

SOFTWARE- ARCHITEKTUREN DOKUMENTIEREN UND KOMMUNIZIEREN

Entwürfe, Entscheidungen und
Lösungen nachvollziehbar
und wirkungsvoll festhalten



2. Auflage

stefan ZÖRNER

HANSER

Mit einem Geleitwort von Gernot Starke

Zörner

Softwarearchitekturen dokumentieren und kommunizieren

Bleiben Sie auf dem Laufenden!



Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter



www.hanser-fachbuch.de/newsletter



Hanser Update ist der IT-Blog des Hanser Verlags mit Beiträgen und Praxistipps von unseren Autoren rund um die Themen Online Marketing, Webentwicklung, Programmierung, Softwareentwicklung sowie IT- und Projektmanagement. Lesen Sie mit und abonnieren Sie unsere News unter



www.hanser-fachbuch.de/update



Stefan Zörner

Softwarearchitekturen dokumentieren und kommunizieren

Entwürfe, Entscheidungen
und Lösungen
nachvollziehbar und wirkungsvoll
festhalten

2., überarbeitete und erweiterte Auflage

HANSER

Der Autor:

Stefan Zörner, Buchholz in der Nordheide

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2015 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Brigitte Bauer-Schiewek

Herstellung: Irene Weilhart

Copy editing: Petra Kienle, Fürstenfeldbruck

Umschlagdesign: Marc Müller-Bremer, www.rebranding.de, München

Umschlagrealisation: Stephan Rönigk

Foto des Autors: Jan Gentsch, www.jangentsch.de, Hamburg

Layout: Manuela Treindl, Fürth

Druck und Bindung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-44348-8

E-Book-ISBN: 978-3-446-44442-3

Inhalt

Geleitwort zur 1. Auflage	XI
Überblick: Dokumentationsmittel im Buch	XIII
1 Warum Softwarearchitekturen dokumentieren?	1
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
2 Was Softwarearchitektur ist und worauf sie aufbaut	17
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	57
2.6.5	Technische Risiken (Dokumentationsmittel)	58
2.6.6	Fallbeispiel: Technische Risiken „DokChess“	58
2.6.7	Glossar (Dokumentationsmittel)	59
3	Entscheidungen treffen und festhalten	61
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
3.4	Kompakte Darstellung der Lösungsstrategie	74
3.4.1	Softwarearchitektur auf einem Bierdeckel?	75
3.4.2	Lösungsstrategie (Dokumentationsmittel)	75
3.4.3	Fallbeispiel: Lösungsstrategie „DokChess“	77
3.4.4	Als Ergänzung: ein Überblicksbild	78
3.4.5	Eine Architekturbewertung auf dem Bierdeckel	78

4	Plädoyer für eine feste Gliederung	79
4.1	Der jüngste Spieler fängt an!	79
4.2	Vorteile einer festen Struktur	80
4.3	arc42 – Vorschlag für eine Gliederung	82
4.3.1	Was ist arc42?	82
4.3.2	Die Struktur der arc42-Vorlage	83
4.3.3	Wo funktioniert arc42 besonders gut?	85
4.3.4	arc42 in diesem Buch	86
4.4	Alternativen zu arc42	87
4.4.1	Standards zur Architekturbeschreibung	87
4.4.2	Vorgehensmodelle	88
4.4.3	Architektur-Frameworks	90
4.4.4	Fachliteratur als Inspiration?	91
5	Sichten auf Softwarearchitektur	95
5.1	Strukturen entwerfen und festhalten	95
5.1.1	Was ist was? Band 127: Unser Softwaresystem	95
5.1.2	Schritte der Zerlegung dokumentieren	96
5.1.3	Bausteinsicht (Dokumentationsmittel)	97
5.1.4	Fallbeispiel: Bausteinsicht „DokChess“ (Ausschnitt)	98
5.1.5	Komponenten: Babylonische Sprachverwirrung 2.0	99
5.1.6	Tipps zur Erstellung der Bausteinsicht	100
5.1.7	Interaktionspunkte beschreiben	105
5.1.8	Schnittstellenbeschreibung (Dokumentationsmittel)	108
5.1.9	Fallbeispiel: Schnittstellen der Eröffnung in „DokChess“	110
5.2	Verschiedene Blickwinkel	113
5.2.1	Hat Mozart modelliert?	113
5.2.2	Fachliche Zerlegung vs. technische Zerlegung	115
5.2.3	Fallbeispiel: Bausteinsicht „immer-nur-schach.de“	117
5.3	Verhalten und Abläufe beschreiben	120
5.3.1	Abläufe in Entwurf und Dokumentation	120
5.3.2	Darstellungen für Abläufe	120
5.3.3	Laufzeitsicht (Dokumentationsmittel)	123
5.3.4	Fallbeispiel: Ablauf in DokChess	124
5.3.5	Fallbeispiel: Zustandsautomat XBoard (DokChess)	124
5.4	Die Dinge zum Einsatz bringen	125
5.4.1	Betriebsaspekte in der Architekturdokumentation	126
5.4.2	Darstellungen für Verteilung	127
5.4.3	Verteilungssicht (Dokumentationsmittel)	129
5.4.4	Fallbeispiel: „immer-nur-schach.de“	131
5.5	Alternative Vorschläge für Sichten	132
5.6	Muster kommunizieren	135
5.6.1	Muster in der Softwareentwicklung	135
5.6.2	Wann sollte man Muster dokumentieren (und wo)?	136
5.6.3	Einsatz von Mustern dokumentieren	136
5.6.4	Fallbeispiel: DokChess	138

6	Übergreifende Konzepte	139
6.1	Warum übergreifende Themen?	139
6.2	Themen und Lösungsoptionen.	140
6.2.1	Mögliche Themen für übergreifende Konzepte	140
6.2.2	Typische Lösungsoptionen	142
6.3	Themenauswahl	144
6.3.1	Wie wählt man Themen für die Dokumentation aus?	145
6.3.2	Fallbeispiel: Übergreifende Themen „DokChess“	146
6.4	Eine Gliederungstechnik für Konzepte	148
6.4.1	Werkzeug: Warum? Was? Wie? Wohin noch?	148
6.4.2	Gliederung für ein Konzept	150
6.4.3	Informeller Text für den Architekturüberblick	152
6.5	Tipps zur Erstellung übergreifender Konzepte	153
7	Werkzeuge zur Dokumentation	157
7.1	Notationen passgenau wählen	157
7.2	Toolparade zur Architekturdokumentation	162
7.2.1	Erstellung und Pflege	162
7.2.2	Verwaltung von Inhalten	168
7.2.3	Kommunikation von Lösungen	170
7.3	Repository: UML vs. Wiki	172
7.3.1	Steht alles im Wiki?	173
7.3.2	Steht alles im UML-Tool?	176
7.3.3	UML-Tool + Wiki == Traumpaar?	179
7.4	Wie auswählen?	180
8	Lightfäden für das Vorgehen zur Dokumentation	183
8.1	Während der Entwicklung dokumentieren	183
8.1.1	Zielgruppen Ihrer Dokumentation	183
8.1.2	Dokumentationsmittel und Dokumente	186
8.1.3	Womit anfangen?	189
8.1.4	Während der Arbeit: Kommunizieren und Pflegen	190
8.2	Der Softwaredetektiv: Bestehendes Dokumentieren	192
8.2.1	Auslöser für Dokumentationsbedarf	192
8.2.2	Mögliche Szenarien und Ziele des Dokumentierens im Nachhinein	193
8.2.3	Sherlock Holmes vs. Die drei ???	194
8.2.4	Informationsquellen identifizieren	195
8.2.5	Dokumentationsmittel unter der Lupe	196
8.2.6	Exkurs: Werkzeuge zur Rekonstruktion	201
8.3	Variationen von „Ein System“	206
8.3.1	Dokumentation von Systemlandschaften	206
8.3.2	Dokumentation von Frameworks und Blue Prints	208
9	Architekturüberblick DokChess	211
9.1	Einführung und Ziele	212
9.1.1	Aufgabenstellung	212

9.1.2	Qualitätsziele	212
9.1.3	Stakeholder	213
9.2	Randbedingungen	216
9.2.1	Technische Randbedingungen	216
9.2.2	Organisatorische Randbedingungen	216
9.2.3	Konventionen	217
9.3	Kontextabgrenzung	218
9.3.1	Fachlicher Kontext	218
9.3.2	Technischer- oder Verteilungskontext	219
9.4	Lösungsstrategie	220
9.4.1	Aufbau von DokChess	221
9.4.2	Spielstrategie	222
9.4.3	Die Anbindung	222
9.5	Bausteinsicht	223
9.5.1	Ebene 1	223
9.5.2	XBoard-Protokoll (Blackbox)	224
9.5.3	Spielregeln (Blackbox)	225
9.5.4	Engine (Blackbox)	226
9.5.5	Eröffnung (Blackbox)	227
9.5.6	Ebene 2: Engine (Whitebox)	229
9.5.7	Zugsuche (Blackbox)	229
9.5.8	Stellungsbewertung (Blackbox)	231
9.6	Laufzeitsicht	232
9.6.1	Zugermittlung Walkthrough	232
9.7	Verteilungssicht	233
9.7.1	Infrastruktur Windows	233
9.8	Konzepte	235
9.8.1	Abhängigkeiten zwischen Modulen	235
9.8.2	Schach-Domänenmodell	235
9.8.3	Benutzungsoberfläche	237
9.8.4	Plausibilisierung und Validierung	238
9.8.5	Ausnahme- und Fehlerbehandlung	239
9.8.6	Logging, Protokollierung, Tracing	239
9.8.7	Testbarkeit	240
9.9	Entwurfsentscheidungen	242
9.9.1	Wie kommuniziert die Engine mit der Außenwelt?	242
9.9.2	Sind Stellungsobjekte veränderlich oder nicht?	244
9.10	Qualitätsszenarien	247
9.10.1	Qualitätsbaum	247
9.10.2	Bewertungsszenarien	248
9.11	Risiken	249
9.11.1	Risiko: Anbindung an das Frontend	249
9.11.2	Risiko: Aufwand der Implementierung	249
9.11.3	Risiko: Erreichen der Spielstärke	250
9.12	Glossar	251

10 Stolpersteine der Architekturdokumentation	253
10.1 Probleme	253
10.2 Fiese Fallen	255
10.3 ... und wie man sie umgeht oder entschärft	257
10.4 Reviews von Architekturdokumentation	258
Glossar	265
Literaturverzeichnis	269
Stichwortverzeichnis	273

Geleitwort zur 1. Auflage

Dokumentation – Unwort der IT?

Viele IT-Systeme gelten zu Recht als schlecht erweiterbar, schwer verständlich und unheimlich komplex. Teilweise liegt das an ihrer mangelhaften Dokumentation, an fehlenden oder unklaren Erläuterungen. Bei anderen Systemen begegnet mir das Gegenteil: Hunderte von Dokumenten, ungeordnet auf Netzlaufwerken, ohne klaren Einstiegspunkt. Kein Wunder, dass Dokumentation als Unwort gilt.

Die meisten Teams, die ich in meinem IT-Leben begleitet habe, konnten gut bis sehr gut programmieren, viele haben ausgezeichnete technische Konzepte entwickelt und umgesetzt. Aber kaum eines dieser Teams konnte auch nur halbwegs ordentlich dokumentieren. Oft als lästige Nebensache verflucht, mit fadenscheinigen Argumenten auf „später“ verschoben oder von Anfang an aufs Abstellgleis verbannt: Dokumentation gilt in Projekten als uncool oder, schlimmer noch, als Strafarbeit: Doku – das sollen andere machen.

Hinter dieser weit verbreiteten, negativen Haltung steckt Unsicherheit: Kaum ein Entwickler, Architekt oder Projektleiter hat jemals gelernt, über Systeme zielorientiert, methodisch und mit moderatem Aufwand zu kommunizieren – und Dokumentation ist schriftliche (d. h. persistente) Kommunikation.

Genau dafür stellt dieses Buch großartige, praxiserprobte und direkt umsetzbare Lösungen bereit: Sie erfahren, wie Sie mit einfachen Mitteln die Anforderungen an langfristige, lesbare und verständliche Dokumentation erfüllen können. Stefan Zörner erklärt Ihnen, wie Sie sowohl den großen Überblick als auch das notwendige kleine Detail für Ihre Leser sachgerecht aufbereiten und darstellen. Besonders freut mich natürlich, dass er etwas Werbung für unser (freies) arc42-Template macht :-)

Ein echtes Novum finden Sie in Kapitel 6 über technische Konzepte: Überall heißt es in der Praxis: „Wir brauchen ein Konzept für <schwieriges technisches Problem>“ ... aber niemand erklärt uns, wie solche Konzepte denn genau aussehen sollen. Wir überlegen jedes Mal neu, in welcher Struktur wir unsere Lösungsideen darstellen und wie wir argumentieren sollen. Stefan eilt mit diesem Buch zu Hilfe: Er hat (unterstützt durch Uwe Vigenschow) aus den langjährigen Erfahrungen von Lernmethodikern und Hirnforschern genau die Hinweise extrahiert, die wir für verständliche, nachvollziehbare und klare Konzepte benötigen. (Neugierig geworden? Blättern Sie direkt mal zu Seite 148 und überfliegen das Vier-Quadranten-Modell.)

Aber damit nicht genug: Getreu dem Motto, dass Beispiele die besten Lehrmeister sind, hat Stefan ein wirklich cooles System entworfen, gebaut und für dieses Buch vorbildlich dokumentiert: Seine Schach-Engine DokChess illustriert, wie gute Dokumentation aussehen kann (und spielt außerdem noch ganz passabel Schach).

Ich wünsche Ihnen Freude mit diesem Buch. Als Reviewer durfte ich ja schon vor längerer Zeit frühe Versionen testlesen. Mehr als einmal haben mir Stefans Ratschläge in konkreten Projektsituationen seitdem geholfen.

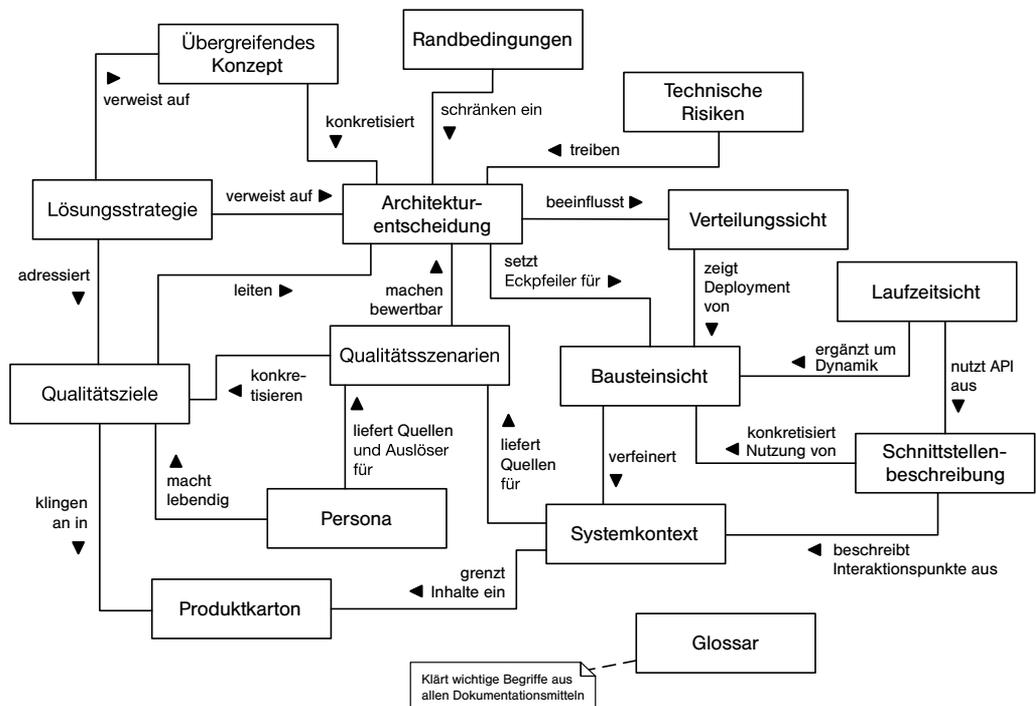
May the force of the proper word and diagram be with you.

Köln, im März 2012

Gernot Starke

Überblick: Dokumentationsmittel im Buch

Die Abbildung zeigt alle im Buch vorgestellten Dokumentationsmittel („Zutaten“) für Softwarearchitektur. Verbindungslinien visualisieren wichtige methodische Zusammenhänge. Die Pfeile an den Linien geben die Leserichtung für die Beschriftung an (Beispiel: Bausteinsicht verfeinert Systemkontext).



Dokumentationsmittel des Buchs mit wichtigen Zusammenhängen

Der Tabelle auf der nächsten Seite können Sie entnehmen, auf welcher Seite im Buch Sie den Steckbrief zum betreffenden Dokumentationsmittel finden.

Überblick über die Dokumentationsmittel

Dokumentationsmittel	Nutzen	Steckbrief
Architekturentscheidung	Nachvollziehbare Darstellung einer zentralen, risikoreichen Entscheidung	Seite 69
Bausteinsicht	Visualisierung der Struktur des Softwaresystems und wie die Teile voneinander abhängen	Seite 104
Glossar	Etablieren eines einheitlichen Wortschatzes im ganzen Vorhaben	Seite 60
Laufzeitsicht	Visualisierung von dynamischen Strukturen und Verhalten, vor allem von Abläufen	Seite 123
Lösungsstrategie	Stark verdichteter Architekturüberblick; Gegenüberstellung der wichtigsten Ziele und Lösungsansätze	Seite 76
Persona	Archetypische Beschreibung einer Personengruppe und deren Ziele (Stakeholder)	Seite 56
Produktkarton	Plakative Darstellung der wesentlichen Funktionen, Ziele und Merkmale des Systems	Seite 28
Qualitätsszenarien	Konkretisierung von Qualitätsanforderungen in kurzen, beispielhaften Sätzen	Seite 52
Qualitätsziele	Motivation der wichtigsten an das System gestellten Qualitätsanforderungen	Seite 45
Randbedingungen	Sammlung der technischen bzw. organisatorischen Vorgaben, die beim Entwurf einzuhalten sind (oder waren)	Seite 41
Schnittstellenbeschreibung	Detaillierte Beschreibung, wie ein Baustein Funktionalität bereitstellt (oder welche er benötigt)	Seite 112
Systemkontext	Visualisierung der Fremdsysteme und Benutzer, mit denen das System interagiert	Seite 34
Technische Risiken	Beschreibung der Risiken, die Einfluss auf die Softwarearchitektur haben (oder hatten)	Seite 59
Übergreifendes Konzept	Darstellung eines übergreifenden Themas, zur Vereinheitlichung im System oder zur Detaillierung eines Ansatzes	Seite 152
Verteilungssicht	Visualisierung der Zielumgebung, der Inbetriebnahme und des Betriebs des Systems	Seite 130