

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	Ausprägungen von softwareintensiven Systemen	14
2.2.3	Bedeutung der Softwarearchitektur für ein softwareintensives System	18
2.3	Grundlegende Konzepte von Softwarearchitekturen	19
2.3.1	Was ist eine Softwarearchitektur?	20
2.3.2	Bausteine, Schnittstellen und Konfigurationen	21
2.3.3	Konzepte der Beschreibung von Softwarearchitekturen	28
2.3.4	Architekturbeschreibung und Architekturebenen	32
2.3.5	Wechselwirkungen zwischen Softwarearchitektur und Umgebung	34
2.3.6	Qualität und Nutzen der Softwarearchitektur	36

2.4	Der Softwarearchitekturentwurf aus der Vogelperspektive	37
2.4.1	Ziele und Aufgaben des Softwarearchitekturentwurfs	38
2.4.2	Der Softwarearchitekturentwurf im Überblick	40
2.4.3	Wechselspiel der Tätigkeiten und Abstraktionsstufen im Entwurf	42
2.4.4	Aufgaben des Softwarearchitekten und Bezug zu anderen Rollen	44
2.5	Lernkontrolle	46
3	Entwurf von Softwarearchitekturen	49
3.1	Einbettung in den iSAQB-Lehrplan	50
3.1.1	Lernziele	50
3.2	Überblick über das Vorgehen beim Architekturentwurf	50
3.3	Entwurfsprinzipien und Heuristiken	57
3.3.1	Top-down und bottom-up	57
3.3.2	Hierarchische (De-)Komposition	59
3.3.2.1	Divide et impera	59
3.3.2.2	Prinzipien bei der Zerlegung	59
3.3.2.3	So-einfach-wie-möglich-Prinzip	60
3.3.2.4	Trennung von Verantwortlichkeiten	60
3.3.3	Schmale Schnittstellen und Information Hiding	61
3.3.3.1	Information Hiding	61
3.3.3.2	Verwendung von Schnittstellen	61
3.3.4	Regelmäßiges Refactoring und Redesign	61
3.4	Architekturzentrierte Entwicklungsansätze	62
3.4.1	Domain Driven Design	62
3.4.1.1	Fachmodelle als Basis	63
3.4.1.2	Systematische Verwaltung der Domänenobjekte	63
3.4.1.3	Strukturierung der Fachdomäne	64
3.4.1.4	Arten von Domänen	65
3.4.1.5	Integration von Domänen	65
3.4.2	MDA	66
3.4.3	Referenzarchitekturen	67
3.4.3.1	Generative Erzeugung von Systembausteinen	67
3.4.3.2	Aspektororientierung	67
3.4.3.3	Objektorientierung	68
3.4.3.4	Prozedurale Ansätze	69

3.5	Techniken für einen guten Entwurf	69
3.5.1	Ausgangssituation und Motivation: Degeneriertes Design	70
3.5.2	Lose Kopplung	71
3.5.3	Hohe Kohäsion	72
3.5.4	Offen-Geschlossen-Prinzip	73
3.5.5	Umkehr der Abhängigkeiten	73
3.5.6	Abtrennung von Schnittstellen	74
3.5.7	Zyklische Abhängigkeiten auflösen	75
3.5.8	Liskov'sches Substitutionsprinzip	75
3.6	Architekturmuster	76
3.6.1	Adaptierbare Systeme	77
3.6.1.1	Dependency Injection	77
3.6.2	Interaktive Systeme	78
3.6.2.1	Model View Controller	78
3.6.2.2	Model View Presenter	79
3.6.2.3	Presentation Abstraction Control	80
3.6.3	Vom Chaos zur Struktur	81
3.6.3.1	Schichtenarchitektur	81
3.6.3.2	Pipes and Filters	82
3.6.3.3	Blackboard	83
3.6.4	Verteilte Systeme	83
3.6.4.1	Broker	84
3.6.4.2	Serviceorientierung	85
3.6.4.3	Modularisierung	86
3.6.4.4	Microservices	86
3.7	Entwurfsmuster	87
3.7.1	Adapter	87
3.7.2	Observer	88
3.7.3	Decorator	89
3.7.4	Proxy	89
3.7.5	Fassade	90
3.7.6	Brücke	91
3.7.7	State	91
3.7.8	Mediator	92
3.8	Lernkontrolle	93

4	Beschreibung und Kommunikation von Softwarearchitekturen	97
4.1	Einbettung in den iSAQB-Lehrplan	97
4.1.1	Lernziele	98
4.2	Das CoCoME-Beispiel	98
4.2.1	Anwendungsfälle im CoCoME-System	99
4.2.2	Übersicht über den strukturellen Aufbau des CoCoME-Systems	100
4.3	Sichten und Schablonen	101
4.3.1	Bewährte Sichten nach iSAQB	101
4.3.2	UML-Diagramme als Notationsmittel in Sichtenbeschreibungen	103
4.3.3	Sichtenbeschreibung – Grobaufbau und Einführungsbeispiel	106
4.3.3.1	Grobaufbau – schablonenartige Sichtenbeschreibung	106
4.3.3.2	Beispiel: Auszug aus einer Sichten- beschreibung für eine Bausteinsicht	108
4.3.4	Kontextsicht oder Kontextabgrenzung	110
4.3.5	Bausteinsicht	114
4.3.6	Laufzeitsicht	117
4.3.7	Verteilungssicht bzw. Infrastruktursicht	121
4.3.8	Wechselwirkungen zwischen Architektursichten	125
4.3.9	Hierarchische Verfeinerung von Architektursichten	125
4.4	Technische oder querschnittliche Konzepte in Softwarearchitekturen	128
4.4.1	Technische bzw. querschnittliche Konzepte: Beispieldimensionen	129
4.4.2	Beispiel: Fehlerbehandlung	129
4.4.3	Beispiel: Sicherheit	130
4.5	Architektur und Implementierung	131
4.5.1	Beispiel Implementierung	132
4.6	Übliche Dokumenttypen für Softwarearchitekturen	133
4.6.1	Zentrale Architekturbeschreibung	133
4.6.2	Architekturüberblick	134
4.6.3	Dokumentübersicht	134
4.6.4	Übersichtspräsentation	134
4.6.5	»Architekturtapete«	134

4.6.6	Handbuch zur Dokumentation	135
4.6.7	Technische Informationen	135
4.6.8	Dokumentation von externen Schnittstellen	135
4.6.9	Template	135
4.7	Praxisregeln zur Dokumentation	136
4.7.1	Regel 1: »Schreiben aus der Sicht des Lesers«	136
4.7.2	Regel 2: »Unnötige Wiederholung vermeiden«	137
4.7.3	Regel 3: »Mehrdeutigkeit vermeiden«	137
4.7.4	Regel 4: »Standardisierte Organisationsstruktur bzw. Schablonen«	137
4.7.5	Regel 5: »Begründen Sie wesentliche Entscheidungen schriftlich«	138
4.7.6	Regel 6: »Überprüfung auf Gebrauchstauglichkeit«	138
4.7.7	Regel 7: »Übersichtliche Diagramme«	138
4.7.8	Regel 8: »Regelmäßige Aktualisierungen«	139
4.8	Beispiele weiterer Architektur-Frameworks	139
4.8.1	4+1-Framework	140
4.8.2	RM-ODP	140
4.8.3	SAGA	142
4.9	Lernkontrolle	142
5	Softwarearchitekturen und Qualität	145
5.1	Einbettung in den iSAQB-Lehrplan	146
5.1.1	Lernziele	146
5.2	Bewertung von Softwarearchitekturen	147
5.2.1	Qualitative Bewertung	147
5.2.1.1	DIN ISO/IEC 25010	147
5.2.1.2	Qualitätsmerkmale	147
5.2.1.3	Weitere Qualitätsmerkmale	149
5.2.1.4	Auswirkungen bestimmter Qualitätsmerkmale	150
5.2.1.5	Taktiken und Praktiken	150
5.2.2	Quantitative Bewertung	152
5.2.2.1	Überprüfung von Architekturregeln	152
5.2.2.2	Metriken	153
5.2.2.3	Zyklomatische Komplexität	154

5.3	Prototyp und technischer Durchstich	154
5.3.1	Technischer Durchstich	155
5.3.2	Prototyp	155
5.3.2.1	Einsatz von Softwareprototypen	155
5.3.2.2	Arten von Softwareprototypen	156
5.4	Architekturanalyse	156
5.4.1	ATAM-Methode	156
5.5	Lernkontrolle	163
6	Werkzeuge für Softwarearchitekten	165
6.1	Einbettung in den iSAQB-Lehrplan	165
6.1.1	Lernziele	165
6.2	Allgemeine Hinweise zu Werkzeugen	165
6.2.1	Kosten von Werkzeugen	166
6.2.2	Lizenzen und Lizenzbedingungen	166
6.3	Werkzeuge zum Anforderungsmanagement	167
6.3.1	Anforderungen und Entscheidungskriterien	167
6.3.2	Herausforderungen von Werkzeugen für das Anforderungsmanagement	167
6.3.3	Beispielhafte Vertreter	167
6.4	Werkzeuge zur Modellierung	168
6.4.1	Anforderungen und Entscheidungskriterien	168
6.4.2	Herausforderungen von Werkzeugen für die Modellierung	168
6.4.3	Beispielhafte Vertreter	169
6.5	Werkzeuge zur Generierung	169
6.5.1	Anforderungen und Entscheidungskriterien	169
6.5.2	Herausforderungen von Codegeneratoren	169
6.5.3	Beispielhafte Vertreter	170
6.6	Werkzeuge zur statischen Codeanalyse	170
6.6.1	Anforderungen und Entscheidungskriterien	170
6.6.2	Herausforderungen von Werkzeugen zur statischen Codeanalyse	170
6.6.3	Beispielhafte Vertreter	171

6.7	Werkzeuge zur dynamischen Analyse	171
6.7.1	Anforderungen und Entscheidungskriterien	171
6.7.2	Herausforderungen von Werkzeugen zur dynamischen Analyse	171
6.7.3	Beispielhafte Vertreter	172
6.8	Werkzeuge zum Build-Management	172
6.8.1	Anforderungen und Entscheidungskriterien	172
6.8.2	Herausforderungen von Werkzeugen zum Build-Management	172
6.8.3	Beispielhafte Vertreter	173
6.9	Werkzeuge zum Konfigurations- und Versionsmanagement	173
6.9.1	Anforderungen und Entscheidungskriterien	173
6.9.2	Herausforderungen von Werkzeugen zum Konfigurations- und Versionsmanagement	174
6.9.3	Beispielhafte Vertreter	174
6.10	Werkzeuge zum Codemanagement	174
6.10.1	Herausforderungen von Werkzeugen zum Codemanagement	175
6.10.2	Beispielhafte Vertreter	175
6.11	Werkzeuge zum Test	175
6.11.1	Anforderungen und Entscheidungskriterien	175
6.11.2	Herausforderungen von Testwerkzeugen	176
6.11.3	Beispielhafte Vertreter	176
6.12	Werkzeuge zur Dokumentation	176
6.12.1	Anforderungen und Entscheidungskriterien	176
6.12.2	Herausforderungen von Dokumentationswerkzeugen	177
6.12.3	Beispielhafte Vertreter	177
6.13	Lernkontrolle	177

Anhang

179

A	Beispielfragen	181
A.1	Auszüge aus der Prüfungsordnung	181
A.2	Beispielfragen	182