
	Die vier Ks der Architektur	359
	Seien Sie pragmatisch, aber visionär	361
	Teams mit gutem Beispiel vorangehen	363
	Abstimmung mit dem Entwicklungsteam	367
	Zusammenfassung	370

24 Eine berufliche Laufbahn entwickeln	371
Die 20-Minuten-Regel.	371
Ein persönliches Radar entwickeln.	373
Das ThoughtWorks Technology Radar.	373
Open-Source-Visualisierungen.	377
Soziale Medien verwenden	377
Rat zum Abschied.	378
A Fragen zur Selbstbeurteilung	381
Index	391