

Ralf Reussner • Wilhelm Hasselbring (Hrsg.)

# Handbuch der Software-Architektur

*m* HOCHSCHULE  
*m* LIECHTENSTEIN  
Bibliothek

Pin dpunkt.verlag

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Die Rolle der Software-Architekten</b>	<b>9</b>
2.1	Wer wird als Software-Architekt angesehen?	9
2.1.1	Der Plattformspezialist als Software-Architekt	10
2.1.2	Der Entwurfsspezialist als Software-Architekt	11
2.1.3	Der Strategie als Software-Architekt	11
2.2	Probleme der technologiebezogenen Sichtweisen	12
2.2.1	Der Software-Architekt als Wunderheiler	13
2.2.2	Der Software-Architekt als kleiner König	13
2.2.3	Resultierende Konflikte	14
2.2.4	Die generische Flexibilisierungsfalle	15
2.3	Aufgaben von Architekten	16
2.3.1	Entwurf	17
2.3.2	Planung und Organisation	18
2.3.3	Bauüberwachung	19
2.4	Wer sollte Architekten beauftragen?	19
2.4.1	Die Bauherrenrolle	20
2.4.2	Entwurfsentscheide muss der Bauherr fällen	21
2.4.3	Qualitätssicherung braucht der Bauherr/:	21
2.5	Architektur studieren	22
2.5.1	Ausbildungsziele für ein Studium	23
2.5.2	Architektur eines Architektur-Studiengangs	24
2.6	Fazit	28
<b>I</b>	<b>Konstruktion von Architekturen</b>	<b>31</b>
<b>3</b>	<b>Architekturbeschreibung</b>	<b>35</b>
3.1	Standpunkte und Sichten	35
3.1.1	DerANSI/IEEE-Standard 1471-2000	35
3.1.2	Vorgehensweise bei der Wahl der Standpunkte	37
3.1.3	Vergleich verschiedener Standpunktmengen	37
3.2	Architekturbeschreibung mit der UML	40

## Inhaltsverzeichnis

3.2.1	Von der UML 1 zur UML 2 . . . . .	40
3.2.2	Verschiedene Standpunkte . . . . .	41
3.2.3	Relevante Diagrammtypen . . . . .	42
3.2.4	Spezifikation nichtfunktionaler Eigenschaften . . . . .	58
3.3	Das Metamodell der UML 2 für Architekturelemente . . . . .	60
<b>4</b>	<b>Prozess der Architektur- und Komponentenentwicklung . . . . .</b>	<b>65</b>
4.1	Charakter des Prozesses . . . . .	65
4.2	Aktivitäten innerhalb einer Iteration . . . . .	66
4.3	Entwurfsprinzipien . . . . .	72
4.3.1	Abstraktion . . . . .	72
4.3.2	Modularisierung . . . . .	73
4.3.3	Kapselung . . . . .	73
4.3.4	Hierarchische Dekomposition . . . . .	74
4.3.5	Separation of Concerns . . . . .	75
4.3.6	Einheitlichkeit . . . . .	76
4.4	Entwicklungstätigkeiten . . . . .	77
4.5	Entwurfsentscheidungen . . . . .	81
4.6	Arbeitsmittelvorrat des Entwicklers . . . . .	86
<b>5</b>	<b>Identifikation von Komponenten . . . . .</b>	<b>89</b>
5.1	Beispiel: Schafkopfen mit dem Computer . . . . .	90
5.2	Software-Kategorien und Komplexität . . . . .	92
5.3	Was gehört zu O-Software? . . . . .	94
5.4	Kommunikation zwischen Komponenten . . . . .	95
5.5	Anwendung und Technik . . . . .	96
5.5.1	Anwendungs-Software (A-Software) . . . . .	97
5.5.2	Technik-Software (T-Software) . . . . .	97
5.5.3	Kombination von Anwendung und Technik . . . . .	98
5.6	Trennung von Anwendung und Technik . . . . .	99
5.7	Regeln . . . . .	101
<b>6</b>	<b>Model-Driven Architecture . . . . .</b>	<b>103</b>
6.1	Begriffserklärungen . . . . .	104
6.1.1	Modell . . . . .	104
6.1.2	Metamodell . . . . .	106
6.1.3	Modelltransformation . . . . .	107
6.1.4	MDA-Marks . . . . .	107
6.1.5	Plattform . . . . .	109
6.1.6	MDA und die Begriffe CIM, PIM, PSM . . . . .	109
6.1.7	MDA-Konfiguration . . . . .	111
6.2	Vereinfachte Architekturevolution durch Abstraktion . . . . .	112
6.3	Modelltransformationen und Verfeinerung . . . . .	113

6.3.1	Klassifikation verschiedener Transformationsarten .....	114
6.3.2	Nachverfolgbarkeit .....	116
6.3.3	Iterative Ausführung.....	117
6.3.4	Inkrementelle Ausführung.....	117
6.3.5	Marks und Verfeinerung.....	118
6.4	Modellgetriebene Entwicklung in Teams.....	119
6.5	Die Rolle von Standards.....; ;, .....	122
6.6	Entwurf von MDA-Konfigurationen.....	124
6.6.1	Merkmalanalyse und domänenspezifische Sprachen.....	126
6.6.2	Zuschnitt von Modelltransformatibnen.....	126
6.6.3	UML-Profile.....	127

**II Evolution von Architekturen** **129**

<b>7</b>	<b>Grundlagen der Evolution von Software-Architekturen .....</b>	<b>133</b>
7.1	Grundlegende Motivation zur Evolution.....	134
7.1.1	Methode.....	135
7.1.2	Anwendbarkeit der Evolution.....	137
7.1.3	Domänenspezifische Sprachen ...*.....•...	138
7.2	Ansätze und Konzepte zur Software-Evolution.....	139
7.3	Refactoring: Evolution im Kleinen.....	140
7.4	Software-Evolution am Beispiel.....	143
7.4.1	Validierung von Refactorings durch Testen.....	147
7.5	Fazit.....	148
<b>8</b>	<b>Reverse Engineering .....</b>	<b>151</b>
8.1	Reverse Engineering ist zielgetrieben.....	152
8.2	Wissensbasiertes Reverse Engineering von Architekturen.....	153
8.3	Reverse Engineering.....	154
8.4	Abhängigkeitsbasiertes Reverse Engineering.....	155
8.4.1	Die Rolle von Metriken im Reverse-Engineering-Prozess.....	156
8.4.2	Programmabhängigkeitsgraphen.....	157
8.4.3	Programmschneiden .....	157
8.5	Benutzereinbindung in den Reverse-Engineering-Prozess.....	158
8.5.1	Automatisches Reverse Engineering.....; *.....: .....	158
8.5.2	Halbautomatisches Reverse Engineering.....	160
8.6	Reverse Engineering als Teil des agilen Entwicklungsprozesses.....	161
8.7	Industrielle Reife und Einsetzbarkeit .....	162
<b>9</b>	<b>Migration von Altsystemem .....</b>	<b>165</b>
9.1	Projektkontext ...;.....	166
9.2	Technische Aspekte sanfter Migration.....••	167
9.2.1	Multi-Schichten-Architekturen für betriebliche Informationssysteme	168

## xii Inhaltsverzeichnis

9.2.2	Architektur und Migrationsprozess.....*	168
9.2.3	Das Dublo-Muster: DUal Business LOgic.....	172
9.3	Dienstorientierte Zielarchitektur.....*	176
9.4	Nichttechnische Aspekte sanfter Migration.....	178
9.4.1	Betriebswirtschaftliche Aspekte sanfter Migration.....	178
9.4.2	Organisatorische Aspekte sanfter Migration.....	181
9.4.3	Psychologische Aspekte sanfter Migration.....	182

## III Management von Architekturen 183

<b>10</b>	<b>Integratives IT-Architekturmanagement.....</b>	<b>187</b>
10.1	Situation im Unternehmen.....	188
10.2	Modellbasiertes Architekturmanagement.....	189.
10.3	Analyse der Anwendungsdomäne.....	191
10.3.1	Typische Fragen an die konzeptuelle IT-Architektur.....	191
10.3.2	Anwenderrollen einer konzeptuellen IT-Architektur.....	192
10.3.3	Defizitanalyse konzeptueller IT-Architekturen.....	194
10.4	Integratives IT-Architekturmanagement.....	195
10.4.1	Anforderungen an die konzeptuelle IT-Architektur.....	195
10.4.2	Lösungsansätze zur Modellstrategie.....	196
10.4.3	Metadaten und Modellmanagement.....	198
10.5	Vorgehensmodell.....	200
10.5.1	Lebenszyklus einer konzeptuellen IT-Architektur.....	200
10.5.2	Integration in vorhandene Strukturen und Modelle.....	202
10.6	Kriterien zur Werkzeugunterstützung.....	204
10.6.1	Verzahnung des Repositoriums mit der Umwelt.....	204
10.6.2	Unterstützung des Lebenszyklus.....	206
10.6.3	Weitere technische Aspekte.....	208
<b>11</b>	<b>Organisation des Architekturmanagements.....</b>	<b>211</b>
11.1	Bedeutung der Architektur.....	212
11.1.1	Gewachsene Applikationslandschaft eines Unternehmens.....	213
11.1.2	Notwendigkeit der integrierten Gesamtsicht- auf die Applikations- landschaft.....	213
11.1.3	Eingrenzung des Architekturbegriffs.....	214
11.2	Positionierung der Entwicklung und Bereitsstellung.....	215
11.2.1	IT-Services als IT-Produkte in Kombination mit kundenorientierten Mehrwertservices.....	217
11.2.2	Berücksichtigung der Architektur im »Application Management« ..	218
11.3	Architekturmanagement.....	219
11.3.1	Kontinuierliche Entwicklung und Pflege der Architektur.....	220
11.3.2	Situationsabhängige Durchsetzung der Architektur.....	221

11.3.3	Aufgaben des Architekturmanagements.....	223
11.3.4	Institutionalisierung des Architekturmanagements. ....	224
11.4	Ergänzung des IT-Servicemanagements. ....	225
<b>12</b>	<b>Dienstorientierte Architekturgestaltung. ....</b>	<b>229</b>
12.1	Dienstorientierung in der Unternehmensarchitektur.....	229
12.2	Architekturkonstrukte des Business Engineering. ....	232
12.3	Dienste im Business-Engineering-Framework.....	234
12.3.1	Dienste auf Strategieebene. ....	234
12.3.2	Dienste auf Prozessebene. ....	235
12.3.3	Dienste auf Applikationsebene. ....	237
12.4	Dienstorientierung vs. Infrastruktur.....	240
<b>13</b>	<b>Kommunikation über Systemarchitekturen. ....</b>	<b>243</b>
13.1	Systemmodelle als Kommunikationsmittel des Architekten. ....	244
13.2	Typische Szenarien.....	244
13.3	Darstellungsmittel für Systemmodelle. ....	246
13.3.1	Anforderungen an Darstellungsmittel. ....	246
13.3.2	Grafische Gestaltungsrichtlinien. ....	247
13.4	Fundamental Modeling Concepts (FMC).....	249
13.4.1	FMC-Strukturtypen und zugehörige Diagramme. ....	249
13.4.2	Aufbaustrukturen. ....	250
13.4.3	Ablaufstruktur. ....	253
13.4.4	Wertebereichsstruktur.....	255
13.5	Sichten auf Systeme in FMC.....	256
13.5.1	Abstraktion und Verfeinerung. ....	256
13.5.2	Aspekt- und Szenariomodelle. ....	259
13.5.3	Abbildung von Systemmodellen auf objektorientierte Software- strukturen. ....	260
13.6	FMC-Modellierungsprinzipien. ....	268
13.7	Praxiserfahrungen und weiter gehende Ansätze. ....	269
13.7.1	Dokumentation bestehender Systeme. ....	269
13.7.2	Anforderungsanalyse und »Mental Prototyping«.....	269
13.7.3	Systemmodelle als Basis der Projektsteuerung. ....	270
13.7.4	Architekturbasierte Beschreibung transaktionaler Systeme. ....	271
<b>IV</b>	<b>Bewertung von Architekturen</b>	<b>273</b>
<b>14</b>	<b>Bewertungstechniken - eine allgemeine Übersicht. ....</b>	<b>277</b>
14.1	Begriffsdefinitionen. ....	278
14.1.1	Qualität. ....	278
14.1.2	Ausgewählte Qualitätsmerkmale. ....	279
14.2	Grundlagen der Architekturbewertung. ....	281

14.2.1	Ziel der Architekturbewertung	282
14.2.2	Vorteile der Architekturbewertung	282
14.2.3	Probleme der Architekturbewertung	283
14.2.4	Eine Kategorisierung von Architekturbewertungstechniken	284
14.3	Geeignete Architekturbewertungstechniken	285
14.3.1	Evaluationstechniken vs. Qualitätseigenschaft	286
14.3.2	SAAM (Scenario-based Architecture Analysis Method)	286
14.3.3	ATAM (Architecture Trade-off Analysis Method)	289
14.3.4	CBAM (Cost Benefit Analysis Method)	290
14.3.5	ALMA (Architecture-Level Modifiability Analysis)	292
<b>15</b>	<b>Bewertungstechniken für die Systemsicherheit</b>	<b>295</b>
15.1	Vorgehensmodell der Gefährdungsanalyse	297
15.2	Eine Klassifizierung von Gefährdungsanalysetechniken	298
15.3	Gefährdungsanalysetechniken	299
15.3.1	Fehlerbaumanalyse und Komponentenfehlerbäume	299
15.3.2	Zuverlässigkeits-Blockdiagramme	302
15.3.3	Markov-Analysen	302
15.3.4	HAZOP und HAZOP-basierte Techniken	304
15.3.5	FMEA und IF-FMEA	306
15.3.6	HiP-HOPS	308
15.4	Fazit	310
<b>16</b>	<b>Bewertungstechniken für die Performanz</b>	<b>311</b>
16.1	Allgemeines Vorgehensmodell	312
16.2	Klassifikation von Performanzbewertungstechniken	314
16.2.1	Architekturbeschreibung	314
16.2.2	Performanzmodelle	316
16.2.3	Auswertungsmethoden	318
16.3	Ausgewählte Verfahren	319
16.3.1	SPE	321
16.3.2	CB-SPE	322
16.3.3	umIPSI	323
16.3.4	LQN	324
16.3.5	CP	325
<b>V</b>	<b>Wiederverwendung von Architekturen</b>	<b>327</b>
<b>17</b>	<b>Software-Muster</b>	<b>331</b>
17.1	Was ist ein Muster	333
17.2	Vorteile von Mustern	340
17.3	Musterkategorien	341
17.3.1	Architekturmuster	342

## Inhaltsverzeichnis

17.3.2	Entwurfsmuster	342
17.3.3	Idiome	343
17.4	Klassifikation von Mustern	344
17.5	Musterkataloge, Mustersysteme und Mustersprachen	344
17.6	Best-Practice-Mustersysteme	345
17.7	Muster-Enzyklopädien	347
17.8	Beziehungen zwischen Mustern	347
17.9	Instanziierung von Mustern	348
17.10	Pattern-Mining	349
17.11	Muster und andere Methoden	350
17.12	Weiterführende Hinweise	352
17.13	Fazit	354
<b>18</b>	<b>Referenzarchitekturen</b>	<b>357</b>
18.1	Typen von Referenzarchitekturen	358
18.1.1	Funktionale Referenzarchitekturen	359
18.1.2	Logische Referenzarchitekturen	360
18.1.3	Technische Referenzarchitekturen	361
18.2	Beschreibung von Referenzarchitekturen	362
18.2.1	Architekturüberblick	363
18.2.2	Texturen	363
18.2.3	Referenzschnittstellen	365
18.2.4	Infrastrukturen	365
18.3	Nutzung von Referenzarchitekturen	367
18.3.1	Ausgangspunkt der Software-Entwicklung	367
18.3.2	Grundlage für modellgetriebene Software-Entwicklung	367
18.3.3	Einheitliche Sprache im Unternehmen	367
18.3.4	Betrieb und Integration	368
18.4	Entwicklung, Pflege und Weiterentwicklung	369
18.4.1	Entwicklung als Reifungsprozess	369
18.4.2	Pflege und Weiterentwicklung als Produkt	369
<b>19</b>	<b>Software-Produktlinien</b>	<b>371</b>
19.1	Entwicklung von Software-Produktlinien	372
19.2	Variabilität	375
19.2.1	Variabilität des Produktraums	376
19.2.2	Variabilität der Software-Produktlinien-Artefakte	377
19.2.3	Variabilitätsarten	378
19.2.4	Auswirkungen der Variabilitätsarten auf die Software-Produktlinien-Architektur	379
19.2.5	Orthogonale Dokumentation der Variabilität	382
19.3	Evolution von Software-Produktlinien	383
19.3.1	Reifegrade von Software-Produktlinien	383

19.3.2	Evolution der Software-Produktlinien-Artefakte	386
19.4	Vorgehensmodelle	390
19.4.1	Proaktive Entwicklungsstrategien	390
19.4.2	Reaktive Entwicklungsstrategien	391
19.5	Abgrenzung zu anderen Konzepten	392
19.6	Fazit	393
<b>20</b>	<b>Framework-Entwurf</b>	<b>395</b>
20.1	Eigenschaften von Frameworks	396
20.1.1	Umkehrung des Kontrollflusses	396
20.1.2	Vorgabe einer konkreten Anwendungsarchitektur	397
20.1.3	Anpassbarkeit durch Variationspunkte	397
20.2	Arten von Frameworks	397
20.2.1	Objektorientierte Frameworks	398
20.2.2	Komponentenbasierte Frameworks	401
20.2.3	Mischformen von Frameworks	402
20.2.4	Abgrenzung von Frameworks zu anderen Konzepten	403
20.3	Der Entwicklungsprozess von Frameworks	403
20.3.1	Entwicklung von Frameworks	404
20.3.2	Nutzung von Frameworks	405
20.3.3	Komposition von Frameworks	406
20.3.4	Evolution und Wartung von Frameworks	406
20.4	Entwurf objektorientierter Frameworks	406
20.4.1	Der Hotspot-getriebene Entwurf von objektorientierten Frameworks	407
20.4.2	Definition eines speziellen Objektmodells	407
20.4.3	Identifikation der Hotspots und Erstellung der Hotspot-Karten	409
20.4.4	Entwurf und Überarbeitung des objektorientierten Frameworks mit Metamustern	411
20.4.5	Einsatz des objektorientierten Frameworks	415
20.4.6	Andere Entwurfsmethoden für objektorientierte Frameworks	415
20.5	Entwurf von Komponenten-Frameworks	415
20.6	Fazit	416

**VI Beispiele von Architekturen 419**

<b>21</b>	<b>Multimedia-Architekturen</b>	<b>423</b>
21.1	Aspekte von Multimedia	424
21.2	Beispiele für Multimedia-Architekturen	425
21.2.1	Beispiele für (Multi-)Media-Datenbanken	425
21.2.2	Beispiele für Streaming-Architekturen	426
21.2.3	Beispiele für Präsentationsarchitekturen	427
21.3	Hintergrund zur Entwicklung von MM4U	428

21.4	Vorgehensweise bei der Entwicklung von MM4U . . . . .	429
21.5	Entwicklung des MM4U-Frameworks . . . . .	433
21.5.1	Analyse verwandter Arbeiten und spezielle Objektmodelle . . . . .	433
21.5.2	Identifizierung und Erstellung der Hotspot-Karten und Gruppen-Hotspot-Karten . . . . .	434
21.5.3	Identifikation der Komponenten und Entwurf der Framework-Architektur . . . . .	435
21.5.4	Entwurf der Multimedia-Kompositionskomponente . . . . .	435
21.5.5	Entwurf der Präsentationsformat-Generatorenkomponente . . . . .	440
21.5.6	Implementierung, Überarbeitung und Nutzung des MM4U-Frameworks . . . . .	441
21.6	Fazit . . . . .	442
<b>22</b>	<b>Peer-to-Peer-Architekturen . . . . .</b>	<b>445</b>
22.1	Definitionen . . . . .	445
22.2	Klassifikation von P2P-Architekturen . . . . .	447
22.2.1	Reine P2P-Architekturen . . . . .	447
22.2.2	Hybride P2P-Architekturen . . . . .	449
22.3	Schichten einer P2P-Applikation . . . . .	451
22.3.1	Benutzungsschnittstellen . . . . .	451
22.3.2	P2P-Dienste . . . . .	452
22.3.3	P2P-Netzwerk . . . . .	452
22.4	Beispiele von P2P-Systemen . . . . .	453
22.4.1	Napster . . . . .	453
22.4.2	Freenet . . . . .	454
22.4.3	Chord . . . . .	457
22.4.4	FastTrack . . . . .	459
22.4.5	JXTA . . . . .	460

**VII Anhang 463**

<b>A</b>	<b>Formale Techniken zur Architekturbeschreibung . . . . .</b>	<b>465</b>
A.1	Einsatz formaler Spezifikationstechniken . . . . .	465
A.2	Überblick und tabellarische Zusammenfassung . . . . .	467
A.3	Ein generischer Ansatz: Formal oder lieber doch nicht? . . . . .	475
A.4	Fazit . . . . .	476
<b>B</b>	<b>Architekturbeschreibungssprachen . . . . .</b>	<b>477</b>
B.1	Darwin . . . . .	478
B.2	MetaH/ControlH . . . . .	479
B.3	Rapide . . . . .	480
B.4	Wright . . . . .	481

<b>Glossar</b> .....	<b>483</b>
<b>Abkürzungsverzeichnis</b> .....	<b>491</b>
<b>Autorenverzeichnis</b> .....	<b>497</b>
<b>Literatur</b> .....	<b>511</b>
<b>Index</b> .....	<b>553</b>