

Inhaltsübersicht

1	Einleitung	1
1.1	Softwarearchitektur als Disziplin im Software Engineering	2
1.2	iSAQB – International Software Architecture Qualification Board	4
1.3	Certified Professional for Software Architecture – Foundation und Advanced Level	5
1.4	Zielsetzung des Buches	7
1.5	Voraussetzungen	8
1.6	Leitfaden für den Leser	9
1.7	Zielpublikum	10
1.8	Danksagungen	10
2	Grundlagen von Softwarearchitekturen	11
2.1	Einbettung in den iSAQB-Lehrplan	12
2.2	Softwareintensive Systeme und Softwarearchitekturen	13
2.3	Grundlegende Konzepte von Softwarearchitekturen	20
2.4	Der Softwarearchitektorentwurf aus der Vogelperspektive	38
2.5	Lernkontrolle	48
3	Entwurf von Softwarearchitekturen	51
3.1	Einbettung in den iSAQB-Lehrplan	52
3.2	Überblick über das Vorgehen beim Architektorentwurf	52
3.3	Entwurfsprinzipien und Heuristiken	59
3.4	Architekturzentrierte Entwicklungsansätze	64
3.5	Techniken für einen guten Entwurf	72
3.6	Architekturmuster	79
3.7	Entwurfsmuster	90
3.8	Lernkontrolle	96

4	Beschreibung und Kommunikation von Softwarearchitekturen	99
4.1	Einbettung in den iSAQB-Lehrplan	99
4.2	Das CoCoME-Beispiel	100
4.3	Sichten und Schablonen	103
4.4	Technische oder querschnittliche Konzepte in Softwarearchitekturen	132
4.5	Architektur und Implementierung	135
4.6	Übliche Dokumenttypen für Softwarearchitekturen	137
4.7	Praxisregeln zur Dokumentation	140
4.8	Beispiele weiterer Architektur-Frameworks	143
4.9	Lernkontrolle	146
5	Softwarearchitekturen und Qualität	149
5.1	Einbettung in den iSAQB-Lehrplan	150
5.2	Bewertung von Softwarearchitekturen	151
5.3	EXKURS: Prototyp und technischer Durchstich	159
5.4	Architekturanalyse	161
5.5	Lernkontrolle	169
6	EXKURS: Werkzeuge für Softwarearchitekten	171
6.1	Allgemeine Hinweise zu Werkzeugen	171
6.2	Werkzeuge zum Anforderungsmanagement	172
6.3	Werkzeuge zur Modellierung	174
6.4	Werkzeuge zur Generierung	175
6.5	Werkzeuge zur statischen Codeanalyse	176
6.6	Werkzeuge zur dynamischen Analyse	177
6.7	Werkzeuge zum Build-Management	179
6.8	Werkzeuge zum Konfigurations- und Versionsmanagement	180
6.9	Werkzeuge zum Codemanagement	181
6.10	Werkzeuge zum Test	182
6.11	Werkzeuge zur Dokumentation	183

Anhang	185	
A	Beispielfragen	187
A.1	Auszüge aus der Prüfungsordnung	187
A.2	Beispielfragen	189
B	Abkürzungsverzeichnis	193
C	Glossar	195
D	Literaturverzeichnis	209
	Index	215

Inhaltsverzeichnis

1	Einleitung	1
1.1	Softwarearchitektur als Disziplin im Software Engineering	2
1.2	iSAQB – International Software Architecture Qualification Board	4
1.3	Certified Professional for Software Architecture – Foundation und Advanced Level	5
1.4	Zielsetzung des Buches	7
1.5	Voraussetzungen	8
1.6	Leitfaden für den Leser	9
1.7	Zielpublikum	10
1.8	Danksagungen	10
2	Grundlagen von Softwarearchitekturen	11
2.1	Einbettung in den iSAQB-Lehrplan	12
2.1.1	Lernziele	12
2.2	Softwareintensive Systeme und Softwarearchitekturen	13
2.2.1	Was ist ein softwareintensives System?	13
2.2.2	EXKURS: Ausprägungen von softwareintensiven Systemen	15
2.2.3	Bedeutung der Softwarearchitektur für ein softwareintensives System	19
2.3	Grundlegende Konzepte von Softwarearchitekturen	20
2.3.1	Was ist eine Softwarearchitektur?	21
2.3.2	Bausteine, Schnittstellen und Konfigurationen	22
2.3.3	Konzepte der Beschreibung von Softwarearchitekturen	29
2.3.4	Architekturbeschreibung und Architekturebenen	33
2.3.5	Wechselwirkungen zwischen Softwarearchitektur und Umgebung	35
2.3.6	Qualität und Nutzen der Softwarearchitektur	37

2.4	Der Softwarearchitekturentwurf aus der Vogelperspektive	38
2.4.1	Ziele und Aufgaben des Softwarearchitekturentwurfs	39
2.4.2	Der Softwarearchitekturentwurf im Überblick	41
2.4.3	Wechselspiel der Tätigkeiten und Abstraktionsstufen im Entwurf	43
2.4.4	EXKURS: Aufgaben des Softwarearchitekten und Bezug zu anderen Rollen	46
2.5	Lernkontrolle	48
3	Entwurf von Softwarearchitekturen	51
3.1	Einbettung in den iSAQB-Lehrplan	52
3.1.1	Lernziele	52
3.2	Überblick über das Vorgehen beim Architekturentwurf	52
3.3	Entwurfsprinzipien und Heuristiken	59
3.3.1	Top-down und bottom-up	59
3.3.2	Hierarchische (De-)Komposition	61
3.3.2.1	Divide et impera	61
3.3.2.2	Prinzipien bei der Zerlegung	61
3.3.2.3	So-einfach-wie-möglich-Prinzip	62
3.3.2.4	Trennung von Verantwortlichkeiten	62
3.3.3	Schmale Schnittstellen und Information Hiding	63
3.3.3.1	Information Hiding	63
3.3.3.2	Verwendung von Schnittstellen	63
3.3.4	Regelmäßiges Refactoring und Redesign	63
3.4	Architekturzentrierte Entwicklungsansätze	64
3.4.1	EXKURS: Domain Driven Design	65
3.4.1.1	Fachmodelle als Basis	65
3.4.1.2	Systematische Verwaltung der Domänenobjekte	66
3.4.1.3	Strukturierung der Fachdomäne	66
3.4.1.4	Arten von Domänen	67
3.4.1.5	Integration von Domänen	67
3.4.2	EXKURS: MDA	68
3.4.3	Referenzarchitekturen	70
3.4.3.1	Generative Erzeugung von Systembausteinen	70
3.4.3.2	Aspektorientierung	70
3.4.3.3	Objektorientierung	71
3.4.3.4	Prozedurale Ansätze	72

3.5	Techniken für einen guten Entwurf	72
3.5.1	Ausgangssituation und Motivation: degeneriertes Design .	73
3.5.2	Lose Kopplung	74
3.5.3	Hohe Kohäsion	75
3.5.4	Offen-geschlossen-Prinzip	76
3.5.5	Umkehr der Abhängigkeiten	76
3.5.6	Abtrennung von Schnittstellen	77
3.5.7	Zyklische Abhängigkeiten auflösen	78
3.5.8	Liskov'sches Substitutionsprinzip	78
3.6	Architekturmuster	79
3.6.1	Adaptierbare Systeme	80
3.6.1.1	Dependency Injection	80
3.6.2	Interaktive Systeme	81
3.6.2.1	Model View Controller	81
3.6.2.2	Model View Presenter	82
3.6.2.3	Presentation Abstraction Control	83
3.6.3	Vom Chaos zur Struktur	84
3.6.3.1	Schichtenarchitektur	84
3.6.3.2	Pipes and Filters	85
3.6.3.3	Blackboard	86
3.6.4	Verteilte Systeme	86
3.6.4.1	Broker	87
3.6.4.2	EXKURS: Serviceorientierung	88
3.6.4.3	Modularisierung	89
3.6.4.4	Microservices	89
3.7	Entwurfsmuster	90
3.7.1	Adapter	90
3.7.2	Observer	91
3.7.3	Decorator	92
3.7.4	Proxy	92
3.7.5	Fassade	93
3.7.6	Brücke	94
3.7.7	State	94
3.7.8	Mediator	95
3.8	Lernkontrolle	96

4	Beschreibung und Kommunikation von Softwarearchitekturen	99
4.1	Einbettung in den iSAQB-Lehrplan	99
4.1.1	Lernziele	100
4.2	Das CoCoME-Beispiel	100
4.2.1	Anwendungsfälle im CoCoME-System	101
4.2.2	Übersicht über den strukturellen Aufbau des CoCoME-Systems	102
4.3	Sichten und Schablonen	103
4.3.1	Bewährte Sichten nach iSAQB	103
4.3.2	UML-Diagramme als Notationsmittel in Sichtenbeschreibungen	105
4.3.3	Sichtenbeschreibung – Grobaufbau und Einführungsbeispiel	108
4.3.3.1	Grobaufbau – schablonenartige Sichtenbeschreibung	108
4.3.3.2	Beispiel: Auszug aus einer Sichtenbeschreibung für eine Bausteinsicht	110
4.3.4	Kontextsicht oder Kontextabgrenzung	112
4.3.5	Bausteinsicht	116
4.3.6	Laufzeitsicht	119
4.3.7	Verteilungssicht bzw. Infrastruktursicht	124
4.3.8	Wechselwirkungen zwischen Architektursichten	128
4.3.9	Hierarchische Verfeinerung von Architektursichten	129
4.4	Technische oder querschnittliche Konzepte in Softwarearchitekturen	132
4.4.1	Technische bzw. querschnittliche Konzepte: Beispieldimensionen	133
4.4.2	Beispiel: Fehlerbehandlung	133
4.4.3	Beispiel: Sicherheit	134
4.5	Architektur und Implementierung	135
4.5.1	Beispiel: Implementierung	136
4.6	Übliche Dokumenttypen für Softwarearchitekturen	137
4.6.1	Zentrale Architekturbeschreibung	137
4.6.2	Architekturüberblick	138
4.6.3	Dokumentübersicht	138
4.6.4	Übersichtspräsentation	138
4.6.5	»Architekturtafel«	139
4.6.6	Handbuch zur Dokumentation	139

4.6.7	Technische Informationen	139
4.6.8	Dokumentation von externen Schnittstellen	139
4.6.9	Template	140
4.7	Praxisregeln zur Dokumentation	140
4.7.1	Regel 1: »Schreiben aus der Sicht des Lesers«	140
4.7.2	Regel 2: »Unnötige Wiederholung vermeiden«	141
4.7.3	Regel 3: »Mehrdeutigkeit vermeiden«	141
4.7.4	Regel 4: »Standardisierte Organisationsstruktur bzw. Schablonen«	141
4.7.5	Regel 5: »Begründen Sie wesentliche Entscheidungen schriftlich«	142
4.7.6	Regel 6: »Überprüfung auf Gebrauchstauglichkeit«	142
4.7.7	Regel 7: »Übersichtliche Diagramme«	142
4.7.8	Regel 8: »Regelmäßige Aktualisierungen«	143
4.8	Beispiele weiterer Architektur-Frameworks	143
4.8.1	4+1-Framework	144
4.8.2	RM-ODP	144
4.8.3	SAGA	146
4.9	Lernkontrolle	146
5	Softwarearchitekturen und Qualität	149
5.1	Einbettung in den iSAQB-Lehrplan	150
5.1.1	Lernziele	150
5.2	Bewertung von Softwarearchitekturen	151
5.2.1	Qualitative Bewertung	151
5.2.1.1	DIN ISO/IEC 25010	151
5.2.1.2	Qualitätsmerkmale	151
5.2.1.3	Weitere Qualitätsmerkmale	153
5.2.1.4	Auswirkungen bestimmter Qualitätsmerkmale	154
5.2.1.5	Taktiken und Praktiken	154
5.2.2	Quantitative Bewertung	156
5.2.2.1	Überprüfung von Architekturregeln	156
5.2.2.2	Metriken	157
5.2.2.3	Zyklomatische Komplexität	158
5.3	EXKURS: Prototyp und technischer Durchstich	159
5.3.1	Technischer Durchstich	159
5.3.2	Prototyp	159
5.3.2.1	Einsatz von Softwareprototypen	159
5.3.2.2	Arten von Softwareprototypen	160

5.4	Architekturanalyse	161
5.4.1	EXKURS: ATAM-Methode	161
	5.4.1.1 Vorgehen bei der Bewertung	161
5.5	Lernkontrolle	169
6	EXKURS: Werkzeuge für Softwarearchitekten	171
6.1	Allgemeine Hinweise zu Werkzeugen	171
6.1.1	Kosten von Werkzeugen	171
6.1.2	Lizenzen und Lizenzbedingungen	172
6.2	Werkzeuge zum Anforderungsmanagement	172
6.2.1	Anforderungen und Entscheidungskriterien	173
6.2.2	Herausforderungen von Werkzeugen für das Anforderungsmanagement	173
6.2.3	Beispielhafte Vertreter	173
6.3	Werkzeuge zur Modellierung	174
6.3.1	Anforderungen und Entscheidungskriterien	174
6.3.2	Herausforderungen von Werkzeugen für die Modellierung	175
6.3.3	Beispielhafte Vertreter	175
6.4	Werkzeuge zur Generierung	175
6.4.1	Anforderungen und Entscheidungskriterien	176
6.4.2	Herausforderungen von Codegeneratoren	176
6.4.3	Beispielhafte Vertreter	176
6.5	Werkzeuge zur statischen Codeanalyse	176
6.5.1	Anforderungen und Entscheidungskriterien	177
6.5.2	Herausforderungen von Werkzeugen zur statischen Codeanalyse	177
6.5.3	Beispielhafte Vertreter	177
6.6	Werkzeuge zur dynamischen Analyse	177
6.6.1	Anforderungen und Entscheidungskriterien	178
6.6.2	Herausforderungen von Werkzeugen zur dynamischen Analyse	178
6.6.3	Beispielhafte Vertreter	178
6.7	Werkzeuge zum Build-Management	179
6.7.1	Anforderungen und Entscheidungskriterien	179
6.7.2	Herausforderungen von Werkzeugen zum Build-Management	179
6.7.3	Beispielhafte Vertreter	180

6.8	Werkzeuge zum Konfigurations- und Versionsmanagement	180
6.8.1	Anforderungen und Entscheidungskriterien	180
6.8.2	Herausforderungen von Werkzeugen zum Konfigurations- und Versionsmanagement	181
6.8.3	Beispielhafte Vertreter	181
6.9	Werkzeuge zum Codemanagement	181
6.9.1	Herausforderungen von Werkzeugen zum Codemanagement	182
6.9.2	Beispielhafte Vertreter	182
6.10	Werkzeuge zum Test	182
6.10.1	Anforderungen und Entscheidungskriterien	183
6.10.2	Herausforderungen von Testwerkzeugen	183
6.10.3	Beispielhafte Vertreter	183
6.11	Werkzeuge zur Dokumentation	183
6.11.1	Anforderungen und Entscheidungskriterien	183
6.11.2	Herausforderungen von Dokumentationswerkzeugen . . .	184
6.11.3	Beispielhafte Vertreter	184

Anhang **185**

A	Beispielfragen	187
A.1	Auszüge aus der Prüfungsordnung	187
A.2	Beispielfragen	189
B	Abkürzungsverzeichnis	193
C	Glossar	195
D	Literaturverzeichnis	209
	Index	215