

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
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
<b>2</b>	<b>Grundlagen von Softwarearchitekturen</b>	<b>11</b>
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 . . . . .	31
2.3.5	Wechselwirkungen zwischen Softwarearchitektur und Umgebung . . . . .	33
2.3.6	Qualität und Nutzen der Softwarearchitektur . . . . .	35

2.4	Der Softwarearchitekturentwurf aus der Vogelperspektive . . . . .	36
2.4.1	Ziele und Aufgaben des Softwarearchitekturentwurfs . . . .	37
2.4.2	Der Softwarearchitekturentwurf im Überblick . . . . .	38
2.4.3	Wechselspiel der Tätigkeiten und Abstraktionsstufen im Entwurf . . . . .	40
2.4.4	Aufgaben des Softwarearchitekten und Bezug zu anderen Rollen . . . . .	43
2.5	Lernkontrolle . . . . .	44
<b>3</b>	<b>Entwurf von Softwarearchitekturen</b>	<b>47</b>
3.1	Einbettung in den iSAQB-Lehrplan . . . . .	48
3.1.1	Lernziele . . . . .	48
3.2	Überblick über das Vorgehen beim Architekturentwurf . . . . .	48
3.3	Entwurfsprinzipien und Heuristiken . . . . .	55
3.3.1	Top-down und Bottom-up . . . . .	55
3.3.2	Hierarchische (De-)Komposition . . . . .	56
3.3.2.1	Divide et impera . . . . .	56
3.3.2.2	Prinzipien bei der Zerlegung . . . . .	57
3.3.2.3	So-einfach-wie-möglich-Prinzip . . . . .	57
3.3.2.4	Trennung von Verantwortlichkeiten . . . . .	58
3.3.3	Schmale Schnittstellen und Information Hiding . . . . .	58
3.3.3.1	Information Hiding . . . . .	58
3.3.3.2	Verwendung von Schnittstellen . . . . .	59
3.3.4	Regelmäßiges Refactoring und Redesign . . . . .	59
3.4	Architekturzentrierte Entwicklungsansätze . . . . .	60
3.4.1	Domain Driven Design . . . . .	60
3.4.1.1	Fachmodelle als Basis . . . . .	60
3.4.1.2	Systematische Verwaltung der Domänenobjekte . . . . .	61
3.4.1.3	Strukturierung der Fachdomäne . . . . .	62
3.4.2	MDA . . . . .	62
3.4.3	Referenzarchitekturen . . . . .	63
3.4.3.1	Generative Erzeugung von Systembausteinen . . . .	63
3.4.3.2	Aspektorientierung . . . . .	64
3.4.3.3	Objektorientierung . . . . .	65
3.4.3.4	Prozedurale Ansätze . . . . .	66

3.5	Techniken für einen guten Entwurf .....	66
3.5.1	Ausgangssituation und Motivation: Degeneriertes Design .....	66
3.5.2	Lose Kopplung .....	67
3.5.3	Hohe Kohäsion .....	69
3.5.4	Offen-Geschlossen-Prinzip .....	70
3.5.5	Umkehr der Abhängigkeiten .....	70
3.5.6	Abtrennung von Schnittstellen .....	71
3.5.7	Zyklische Abhängigkeiten auflösen .....	72
3.5.8	Liskov'sches Substitutionsprinzip .....	72
3.6	Architekturmuster .....	73
3.6.1	Adaptierbare Systeme .....	74
3.6.1.1	Dependency Injection .....	74
3.6.2	Interaktive Systeme .....	75
3.6.2.1	Model View Controller .....	75
3.6.2.2	Model View Presenter .....	75
3.6.2.3	Presentation Abstraction Control .....	77
3.6.3	Vom Chaos zur Struktur .....	78
3.6.3.1	Schichtenarchitektur .....	78
3.6.3.2	Pipes and Filters .....	79
3.6.3.3	Blackboard .....	79
3.6.4	Verteilte Systeme .....	80
3.6.4.1	Broker .....	80
3.6.4.2	Serviceorientierung .....	81
3.6.4.3	Modularisierung .....	82
3.7	Entwurfsmuster .....	83
3.7.1	Adapter .....	83
3.7.2	Observer .....	84
3.7.3	Decorator .....	85
3.7.4	Proxy .....	85
3.7.5	Fassade .....	86
3.7.6	Brücke .....	87
3.7.7	State .....	87
3.7.8	Mediator .....	88
3.8	Lernkontrolle .....	89

<b>4</b>	<b>Beschreibung und Kommunikation von Softwarearchitekturen</b>	<b>93</b>
4.1	Einbettung in den iSAQB-Lehrplan .....	93
4.1.1	Lernziele .....	94
4.2	Das CoCoME-Beispiel .....	94
4.2.1	Anwendungsfälle im CoCoME-System .....	95
4.2.2	Übersicht über den strukturellen Aufbau des CoCoME-Systems .....	96
4.3	Sichten und Schablonen .....	97
4.3.1	Bewährte Sichten nach iSAQB .....	97
4.3.2	UML-Diagramme als Notationsmittel in Sichtenbeschreibungen .....	99
4.3.3	Sichtenbeschreibung – Grobaufbau und Einführungsbeispiel .....	102
4.3.3.1	Grobaufbau – Schablonenartige Sichtenbeschreibung .....	102
4.3.3.2	Beispiel: Auszug aus einer Sichten- beschreibung für eine Bausteinsicht .....	104
4.3.4	Kontextsicht oder Kontextabgrenzung .....	106
4.3.5	Bausteinsicht .....	110
4.3.6	Laufzeitsicht .....	113
4.3.7	Verteilungssicht bzw. Infrastruktursicht .....	117
4.3.8	Wechselwirkungen zwischen Architektursichten .....	121
4.3.9	Hierarchische Verfeinerung von Architektursichten .....	121
4.4	Technische oder querschnittliche Konzepte in Softwarearchitekturen .....	124
4.4.1	Technische bzw. querschnittliche Konzepte: Beispieldimensionen .....	125
4.4.2	Beispiel: Fehlerbehandlung .....	125
4.4.3	Beispiel: Sicherheit .....	126
4.5	Architektur und Implementierung .....	127
4.5.1	Beispiel Implementierung .....	128
4.6	Übliche Dokumenttypen für Softwarearchitekturen .....	129
4.6.1	Zentrale Architekturbeschreibung .....	129
4.6.2	Architekturüberblick .....	130
4.6.3	Dokumentübersicht .....	130
4.6.4	Übersichtspräsentation .....	130
4.6.5	»Architekturtapete« .....	130

4.6.6	Handbuch zur Dokumentation .....	131
4.6.7	Technische Informationen .....	131
4.6.8	Dokumentation von externen Schnittstellen .....	131
4.6.9	Template .....	131
4.7	Praxisregeln zur Dokumentation .....	132
4.7.1	Regel 1: »Schreiben aus der Sicht des Lesers« .....	132
4.7.2	Regel 2: »Unnötige Wiederholung vermeiden« .....	133
4.7.3	Regel 3: »Mehrdeutigkeit vermeiden« .....	133
4.7.4	Regel 4: »Standardisierte Organisationsstruktur bzw. Schablonen« .....	133
4.7.5	Regel 5: »Begründen Sie wesentliche Entscheidungen schriftlich« .....	134
4.7.6	Regel 6: »Überprüfung auf Gebrauchstauglichkeit« ....	134
4.7.7	Regel 7: »Übersichtliche Diagramme« .....	134
4.7.8	Regel 8: »Regelmäßige Aktualisierungen« .....	135
4.8	Beispiele weiterer Architektur-Frameworks .....	135
4.8.1	4+1-Framework .....	136
4.8.2	RM-ODP .....	136
4.8.3	SAGA .....	138
4.9	Lernkontrolle .....	138
<b>5</b>	<b>Softwarearchitekturen und Qualität</b>	<b>141</b>
5.1	Einbettung in den iSAQB-Lehrplan .....	142
5.1.1	Lernziele .....	142
5.2	Bewertung von Softwarearchitekturen .....	143
5.2.1	Qualitative Bewertung .....	143
5.2.1.1	DIN ISO/IEC 25010 .....	143
5.2.1.2	Qualitätsmerkmale .....	143
5.2.1.3	Weitere Qualitätsmerkmale .....	145
5.2.1.4	Auswirkungen bestimmter Qualitätsmerkmale .....	146
5.2.1.5	Taktiken und Praktiken .....	146
5.2.2	Quantitative Bewertung .....	148
5.2.2.1	Überprüfung von Architekturregeln .....	148
5.2.2.2	Metriken .....	149
5.2.2.3	Zyklomatische Komplexität .....	150

5.3	Prototyp und technischer Durchstich .....	150
5.3.1	Technischer Durchstich .....	151
5.3.2	Prototyp .....	151
5.3.2.1	Einsatz von Softwareprototypen .....	151
5.3.2.2	Arten von Softwareprototypen .....	152
5.4	Architekturanalyse .....	152
5.4.1	ATAM-Methode .....	152
5.5	Lernkontrolle .....	159
<b>6</b>	<b>Werkzeuge für Softwarearchitekten</b>	<b>161</b>
6.1	Einbettung in den iSAQB-Lehrplan .....	161
6.1.1	Lernziele .....	161
6.2	Allgemeine Hinweise zu Werkzeugen .....	161
6.2.1	Kosten von Werkzeugen .....	162
6.2.2	Lizenzen und Lizenzbedingungen .....	162
6.3	Werkzeuge zum Anforderungsmanagement .....	163
6.3.1	Anforderungen und Entscheidungskriterien .....	163
6.3.2	Herausforderungen von Werkzeugen für das Anforderungsmanagement .....	163
6.3.3	Beispielhafte Vertreter .....	163
6.4	Werkzeuge zur Modellierung .....	164
6.4.1	Anforderungen und Entscheidungskriterien .....	164
6.4.2	Herausforderungen von Werkzeugen für die Modellierung .....	164
6.4.3	Beispielhafte Vertreter .....	165
6.5	Werkzeuge zur Generierung .....	165
6.5.1	Anforderungen und Entscheidungskriterien .....	165
6.5.2	Herausforderungen von Codegeneratoren .....	165
6.5.3	Beispielhafte Vertreter .....	166
6.6	Werkzeuge zur statischen Codeanalyse .....	166
6.6.1	Anforderungen und Entscheidungskriterien .....	166
6.6.2	Herausforderungen von Werkzeugen zur statischen Codeanalyse .....	166
6.6.3	Beispielhafte Vertreter .....	167

6.7	Werkzeuge zur dynamischen Analyse .....	167
6.7.1	Anforderungen und Entscheidungskriterien .....	167
6.7.2	Herausforderungen von Werkzeugen zur dynamischen Analyse .....	167
6.7.3	Beispielhafte Vertreter .....	168
6.8	Werkzeuge zum Build-Management .....	168
6.8.1	Anforderungen und Entscheidungskriterien .....	168
6.8.2	Herausforderungen von Werkzeugen zum Build-Management .....	168
6.8.3	Beispielhafte Vertreter .....	169
6.9	Werkzeuge zum Konfigurations- und Versionsmanagement .....	169
6.9.1	Anforderungen und Entscheidungskriterien .....	169
6.9.2	Herausforderungen von Werkzeugen zum Konfigurations- und Versionsmanagement .....	170
6.9.3	Beispielhafte Vertreter .....	170
6.10	Werkzeuge zum Codemanagement .....	170
6.10.1	Herausforderungen von Werkzeugen zum Codemanagement .....	171
6.10.2	Beispielhafte Vertreter .....	171
6.11	Werkzeuge zum Test .....	171
6.11.1	Anforderungen und Entscheidungskriterien .....	171
6.11.2	Herausforderungen von Testwerkzeugen .....	172
6.11.3	Beispielhafte Vertreter .....	172
6.12	Werkzeuge zur Dokumentation .....	172
6.12.1	Anforderungen und Entscheidungskriterien .....	172
6.12.2	Herausforderungen von Dokumentationswerkzeugen ..	172
6.12.3	Beispielhafte Vertreter .....	173
6.13	Lernkontrolle .....	173

## Anhang

<b>A</b>	<b>Beispielfragen</b>	<b>177</b>
A.1	Auszüge aus der Prüfungsordnung .....	177
A.2	Beispielfragen .....	178

<b>B</b>	<b>Abkürzungsverzeichnis</b>	<b>181</b>
<b>C</b>	<b>Glossar</b>	<b>183</b>
<b>D</b>	<b>Literaturverzeichnis</b>	<b>195</b>
	<b>Index</b>	<b>201</b>