

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 Softwarearchitekturentwurf 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 Architekturentwurf	52
3.3	Arbeit mit Randbedingungen und äußeren Einflussfaktoren	59
3.4	Entwurfsprinzipien und Heuristiken	61
3.5	Architekturzentrierte Entwicklungsansätze	68
3.6	Techniken für einen guten Entwurf	78
3.7	Architekturmuster	86
3.8	EXKURS: Entwurfsmuster	98
3.9	Deployment und Betrieb	106
3.10	Lernkontrolle	110

4	Beschreibung und Kommunikation von Softwarearchitekturen	115
4.1	Einbettung in den iSAQB-Lehrplan	115
4.2	Das CoCoME-Beispiel	116
4.3	Sichten und Schablonen	119
4.4	Technische oder querschnittliche Konzepte in Softwarearchitekturen	148
4.5	Architektur und Implementierung	151
4.6	Übliche Dokumenttypen für Softwarearchitekturen	153
4.7	Praxisregeln zur Dokumentation	157
4.8	Beispiele weiterer Architektur-Frameworks	160
4.9	Lernkontrolle	162
5	Softwarearchitekturen und Qualität	165
5.1	Einbettung in den iSAQB-Lehrplan	166
5.2	Bewertung von Softwarearchitekturen	167
5.3	EXKURS: Prototyp und technischer Durchstich	176
5.4	Architekturanalyse	178
5.5	Lernkontrolle	185
6	EXKURS: Werkzeuge für Softwarearchitektinnen	187
6.1	Allgemeine Hinweise zu Werkzeugen	187
6.2	Werkzeuge zum Anforderungsmanagement	188
6.3	Werkzeuge zur Modellierung	190
6.4	Werkzeuge zur statischen Codeanalyse	191
6.5	Werkzeuge zur dynamischen Analyse	193
6.6	Werkzeuge zum Konfigurations- und Versionsmanagement	194
6.7	Werkzeuge zum Codemanagement	195
6.8	Werkzeuge zum Test	196
6.9	Werkzeuge zur Dokumentation	197
Anhang		199
A	Beispielfragen	201
A.1	Auszüge aus der Prüfungsordnung	201
A.2	Beispielfragen	203
B	Abkürzungsverzeichnis	207
C	Glossar	209
D	Literaturverzeichnis	221
	Index	227

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 der Softwarearchitektin 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	Arbeit mit Randbedingungen und äußeren Einflussfaktoren	59
3.3.1	Arten von Einflussfaktoren	59
3.4	Entwurfsprinzipien und Heuristiken	61
3.4.1	Top-down und bottom-up	61
3.4.2	Hierarchische (De-)Komposition	62
3.4.2.1	Divide et impera	62
3.4.2.2	Prinzipien bei der Zerlegung	63
3.4.2.3	So-einfach-wie-möglich-Prinzip	63
3.4.2.4	Trennung von Verantwortlichkeiten	64
3.4.3	Konzeptionelle Integrität	65
3.4.4	Erwarte Fehler	65
3.4.4.1	Postels' Law	66
3.4.5	Schmale Schnittstellen und Information Hiding	66
3.4.5.1	Information Hiding	66
3.4.5.2	Verwendung von Schnittstellen	66
3.4.6	Regelmäßiges Refactoring und Redesign	67
3.5	Architekturzentrierte Entwicklungsansätze	68
3.5.1	EXKURS: Domain-Driven Design	68
3.5.1.1	Fachmodelle als Basis	68
3.5.1.2	Systematische Verwaltung der Domänenobjekte	69
3.5.1.3	Strukturierung der Fachdomäne	70
3.5.1.4	Arten von Domänen	70
3.5.1.5	Integration von Domänen	71
3.5.2	EXKURS: Globale Analyse	71
3.5.3	EXKURS: Evolutionäre Architektur	72
3.5.3.1	Prinzipien	72
3.5.3.2	Fitnessfunktionen	73

3.5.4	EXKURS: Modellgetriebene Architektur	74
3.5.5	Referenzarchitekturen	76
3.5.5.1	Generative Erzeugung von Systembausteinen	76
3.5.5.2	Aspektorientierung	76
3.5.5.3	Objektorientierung	77
3.5.5.4	Prozedurale Ansätze	78
3.6	Techniken für einen guten Entwurf	78
3.6.1	Ausgangssituation und Motivation: degeneriertes Design	79
3.6.2	Lose Kopplung	80
3.6.3	Hohe Kohäsion	81
3.6.4	Single-Responsibility-Prinzip	82
3.6.5	Offen-geschlossen-Prinzip	82
3.6.6	Umkehr der Abhängigkeiten	83
3.6.7	Abtrennung von Schnittstellen	84
3.6.8	Zyklische Abhängigkeiten auflösen	84
3.6.9	Liskov'sches Substitutionsprinzip	85
3.7	Architekturmuster	86
3.7.1	Adaptierbare Systeme	86
3.7.1.1	Dependency Injection	86
3.7.2	Interaktive Systeme	87
3.7.2.1	Model View Controller	87
3.7.2.2	Model View Presenter	88
3.7.2.3	Presentation Abstraction Control	89
3.7.3	Vom Chaos zur Struktur	90
3.7.3.1	Schichtenarchitektur	90
3.7.3.2	Pipes and Filters	91
3.7.3.3	Blackboard	92
3.7.4	Verteilte Systeme	93
3.7.4.1	Herausforderungen verteilter Systeme	93
3.7.4.2	Broker	95
3.7.4.3	EXKURS: Serviceorientierung	96
3.7.4.4	Modularisierung	97
3.7.4.5	Microservices	97
3.8	EXKURS: Entwurfsmuster	98
3.8.1	Adapter	98
3.8.2	Observer	99
3.8.3	Decorator	99
3.8.4	Proxy	100
3.8.5	Fassade	101

3.8.6	Brücke	102
3.8.7	State	103
3.8.8	Mediator	103
3.8.9	Fabrik	104
3.8.10	Interpreter	104
3.8.11	Plug-in	105
3.8.12	Kombinator	106
3.9	Deployment und Betrieb	106
3.9.1	Deployment	106
3.9.2	Betrieb	108
3.9.3	EXKURS: DevOps	109
3.10	Lernkontrolle	110
4	Beschreibung und Kommunikation von Softwarearchitekturen	115
4.1	Einbettung in den iSAQB-Lehrplan	115
4.1.1	Lernziele	116
4.2	Das CoCoME-Beispiel	116
4.2.1	Anwendungsfälle im CoCoME-System	117
4.2.2	Übersicht über den strukturellen Aufbau des CoCoME-Systems	118
4.3	Sichten und Schablonen	119
4.3.1	Bewährte Sichten nach iSAQB	119
4.3.2	UML-Diagramme als Notationsmittel in Sichtenbeschreibungen	121
4.3.3	Sichtenbeschreibung – Grobaufbau und Einführungsbeispiel	124
4.3.3.1	Grobaufbau – schablonenartige Sichtenbeschreibung	124
4.3.3.2	Beispiel: Auszug aus einer Sichtenbeschreibung für eine Bausteinsicht	126
4.3.4	Kontextsicht oder Kontextabgrenzung	128
4.3.5	Bausteinsicht	132
4.3.6	Laufzeitsicht	135
4.3.7	Verteilungssicht bzw. Infrastruktursicht	140
4.3.8	Wechselwirkungen zwischen Architektursichten	144
4.3.9	Hierarchische Verfeinerung von Architektursichten	145

4.4	Technische oder querschnittliche Konzepte in Softwarearchitekturen	148
4.4.1	Technische bzw. querschnittliche Konzepte: Beispieldimensionen	149
4.4.2	Beispiel: Fehlerbehandlung	149
4.4.3	Beispiel: Sicherheit	150
4.5	Architektur und Implementierung	151
4.5.1	Beispiel: Implementierung	152
4.6	Übliche Dokumenttypen für Softwarearchitekturen	153
4.6.1	Zentrale Architekturbeschreibung	153
4.6.2	Architekturüberblick	154
4.6.3	Dokumentübersicht	154
4.6.4	Übersichtspräsentation	154
4.6.5	»Architekturtapete«	154
4.6.6	Handbuch zur Dokumentation	155
4.6.7	Architecture Decision Record	155
4.6.8	Technische Informationen	156
4.6.9	Dokumentation von externen Schnittstellen	156
4.6.10	Template	156
4.7	Praxisregeln zur Dokumentation	157
4.7.1	Regel 1: »Schreiben aus der Sicht der Leserin«	157
4.7.2	Regel 2: »Unnötige Wiederholung vermeiden«	157
4.7.3	Regel 3: »Mehrdeutigkeit vermeiden«	157
4.7.4	Regel 4: »Standardisierte Organisationsstruktur bzw. Schablonen«	158
4.7.5	Regel 5: »Begründen Sie wesentliche Entscheidungen schriftlich«	158
4.7.6	Regel 6: »Überprüfung auf Gebrauchstauglichkeit«	159
4.7.7	Regel 7: »Übersichtliche Diagramme«	159
4.7.8	Regel 8: »Regelmäßige Aktualisierungen«	159
4.7.9	EXKURS: Regel 9: »Passen Sie die Änderbarkeit der Dokumentation an die Architektur an«	160
4.8	Beispiele weiterer Architektur-Frameworks	160
4.8.1	4+1-Framework	161
4.8.2	SAGA	161
4.9	Lernkontrolle	162

5	Softwarearchitekturen und Qualität	165
5.1	Einbettung in den iSAQB-Lehrplan	166
5.1.1	Lernziele	166
5.2	Bewertung von Softwarearchitekturen	167
5.2.1	Qualitative Bewertung	167
5.2.1.1	DIN ISO/IEC 25010	167
5.2.1.2	Qualitätsmerkmale	168
5.2.1.3	Weitere Qualitätsmerkmale	170
5.2.1.4	Auswirkungen bestimmter Qualitätsmerkmale . .	170
5.2.1.5	Taktiken und Praktiken	171
5.2.2	Quantitative Bewertung	172
5.2.2.1	Überprüfung von Architekturregeln	173
5.2.2.2	Metriken	174
5.2.2.3	Zyklomatische Komplexität	175
5.2.2.4	Goodharts Gesetz	175
5.3	EXKURS: Prototyp und technischer Durchstich	176
5.3.1	Technischer Durchstich	176
5.3.2	Prototyp	176
5.3.2.1	Einsatz von Softwareprototypen	176
5.3.2.2	Arten von Softwareprototypen	177
5.4	Architekturanalyse	178
5.4.1	EXKURS: ATAM-Methode	178
5.4.1.1	Vorgehen bei der Bewertung	178
5.5	Lernkontrolle	185
6	EXKURS: Werkzeuge für Softwarearchitektinnen	187
6.1	Allgemeine Hinweise zu Werkzeugen	187
6.1.1	Kosten von Werkzeugen	187
6.1.2	Lizenzen und Lizenzbedingungen	188
6.2	Werkzeuge zum Anforderungsmanagement	188
6.2.1	Anforderungen und Entscheidungskriterien	189
6.2.2	Herausforderungen von Werkzeugen für das Anforderungsmanagement	189
6.2.3	Beispielhafte Vertreter	189
6.3	Werkzeuge zur Modellierung	190
6.3.1	Anforderungen und Entscheidungskriterien	190
6.3.2	Herausforderungen von Werkzeugen für die Modellierung	191
6.3.3	Beispielhafte Vertreter	191

6.4	Werkzeuge zur statischen Codeanalyse	191
6.4.1	Anforderungen und Entscheidungskriterien	192
6.4.2	Herausforderungen von Werkzeugen zur statischen Codeanalyse	192
6.4.3	Beispielhafte Vertreter	192
6.5	Werkzeuge zur dynamischen Analyse	193
6.5.1	Anforderungen und Entscheidungskriterien	193
6.5.2	Herausforderungen von Werkzeugen zur dynamischen Analyse	193
6.5.3	Beispielhafte Vertreter	193
6.6	Werkzeuge zum Konfigurations- und Versionsmanagement	194
6.6.1	Anforderungen und Entscheidungskriterien	194
6.6.2	Herausforderungen von Werkzeugen zum Konfigurations- und Versionsmanagement	194
6.6.3	Beispielhafte Vertreter	195
6.7	Werkzeuge zum Codemanagement	195
6.7.1	Herausforderungen von Werkzeugen zum Codemanagement	195
6.7.2	Beispielhafte Vertreter	196
6.8	Werkzeuge zum Test	196
6.8.1	Anforderungen und Entscheidungskriterien	196
6.8.2	Herausforderungen von Testwerkzeugen	197
6.8.3	Beispielhafte Vertreter	197
6.9	Werkzeuge zur Dokumentation	197
6.9.1	Anforderungen und Entscheidungskriterien	197
6.9.2	Herausforderungen von Dokumentationswerkzeugen	198
6.9.3	Beispielhafte Vertreter	198

Anhang

199

A	Beispielfragen	201
A.1	Auszüge aus der Prüfungsordnung	201
A.2	Beispielfragen	203
B	Abkürzungsverzeichnis	207
C	Glossar	209
D	Literaturverzeichnis	221
	Index	227