

Basiswissen Softwarearchitektur

Verstehen, entwerfen, wiederverwenden

von

Torsten Posch, Klaus Birken, Michael Gerdom

3., akt. u. erw. Aufl.

[Basiswissen Softwarearchitektur – Posch / Birken / Gerdom](#)

schnell und portofrei erhältlich bei [beck-shop.de](#) DIE FACHBUCHHANDLUNG

Thematische Gliederung:

[Software Engineering](#)

dpunkt.verlag 2011

Verlag C.H. Beck im Internet:

[www.beck.de](#)

ISBN 978 3 89864 736 6

Inhaltsverzeichnis

Teil I	Grundlagen und Organisation	1
1	Grundlagen	3
1.1	Warum Softwarearchitektur?	4
1.2	Was ist Softwarearchitektur?	6
1.2.1	Definition von Softwarearchitektur	6
1.2.2	Ziele und Aufgaben von Softwarearchitektur	13
1.2.3	Wodurch wird Softwarearchitektur beeinflusst?	18
1.3	Bedeutung von Softwarearchitektur	20
1.3.1	Symptome bei fehlender Softwarearchitektur	22
1.4	Zusammenfassung	23
2	Softwarearchitektur in der Organisationsstruktur	25
2.1	Wechselwirkungen zwischen Architektur und Unternehmen	25
2.2	Die Rolle des Softwarearchitekten	27
2.2.1	Allgemeine Eigenschaften und Aufgaben	29
2.2.2	Aufgaben im Entwicklungsprojekt	31
2.2.3	Das Architekturteam	39
2.3	Zusammenspiel von Softwarearchitektur und Projektmanagement	43
2.3.1	Bedeutung von Softwarearchitektur für das Projektmanagement	43
2.3.2	Das Führungsteam aus Projektleiter und Softwarearchitekt	50
2.4	Zusammenfassung	53

Teil II Erstellung der Softwarearchitektur	55
<hr/>	
3 Vorgehen	57
3.1 Überblick	58
3.2 Vorbereitungen für den Entwurf	60
3.2.1 Anforderungsanalyse	61
3.2.2 Einflussfaktoren	64
3.3 Iterativ, inkrementeller Entwurf, Dokumentation und Bewertung	65
3.3.1 Der erste Architekturentwurf	66
3.3.2 Iterativ, inkrementelles Ausbauen des Entwurfs	68
3.4 Die Umsetzung der Architektur	69
3.5 Zusammenfassung	70
4 Einflussfaktoren	71
4.1 Bedeutung von Einflussfaktoren	71
4.2 Arten von Einflussfaktoren	74
4.2.1 Organisatorische Faktoren	74
4.2.2 Technologische Faktoren	75
4.2.3 Produktfaktoren	76
4.2.4 Flexibilität, Veränderbarkeit und Einfluss	79
4.3 Spezifikation von Einflussfaktoren	79
4.3.1 Identifizieren und Präzisieren der Faktoren	80
4.3.2 Analyse der Faktoren	84
4.3.3 Identifizieren von Architekturthemen und Entwickeln von Strategien	86
4.4 Zusammenfassung	89
5 Entwurf von Softwarearchitekturen	91
5.1 Entwurfsumfeld und wichtige Begriffe	92
5.1.1 Entwurfsziele	92
5.1.2 Entwurf und Komplexität	94
5.1.3 Vorleistungen	95
5.1.4 Allgemeine Aktivitäten beim Entwurf	97
5.1.5 Fünf Kriterien für einen korrekten Entwurf	99
5.2 Fundamentale Entwurfsprinzipien	102
5.2.1 Abstraktion	103
5.2.2 Kapselung	104
5.2.3 Modularität	105
5.2.4 Hierarchie	106
5.2.5 Konzeptuelle Integrität	107

5.3	Komponenten und Schnittstellen	108
5.3.1	Komponenten – Grundbausteine der Architektur	108
5.3.2	Schnittstellen – Vertragswerk der Softwarearchitektur . . .	112
5.3.3	Techniken zur Adaption von Komponenten	113
5.4	Entwurfsschritte und Heuristiken	114
5.4.1	Konkrete Entwurfsschritte	115
5.4.2	Heuristiken	120
5.5	Zusammenfassung	125
6	Dokumentation	127
6.1	Bedeutung der Dokumentation	128
6.2	Anforderungen an eine Dokumentation	129
6.2.1	Allgemeine Anforderungen an eine Projektdokumentation	130
6.2.2	Anforderungen an Architekturbeschreibungen	132
6.3	Bestandteile einer Architekturdokumentation	134
6.3.1	Sichten eines Systems	134
6.3.2	Zusammenspiel der Sichten	136
6.3.3	Beschreibung des Aufbaus und Hilfestellungen	136
6.3.4	Zusammenfassung	137
6.4	Architektsichten	137
6.4.1	Kontextsicht	139
6.4.2	Struktursicht	140
6.4.3	Verhaltenssicht	141
6.4.4	Abbildungssicht	141
6.4.5	Sichten in der Literatur	142
6.5	UML 2 als Notation für Architektsichten	145
6.5.1	UML-Überblick	145
6.5.2	Darstellungsmöglichkeiten für die Kontextsicht	148
6.5.3	Darstellungsmöglichkeiten für die Struktursicht	153
6.5.4	Darstellungsmöglichkeiten für die Verhaltenssicht	156
6.5.5	Darstellungsmöglichkeiten für die Abbildungssicht	166
6.5.6	Beschreibungsmöglichkeiten für weitere Architekturaspekte	168
6.5.7	UML Erweiterungsmechanismen zur Konsistenzsicherung	172
6.6	Zusammenfassung	174

7	Bewertung	175
7.1	Grundlagen der Architekturbewertung	175
7.1.1	Allgemeines Vorgehen und Ergebnis	177
7.1.2	Arten von Bewertungen und Zeitpunkt	178
7.1.3	Der Faktor Erfahrung	180
7.2	Bewertungsmethoden	181
7.2.1	Fragetechniken	182
7.2.2	Messtechniken	183
7.2.3	Auf Erfahrung basierende Argumentation	184
7.2.4	Kategorisierung der Bewertungsmethoden	184
7.3	Szenariobasierte Bewertung	186
7.3.1	ATAM	186
7.3.2	ATAM-Phasen	187
7.3.3	ATAM-Schritte	192
7.4	Kosten und Nutzen	197
7.4.1	Kosten	198
7.4.2	Nutzen	199
7.5	Zusammenfassung	201
8	Die Toolbox des Softwarearchitekten	203
8.1	Einführung	203
8.1.1	Historie und derzeitiger Stand	204
8.1.2	Vorteile und Aufbau unserer Toolbox	205
8.1.3	Wie erwirbt der Architekt sein Wissen?	207
8.2	Lösungsvorlagen und Methoden	207
8.2.1	Anwendung von Architekturstilen	208
8.2.2	Anwendung von Architekturmustern	213
8.2.3	Anwendung von Entwurfsmustern	219
8.3	Technologien und Werkzeuge	227
8.3.1	Betriebssysteme und Programmiersprachen	228
8.3.2	Bibliotheken, Komponenten und Frameworks	229
8.3.3	Modellierung und Generierung	232
8.3.4	Analyse und Rekonstruktion	234
8.4	Zusammenfassung	236

9	Fallbeispiel	239
9.1	Projektbeschreibung	239
9.2	Schrittweises Vorgehen zur Erstellung der Architektur	240
9.3	Ausgangssituation	242
9.4	Anforderungen und Use Cases	243
9.5	Analysemodell	245
9.6	Aufbau der Architekturendokumentation	246
9.7	Architekterstellung	247
9.7.1	Spezifikation der Einflussfaktoren	247
9.7.2	Entwurf und Dokumentation	253
9.7.3	Umfangreiches Assessment	260
9.8	Umsetzung der Architektur	262
9.9	Zusammenfassung	263
Teil III	Industrielle Softwareentwicklung	265
10	Softwarearchitektur im industriellen Maßstab	267
10.1	Chronische Probleme der heutigen Softwareentwicklung	268
10.1.1	Unnötige Freiheitsgrade bei Sprachen und Tools	268
10.1.2	Schwerpunkt auf Einzelprojekten	270
10.1.3	Ungenügendes Zusammenspiel von Komponenten	270
10.2	Bahnbrechende Innovationen	273
10.2.1	Innovationsfeld 1: Systematische Wiederverwendung	273
10.2.2	Innovationsfeld 2: Modellgetriebene Entwicklung	274
10.3	Komplexität und die Abstraktionslücke	276
10.3.1	Arten von Komplexität in der Softwareentwicklung	276
10.3.2	Die Abstraktionslücke	277
10.3.3	Verkleinern der Abstraktionslücke	278
10.4	Zusammenfassung	279
11	Produktlinien für Software	281
11.1	Was sind Produktlinien?	281
11.1.1	Vom Softwaresystem zur Standardplattform	282
11.1.2	Grundlegende Begriffe	284
11.1.3	Wann sind Softwareproduktlinien sinnvoll?	287
11.1.4	Softwareproduktlinien in drei Dimensionen	293
11.1.5	Wiederverwendung als treibende Kraft	294

11.2	Aktivitäten und Vorgehen	297
11.2.1	Wesentliche Aktivitäten zum Betrieb einer Produktlinie	297
11.2.2	Tätigkeiten des Softwarearchitekten	299
11.2.3	Allgemeine Schritte zum Produktlinienentwurf	301
11.2.4	Softwarebezogene Schritte zur Einführung	302
11.3	Architektur und Software Engineering	303
11.3.1	Aufgaben für Architekt und Softwareingenieur	303
11.3.2	Komponenten – Grundbausteine der Produktlinie	307
11.3.3	Objektorientierte Frameworks	309
11.4	Technische und organisatorische Aufgaben	312
11.4.1	Technische Aufgaben	312
11.4.2	Organisatorische Aufgaben	314
11.5	Zusammenfassung	314
12	Modellbasierte Entwicklung mit MDA und DSLs	317
12.1	Grundidee von MDA	318
12.2	Konzepte	320
12.2.1	Modelle	320
12.2.2	Transformationen	323
12.3	Metamodellierung	325
12.3.1	Vier-Schichten-Modell der Metamodellierung	326
12.4	Fallstudie	328
12.4.1	Plattformunabhängiges Modell (PIM)	328
12.4.2	Technische Lösung und Markierungen	329
12.4.3	Transformation PIM zu PSM	330
12.4.4	Plattformabhängiges Modell (PSM)	331
12.4.5	Transformation PSM zu Code	332
12.5	Alternative: Domänenspezifische Sprachen	334
12.5.1	Grundidee der domänenspezifischen Sprachen	334
12.5.2	Werkzeugunterstützung für DSLs	336
12.5.3	Anwendungsfelder	338
12.6	Zusammenfassung	339
Anhang		341
Literatur		343
Index		349