

# Inhalt

<b>Geleitwort von Gernot Starke</b> .....	<b>XI</b>
<b>1 Warum Softwarearchitekturen dokumentieren?</b> .....	<b>1</b>
1.1 Montagmorgen .....	1
1.1.1 Fragen über Fragen .....	1
1.1.2 Wer fragt, bekommt Antworten .....	2
1.2 Voll unagil? .....	4
1.2.1 Agil vorgehen .....	5
1.2.2 Funktionierende Software vor umfassender Dokumentation .....	6
1.2.3 Dokumentation unterstützt Kommunikation .....	7
1.3 Wirkungsvolle Architekturdokumentation .....	7
1.3.1 Ziel 1: Architekturarbeit unterstützen .....	8
1.3.2 Ziel 2: Architektur nachvollziehbar und bewertbar machen .....	8
1.3.3 Ziel 3: Umsetzung und Weiterentwicklung leiten .....	9
1.3.4 Fremdwort Do ku men ta tion [...zion] [lat.] .....	9
1.4 Mission Statement für dieses Buch .....	10
1.5 Über dieses Buch .....	11
1.5.1 Für wen ich dieses Buch geschrieben habe .....	11
1.5.2 Wie dieses Buch aufgebaut ist .....	12
1.5.3 Wem ich Dankeschön sagen möchte .....	16
<b>2 Was Softwarearchitektur ist und worauf sie aufbaut</b> .....	<b>17</b>
2.1 Softwarearchitektur-Freischwimmer .....	17
2.1.1 Was ist Softwarearchitektur? .....	17
2.1.2 Wie entsteht Softwarearchitektur? .....	18
2.1.3 Wer oder was ist ein Softwarearchitekt? .....	21
2.1.4 Ein Architekturüberblick auf n Seiten, n < 30 .....	23
2.2 Die Zielsetzung vermitteln .....	23
2.2.1 Jetzt kommt ein Karton! .....	23
2.2.2 Virtueller Produktkarton (Dokumentationsmittel) .....	24
2.2.3 Fallbeispiel: Schach-Engine „DokChess“ .....	25
2.2.4 Tipps zum Erstellen von Produktkartons .....	26
2.2.5 Fallbeispiel: Schachplattform „immer-nur-schach.de“ .....	27
2.3 Den Kontext abgrenzen .....	28
2.3.1 Systemkontext (Dokumentationsmittel) .....	29
2.3.2 Fallbeispiel: Systemkontext „immer-nur-schach.de“ .....	30
2.3.3 Tipps zur Erstellung des Systemkontextes .....	31

2.4	Im Rahmen bleiben. . . . .	36
2.4.1	Warum Randbedingungen festhalten? . . . . .	36
2.4.2	Randbedingungen (Dokumentationsmittel) . . . . .	38
2.4.3	Fallbeispiel: Randbedingungen „immer-nur-schach.de“. . . . .	38
2.4.4	Tipps zum Festhalten von Randbedingungen . . . . .	39
2.5	Geforderte Qualitätsmerkmale . . . . .	41
2.5.1	Was sind Qualitätsmerkmale? . . . . .	42
2.5.2	Qualitätsziele (Dokumentationsmittel) . . . . .	43
2.5.3	Fallbeispiel: Qualitätsziele „immer-nur-schach.de“ . . . . .	44
2.5.4	Fallbeispiel: Qualitätsziele „DokChess“ . . . . .	44
2.5.5	Qualitätsmerkmale genauer beschreiben . . . . .	46
2.5.6	Qualitätsszenarien (Dokumentationsmittel) . . . . .	47
2.5.7	Fallbeispiel: Qualitätsszenarien „immer-nur-schach.de“ . . . . .	48
2.5.8	Tipps zum Festhalten von Qualitätsszenarien . . . . .	50
2.6	Weitere Einflüsse und Hilfsmittel . . . . .	53
2.6.1	Stakeholder . . . . .	53
2.6.2	Persona (Dokumentationsmittel) . . . . .	54
2.6.3	Fallbeispiel: Persona „immer-nur-schach.de“ . . . . .	55
2.6.4	Risiken. . . . .	56
2.6.5	Technische Risiken (Dokumentationsmittel). . . . .	57
2.6.6	Fallbeispiel: Technische Risiken „DokChess“ . . . . .	57
2.6.7	Glossar (Dokumentationsmittel). . . . .	59
<b>3</b>	<b>Entscheidungen treffen und festhalten. . . . .</b>	<b>61</b>
3.1	Historisch gewachsen? . . . . .	61
3.2	Architekturentscheidungen. . . . .	62
3.2.1	Architekturentscheidung (Dokumentationsmittel). . . . .	62
3.2.2	Fallbeispiel: Spannende Fragen „DokChess“. . . . .	64
3.2.3	Tipps zur Formulierung von Fragestellungen. . . . .	64
3.2.4	Fallbeispiel: Fragestellungen „immer-nur-schach.de“ . . . . .	66
3.3	Einflussfaktoren auf Entscheidungen. . . . .	70
3.3.1	Den Überblick behalten . . . . .	70
3.3.2	Kreuztabellen . . . . .	71
3.3.3	Fallbeispiel: Einflüsse „immer-nur-schach.de“ . . . . .	72
3.3.4	Tipps zur Anfertigung von Kreuztabellen . . . . .	73
<b>4</b>	<b>Plädoyer für eine feste Gliederung. . . . .</b>	<b>75</b>
4.1	Der jüngste Spieler fängt an! . . . . .	75
4.2	Vorteile einer festen Struktur . . . . .	76
4.3	arc42 – Vorschlag für eine Gliederung. . . . .	78
4.3.1	Was ist arc42? . . . . .	78
4.3.2	Die Struktur der arc42-Vorlage . . . . .	79
4.3.3	Wo funktioniert arc42 besonders gut? . . . . .	81
4.3.4	arc42 in diesem Buch. . . . .	82
4.4	Alternativen zu arc42. . . . .	83
4.4.1	Standards zur Architekturbeschreibung . . . . .	83

4.4.2	Vorgehensmodelle	84
4.4.3	Architektur-Frameworks	86
4.4.4	Fachliteratur als Inspiration?	87
<b>5</b>	<b>Sichten auf Softwarearchitektur</b>	<b>89</b>
5.1	Strukturen entwerfen und festhalten	89
5.1.1	Was ist was? Band 127: Unser Softwaresystem.	89
5.1.2	Schritte der Zerlegung dokumentieren	90
5.1.3	Bausteinsicht (Dokumentationsmittel)	91
5.1.4	Fallbeispiel: Bausteinsicht „DokChess“ (Ausschnitt)	92
5.1.5	Komponenten: Babylonische Sprachverwirrung 2.0.	93
5.1.6	Tipps zur Erstellung der Bausteinsicht	94
5.1.7	Interaktionspunkte beschreiben.	99
5.1.8	Schnittstellenbeschreibung (Dokumentationsmittel)	102
5.1.9	Fallbeispiel: Schnittstellen der Eröffnung in „DokChess“	104
5.2	Verschiedene Blickwinkel	107
5.2.1	Hat Mozart modelliert?	107
5.2.2	Fachliche Zerlegung vs. technische Zerlegung	109
5.2.3	Fallbeispiel: Bausteinsicht „immer-nur-schach.de“	111
5.3	Verhalten und Abläufe beschreiben	114
5.3.1	Abläufe in Entwurf und Dokumentation	114
5.3.2	Darstellungen für Abläufe	114
5.3.3	Laufzeitsicht (Dokumentationsmittel)	117
5.3.4	Fallbeispiel: Ablauf in DokChess	118
5.3.5	Fallbeispiel: Zustandsautomat XBoard (DokChess)	118
5.4	Die Dinge zum Einsatz bringen.	119
5.4.1	Betriebsaspekte in der Architekturdokumentation	120
5.4.2	Darstellungen für Verteilung.	121
5.4.3	Verteilungssicht (Dokumentationsmittel)	123
5.4.4	Fallbeispiel: „immer-nur-schach.de“	125
5.5	Alternative Vorschläge für Sichten	126
5.6	Muster kommunizieren	129
5.6.1	Muster in der Softwareentwicklung.	129
5.6.2	Wann sollte man Muster dokumentieren (und wo)?	130
5.6.3	Einsatz von Mustern dokumentieren.	130
5.6.4	Fallbeispiel: DokChess	132
<b>6</b>	<b>Übergreifende Konzepte</b>	<b>133</b>
6.1	Warum übergreifende Themen?	133
6.2	Themen und Lösungsoptionen	134
6.2.1	Mögliche Themen für übergreifende Konzepte.	134
6.2.2	Typische Lösungsoptionen	136
6.3	Themenauswahl	138
6.3.1	Wie wählt man Themen für die Dokumentation aus?	139
6.3.2	Fallbeispiel: Übergreifende Themen „DokChess“	140
6.4	Eine Gliederungstechnik für Konzepte.	142

6.4.1	Werkzeug: Warum? Was? Wie? Wohin noch?	142
6.4.2	Gliederung für ein Konzept	144
6.4.3	Informeller Text für den Architekturüberblick	146
6.5	Tipps zur Erstellung übergreifender Konzepte	147
<b>7</b>	<b>Werkzeuge zur Dokumentation</b>	<b>151</b>
7.1	Notationen passgenau wählen	151
7.2	Toolparade zur Architekturdokumentation	156
7.2.1	Erstellung und Pflege	156
7.2.2	Verwaltung von Inhalten	162
7.2.3	Kommunikation von Lösungen	164
7.3	Repository: UML vs. Wiki	166
7.3.1	Steht alles im Wiki?	167
7.3.2	Steht alles im UML-Tool?	170
7.3.3	UML-Tool + Wiki == Traumpaar?	173
7.4	Wie auswählen?	174
<b>8</b>	<b>Lightfäden für das Vorgehen zur Dokumentation</b>	<b>177</b>
8.1	Während der Entwicklung dokumentieren	177
8.1.1	Zielgruppen Ihrer Dokumentation	177
8.1.2	Dokumentationsmittel und Dokumente	180
8.1.3	Womit anfangen?	183
8.1.4	Während der Arbeit: Kommunizieren und Pflegen	184
8.2	Der Softwaredetektiv: Bestehendes Dokumentieren	186
8.2.1	Auslöser für Dokumentationsbedarf	186
8.2.2	Mögliche Szenarien und Ziele des Dokumentierens im Nachhinein	187
8.2.3	Sherlock Holmes vs. Die drei ???	188
8.2.4	Informationsquellen identifizieren	189
8.2.5	Dokumentationsmittel unter der Lupe	190
8.2.6	Exkurs: Werkzeuge zur Rekonstruktion	195
8.3	Variationen von „Ein System“	200
8.3.1	Dokumentation von Systemlandschaften	200
8.3.2	Dokumentation von Frameworks und Blue Prints	202
<b>9</b>	<b>Architekturüberblick DokChess</b>	<b>205</b>
9.1	Einführung und Ziele	205
9.1.1	Aufgabenstellung	205
9.1.2	Qualitätsziele	206
9.1.3	Stakeholder	206
9.2	Randbedingungen	208
9.2.1	Technische Randbedingungen	208
9.2.2	Organisatorische Randbedingungen	209
9.2.3	Konventionen	210
9.3	Kontextabgrenzung	210
9.3.1	Fachlicher Kontext	210
9.3.2	Technischer- oder Verteilungskontext	211
9.4	Lösungsstrategie	212

9.4.1	Aufbau der Engine .....	212
9.4.2	Spielstrategie .....	213
9.4.3	Die Anbindung .....	213
9.5	Bausteinsicht .....	214
9.5.1	Ebene 1 .....	214
9.5.2	XBoard-Protokoll (Blackbox) .....	215
9.5.3	Spielregeln (Blackbox) .....	216
9.5.4	Engine (Blackbox) .....	217
9.5.5	Eröffnung (Blackbox) .....	218
9.5.6	Ebene 2: Engine (Whitebox) .....	219
9.5.7	Zügauswahl (Blackbox) .....	220
9.5.8	Stellungsbewertung (Blackbox) .....	221
9.6	Laufzeitsicht .....	222
9.6.1	Zugermittlung Walkthrough .....	222
9.7	Verteilungssicht .....	223
9.7.1	Infrastruktur Windows .....	223
9.8	Konzepte .....	225
9.8.1	Abhängigkeiten zwischen Bausteinen .....	225
9.8.2	Schach-Domänenmodell .....	225
9.8.3	Benutzungsoberfläche .....	227
9.8.4	Plausibilisierung und Validierung .....	228
9.8.5	Ausnahme- und Fehlerbehandlung .....	229
9.8.6	Logging, Protokollierung, Tracing .....	229
9.8.7	Testbarkeit .....	230
9.9	Entwurfsentscheidungen .....	232
9.9.1	Wie kommuniziert die Engine mit der Außenwelt? .....	232
9.9.2	Sind Stellungsobjekte veränderlich oder nicht? .....	233
9.10	Qualitätsszenarien .....	236
9.10.1	Qualitätsbaum .....	236
9.10.2	Bewertungsszenarien .....	237
9.11	Risiken .....	238
9.11.1	Risiko: Anbindung an das Frontend .....	238
9.11.2	Risiko: Aufwand der Implementierung .....	238
9.11.3	Risiko: Erreichen der Spielstärke .....	239
9.12	Glossar .....	239
<b>10</b>	<b>Stolpersteine der Architekturdokumentation .....</b>	<b>241</b>
10.1	Probleme .....	241
10.2	Fiese Fallen ... ..	243
10.3	... und wie man sie umgeht oder entschärft .....	245
10.4	Reviews von Architekturdokumentation .....	246
	<b>Glossar .....</b>	<b>253</b>
	<b>Literaturverzeichnis .....</b>	<b>257</b>
	<b>Stichwortverzeichnis .....</b>	<b>261</b>