

Curriculum für  
CPSA Certified Professional for  
Software Architecture®

– Advanced Level –

**Modul:  
ADOK**

**Architekturdokumentation**



Version 1.6 (September 2018)

**© (Copyright), International Software Architecture Qualification Board e. V.  
(iSAQB® e. V.) 2012**

Die Nutzung des Lehrplans ist nur unter den nachfolgenden Voraussetzungen möglich:

1. Sie möchten das Zertifikat zum „CPSA Certified Professional for Software Architecture Advanced Level®“ erwerben. Für den Erwerb des Zertifikats ist es gestattet, die Text-Dokumente und/oder Lehrpläne zu nutzen, indem eine Arbeitskopie für den eigenen Rechner erstellt wird. Soll eine darüber hinausgehende Nutzung der Dokumente und/oder Lehrpläne erfolgen, zum Beispiel zur Weiterverbreitung an Dritte, Werbung etc., bitte unter [contact@isaqb.org](mailto:contact@isaqb.org) nachfragen. Sie müssten in diesem Fall einen Lizenzvertrag mit dem iSAQB e. V. schließen.
2. Sind Sie Trainer, Anbieter oder Trainingsorganisator, ist die Nutzung der Dokumente und/oder Lehrpläne nach Erwerb einer Nutzungslizenz möglich. Hierzu bitte unter [contact@isaqb.org](mailto:contact@isaqb.org) nachfragen. Lizenzverträge, die eine solche Nutzung regeln, sind vorhanden.
3. Falls Sie weder unter die Kategorie 1. noch unter die Kategorie 2. fallen, aber dennoch die Dokumente und/oder Lehrpläne nutzen möchten, nehmen Sie bitte ebenfalls Kontakt unter [contact@isaqb.org](mailto:contact@isaqb.org) zum iSAQB® e. V. auf. Sie werden dort über die Möglichkeit des Erwerbs entsprechender Lizenzen im Rahmen der vorhandenen Lizenzverträge informiert und können die gewünschten Nutzungsgenehmigungen erhalten.

Grundsätzlich weisen wir darauf hin, dass dieser Lehrplan urheberrechtlich geschützt ist. Alle Rechte an diesen Copyrights stehen ausschließlich dem International Software Architecture Qualification Board e. V. (iSAQB® e. V.) zu.

## Inhaltsverzeichnis

<b>0</b>	<b><u>  EINLEITUNG: ALLGEMEINES ZUM iSAQB-ADVANCED-LEVEL .....</u></b>	<b><u>5</u></b>
0.1	WAS VERMITTELT EIN ADVANCED-LEVEL-MODUL? .....	5
0.2	WAS KÖNNEN ABSOLVENTEN DES ADVANCED-LEVEL (CPSA-A)? .....	5
0.3	VORAUSSETZUNGEN ZUR CPSA-A-ZERTIFIZIERUNG .....	5
<b>1</b>	<b><u>  GRUNDLEGENDES ZUM MODUL ARCHITEKTURDOKUMENTATION.....</u></b>	<b><u>6</u></b>
1.1	GLIEDERUNG DES LEHRPLANS FÜR ARCHITEKTURDOKUMENTATION UND EMPFOHLENE ZEITLICHE AUFTEILUNG .....	6
1.2	DAUER, DIDAKTIK UND WEITERE DETAILS .....	6
1.3	VORAUSSETZUNGEN FÜR DAS MODUL „ARCHITEKTURDOKUMENTATION“ .....	6
1.4	GLIEDERUNG DES LEHRPLANS FÜR ARCHITEKTURDOKUMENTATION.....	7
1.5	ERGÄNZENDE INFORMATIONEN, BEGRIFFE, ÜBERSETZUNGEN .....	7
1.6	CREDIT POINTS FÜR DIESE SCHULUNG .....	7
<b>2</b>	<b><u>  EINFÜHRUNG IN DAS iSAQB-ZERTIFIZIERUNGSPROGRAMM.....</u></b>	<b><u>8</u></b>
2.1	BEGRIFFE UND KONZEPTE .....	8
2.2	LERNZIELE .....	8
<b>3</b>	<b><u>  GRUNDBEGRIFFE VON SOFTWAREARCHITEKTUR-DOKUMENTATION.....</u></b>	<b><u>9</u></b>
3.1	BEGRIFFE UND KONZEPTE .....	9
3.2	LERNZIELE .....	9
<b>4</b>	<b><u>  DOKUMENTATION UND VORGEHEN.....</u></b>	<b><u>10</u></b>
4.1	BEGRIFFE UND KONZEPTE .....	10
4.2	LERNZIELE .....	10
<b>5</b>	<b><u>  BESTANDTEILE VON ARCHITEKTURDOKUMENTATIONEN.....</u></b>	<b><u>11</u></b>
5.1	BEGRIFFE UND KONZEPTE .....	11
5.2	LERNZIELE .....	11
<b>6</b>	<b><u>  WERKZEUGE.....</u></b>	<b><u>13</u></b>
6.1	BEGRIFFE UND KONZEPTE .....	13

6.2	LERNZIELE .....	13
<u>7</u>	<u>DOKUMENTATION BEWERTEN.....</u>	<u>14</u>
7.1	BEGRIFFE UND KONZEPTE .....	14
7.2	LERNZIELE .....	14
<u>8</u>	<u>BEISPIELE FÜR DOKUMENTATION VON SOFTWARE-ARCHITEKTUREN .....</u>	<u>15</u>
8.1	BEGRIFFE UND KONZEPTE .....	15
8.2	LERNZIELE .....	15
8.3	REFERENZEN .....	15
<u>9</u>	<u>QUELLEN UND REFERENZEN ZU SOFTWAREARCHITEKTUR-DOKUMENTATION.....</u>	<u>16</u>

## **0 Einleitung: Allgemeines zum iSAQB-Advanced-Level**

### **0.1 Was vermittelt ein Advanced-Level-Modul?**

- Der iSAQB-Advanced-Level bietet eine modulare Ausbildung in drei Kompetenzbereichen mit flexibel gestaltbaren Ausbildungswegen. Er berücksichtigt individuelle Neigungen und Schwerpunkte.
- Die Zertifizierung erfolgt als Hausarbeit. Die Bewertung und mündliche Prüfung wird durch vom iSAQB benannte Experten vorgenommen.

### **0.2 Was können Absolventen des Advanced-Level (CPSA-A)?**

CPSA-A-Absolventen können:

- Eigenständig und methodisch fundiert mittlere bis große IT-Systeme entwerfen.
- In IT-Systeme mittlerer bis hoher Kritikalität technische und inhaltliche Verantwortung übernehmen.
- Konzeption, Entwurf und Dokumentation von Maßnahmen zur Erreichung nichtfunktionaler Anforderungen. Begleitung des Entwicklungsteams bei der Umsetzung dieser Maßnahmen durchführen.
- Architekturelevante Kommunikation in mittleren bis großen Entwicklungsteams steuern und durchführen.

### **0.3 Voraussetzungen zur CPSA-A-Zertifizierung**

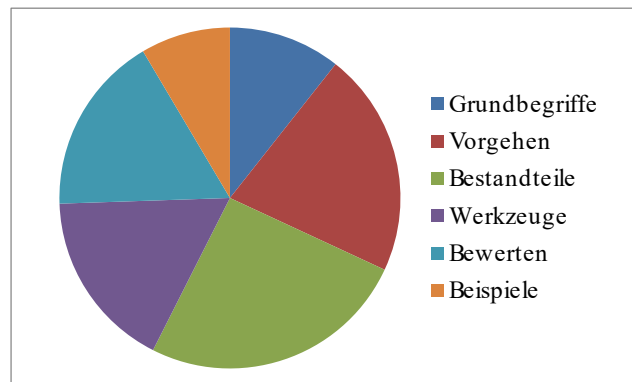
- Eine erfolgreiche Ausbildung und Zertifizierung zum CPSA-F (Certified Professional for Software Architecture Foundation Level®).
- Mindestens drei Jahre Vollzeit-Berufserfahrung in der IT-Branche, dabei Mitarbeit an Entwurf und Entwicklung von mindestens zwei unterschiedlichen IT-Systemen
  - Ausnahmen auf Antrag zulässig (etwa: Mitarbeit in OpenSource-Projekten).
- Aus- und Weiterbildung im Rahmen von iSAQB-Advanced-Level-Schulungen im Umfang von mindestens 70 Credit-Points aus mindestens zwei unterschiedlichen Kompetenzbereichen (detailliert geregelt in Abschnitt ).
  - Bestehende Zertifizierungen können ggfs. auf Antrag auf diese Credit Points angerechnet werden. Die Liste der aktuellen Zertifikate, für die Credit Points angerechnet werden, ist auf der iSAQB-Homepage zu finden.
  - Sonstige Aus- und Weiterbildungen können auf Antrag beim iSAQB ebenfalls anerkannt werden, sofern ein Bezug zu Software-Architektur vorliegt. Die Entscheidung hierüber trifft im Einzelfall die Arbeitsgruppe „Advanced Level“ des iSAQB.
- Erfolgreiche Bearbeitung der CPSA-A-Zertifizierungsprüfung.



# 1 Grundlegendes zum Modul Architekturdokumentation

## 1.1 Gliederung des Lehrplans für Architekturdokumentation und empfohlene zeitliche Aufteilung

- Grundbegriffe von Softwarearchitektur-Dokumentation (mind. 1.25h)
  - Dokumentation und Vorgehen (mind. 2.5h)
  - Bestandteile von Architekturdokumentation (mind. 3h)
  - Werkzeuge (mind. 2h)
  - Dokumentation bewerten (2h)
  - Beispiele für die Dokumentation von Software-Architekturen (mind. 2h)
- (Zeiten jeweils inklusive Übungen)



## 1.2 Dauer, Didaktik und weitere Details

Die genannten Zeiten sind Empfehlungen. Die Dauer entsprechender Schulungen sollte mindestens 2 Tage betragen, kann aber länger sein. Anbieter können sich durch Dauer, Didaktik, Art- und Aufbau der Übungen sowie der detaillierten Kursgliederung voneinander unterscheiden. Insbesondere die Art (fachliche und technische Domänen) der Beispiele und Übungen lässt der Lehrplan komplett offen.

Das Modul kann unabhängig von einer CPSA-F-Zertifizierung besucht werden.

## 1.3 Voraussetzungen für das Modul „Architekturdokumentation“

Teilnehmer **sollten** folgende Kenntnisse und/oder Erfahrung mitbringen:

- Grundlagen der Beschreibung von Architekturen mit Hilfe verschiedener Sichten, übergreifender Konzepte, Entwurfsentscheidungen, Randbedingungen etc., wie es im CPSA-F Foundation Level vermittelt wird.
- Wünschenswert sind eigene Erfahrungen in der Erstellung und Pflege technischer Dokumentation von Software, insbesondere der Architektur von Software- oder Software-nahen Systemen.

**Hilfreich** für das Verständnis einiger Konzepte sind darüber hinaus:

- Kenntnis typischer Herausforderungen bei Erstellung und Pflege von technischer Dokumentation:
  - Auswahl geeigneter Dokumentationsstrukturen, Notationen, Ergebnistypen (Stakeholderorientierung)
  - Behandlung großer Dokumentationen (insbesondere vorhandener oder veralteter Dokumentation)

- Auswahl, Konfiguration und Einführung von Werkzeugketten (mit welchen Mitteln kann Dokumentation angemessen erstellt und gepflegt werden),
- Versionierung von Dokumenten (Modelle und Texte)
- Dokumentation in Teams, arbeitsteilige Erstellung und Pflege
- Inhaltsbezogene und formale Reviews von Dokumentation.

## 1.4 Gliederung des Lehrplans für Architekturdokumentation

Die einzelnen Abschnitte des Lehrplans sind gemäß folgender Gliederung beschrieben:

- **Begriffe/Konzepte:** Wesentliche Kernbegriffe dieses Themas.
- **Unterrichts-/Übungszeit:** Legt die Unterrichts- und Übungszeit fest, die für dieses Thema bzw. dessen Übung in einer akkreditierten Schulung mindestens aufgewendet werden muss.
- **Lernziele:** Beschreibt die zu vermittelnden Inhalte inklusive ihrer Kernbegriffe und -konzepte.

Dieser Abschnitt skizziert damit auch die zu erwerbenden Kenntnisse in entsprechenden Schulungen. Die Lernziele werden differenziert in folgende Kategorien bzw. Unterkapitel:

- Was sollen die Teilnehmer **können**? Diese Inhalte sollen die Teilnehmer nach der Schulung selbständig anwenden können. Innerhalb der Schulung werden diese Inhalte durch Übungen abgedeckt und sind Bestandteil der Abschlussprüfung des iSAQB-Advanced-Levels.
- Was sollen die Teilnehmer **verstehen**? Diese Inhalte können geprüft werden.
- Was sollen die Teilnehmer **kennen**? Diese Inhalte (Begriffe, Konzepte, Methoden, Praktiken oder Ähnliches) können das Verständnis unterstützen oder das Thema motivieren. Diese Inhalte sind nicht Bestandteil der Prüfung, werden in Schulungen thematisiert, aber nicht notwendigerweise ausführlich unterrichtet.

## 1.5 Ergänzende Informationen, Begriffe, Übersetzungen

Soweit für das Verständnis des Lehrplans erforderlich, haben wir Fachbegriffe ins iSAQB-Glossar aufgenommen, definiert und bei Bedarf durch die Übersetzungen der Originalliteratur ergänzt.

## 1.6 Credit Points für diese Schulung

Vom iSAQB e. V. lizenzierte Schulungen gemäß diesem Lehrplan geben 20 Credit Points im Bereich der methodischen Kompetenz.

## 2 Einführung in das iSAQB-Zertifizierungsprogramm

Dauer: 15 Min (optional)	Übungszeit: keine
--------------------------	-------------------

Dieser Abschnitt ist nicht prüfungsrelevant. Falls Teilnehmer bereits CPSA-F zertifiziert sind, kann dieser Abschnitt entfallen.

### 2.1 Begriffe und Konzepte

iSAQB, Advanced-Level-Zertifizierung und Voraussetzung dazu.

### 2.2 Lernziele

Die Teilnehmer lernen den Kontext des iSAQB-Zertifizierungsprogrammes und der zugehörigen Prüfungen beziehungsweise Prüfungsmodalitäten kennen.

#### 2.2.1 Was sollen die Teilnehmer kennen?

- iSAQB als Verein
- Advanced Level in Abgrenzung zu anderen Level
- Randbedingungen und Vorgehen beim iSAQB-Zertifizierungsprogramm.



## 3 Grundbegriffe von Softwarearchitektur-Dokumentation

Dauer: 55 Min	Übungszeit: ca. 20 Min
---------------	------------------------

### 3.1 Begriffe und Konzepte

Software-Architektur, Dokumentation, Stakeholder, Notation, Vorgehensmodell.

### 3.2 Lernziele

#### 3.2.1 Was sollen die Teilnehmer können?

- Den Nutzen und die verschiedenen Ziele von Architekturdokumentation beschreiben und überzeugend vertreten können.
- Massgebliche Nutzer und die Bedeutung von Architekturdokumentation für unterschiedliche Stakeholder erklären können.
- Arbeitsergebnisse aus der Architekturdokumentation gegen Ergebnisse anderer Disziplinen (Anforderungsmanagement, Projektmanagement, ...) abgrenzen können.
- Typische Notationen für Architekturdokumentation nennen können.
- Konkrete Vorgehensmodelle und ihren „Beitrag“ bezüglich Architekturdokumentation nennen können.

#### 3.2.2 Was sollen die Teilnehmer verstehen?

- Architekturdokumentation ist kein Selbstzweck.
- Architekturdokumentation unterstützt beim Lösungsentwurf, und bei der Kommunikation der Lösung dem Team und anderen (z. B. Auftraggebern) gegenüber.
- Verschiedene Vorgehensmodelle machen ausführliche bis gar keine Vorgaben bezüglich konkreter Ergebnisse.
- Je nach Vorgehen und Rahmenbedingungen des Vorhabens sind sehr unterschiedliche Dokumentationsarbeiten zu leisten.
- UML ist im Zusammenhang mit Architekturdokumentation etabliert, aber nicht zwingend erforderlich.

#### 3.2.3 Was sollen die Teilnehmer kennen?

Entfällt.

## 4 Dokumentation und Vorgehen

Dauer: 90 Min	Übungszeit: ca. 60 Min
---------------	------------------------

### 4.1 Begriffe und Konzepte

Vorgehensmodell, Rolle, Artefakt, iterativ/inkrementell, Lebenszyklus, agil.

### 4.2 Lernziele

Teilnehmer können für gegebene Situation angemessene Vorgehensweise für die Dokumentation bestimmen/entwerfen.

#### 4.2.1 Was sollen die Teilnehmer können?

- Architekturdokumentation für folgende Situationen erstellen:
  - Erstellung neuer Systeme
  - Weiterentwicklung bestehender Systeme mit vorhandener Dokumentation
  - Nachdokumentation bestehender Systeme ohne vorhandene Dokumentation
  - Dokumentation in Ausnahmesituationen (wenig Budget, wenig Zeit, wenig verfügbare Informationsquellen etc.)
- Zielgruppen und Ziele der Dokumentation ermitteln
- Zu erstellende Bestandteile bedarfsgerecht auswählen
- Die Vorteile fester Gliederungen und Strukturen für (technische) Dokumentation argumentieren können.

#### 4.2.2 Was sollen die Teilnehmer verstehen?

- Dokumentation entsteht im Idealfall begleitend zu Architektur und Entwicklung, nicht hinterher.
- Agilität und Dokumentation stehen nicht im Widerspruch zueinander.

#### 4.2.3 Was sollen die Teilnehmer kennen?

- Die Rolle von Architekturdokumentation in verschiedenen konkreten Vorgehensmodellen
- Verschiedene Vorschläge und Vorlagen für Standardgliederungen (z. B. arc42, SEI).

## 5 Bestandteile von Architekturdokumentationen

Dauer: 90 Min	Übungszeit: ca. 90 Min
---------------	------------------------

### 5.1 Begriffe und Konzepte

Sichten, Entscheidungen, übergreifende Konzepte und Themen, Schnittstellen, Strukturen, Randbedingungen, Risiken, Qualitätsziele.

### 5.2 Lernziele

#### 5.2.1 Was sollen die Teilnehmer können?

- Die Architektur mittlerer bis großer IT-Systeme selbständig dokumentieren, sowohl entwicklungsbegleitend, wie bei Bedarf auch a posteriori („Nachdokumentation“).
- Vorhandene Architekturdokumentation weiterentwickeln und pflegen, dabei die Qualität der Dokumentation systematisch verbessern.
- Für mittlere bis große Systeme spezifisch geeignete Arten und Strukturen der Architekturdokumentation erarbeiten oder auswählen.
- Faktoren, die Einfluss auf Software-Architektur haben, identifizieren und geeignet dokumentieren, z. B.
  - Randbedingungen
  - Qualitätsziele
  - Technische Risiken
- Architekturentscheidungen nachvollziehbar festhalten
- Unterschiedliche Sichten auf Software-Architektur mit geeigneten Beschreibungsmitteln anfertigen
  - Verschiedene Aspekte von Bausteinsichten erklären und dokumentieren (Darstellung als hierarchische Verfeinerung, Dokumentation wiederkehrender Strukturen oder Muster, Beschreibung interner Schnittstellen, Zusammenhang von Bausteinen zu Quellcode)
  - Laufzeitszenarien, insbesondere zur Erfüllung der wesentlichen Systemfunktionen, Darstellung unvollständiger Szenarien, Darstellung über verschiedene Verfeinerungsebenen
  - Notationen (etwa: UML, SysML o. ä.) zur Beschreibung von Sichten
- Schnittstellen beschreiben
- Technische und oder übergreifende Konzepte zielgruppengerecht anfertigen
- Konsistenz zwischen unterschiedlichen Bestandteilen der Architekturdokumentation prüfen bzw. herstellen
- Bestandteile in Dokumenten strukturieren.

#### 5.2.2 Was sollen die Teilnehmer verstehen?

- Für eine nachvollziehbare Software-Architektur reicht es nicht, die Lösung zu dokumentieren. Auch architekturelevante Anforderungen und Begründungen sind festzuhalten.

#### 5.2.3 Was sollen die Teilnehmer kennen?

- IEEE 1471 („Recommended Practice for Architecture Description of Software-Intensive Systems“)
- Alternative Vorschläge für konkrete Sichten, zum Beispiel
  - Rational Unified Process (4+1)

- Meta-/Sichtenmodell vom Software Engineering Institute
  - arc42
- Grafische Notationen für Sichten
- FMC (Fundamental Modeling Concepts)
- RM-ODP (Reference Model of Open Distributed Processing)

## 6 Werkzeuge

Dauer: 90 Min	Übungszeit: ca. 30 Min
---------------	------------------------

### 6.1 Begriffe und Konzepte

Analoge und digitale Werkzeuge, Modellierungstools, Toolkette.

### 6.2 Lernziele

#### 6.2.1 Was sollen die Teilnehmer können?

- Die richtigen Werkzeuge für die verschiedenen Aktivitäten auswählen:
  - Erstellung und Pflege von Bestandteilen von Architekturdokumentation
  - Verwaltung der Bestandteile
  - Kommunikation von Inhalten mit Unterstützung der Bestandteile
- Analoge und digitale Dokumentationswerkzeuge situations- und bedarfsgerecht einsetzen und die Ergebnisse sichern
- Eine vollständige Toolkette zur Architekturdokumentation anhand konkreter Anforderungen, Randbedingungen und weiterer Einflussfaktoren nachvollziehbar auswählen.

#### 6.2.2 Was sollen die Teilnehmer verstehen?

- Werkzeuge können bei sehr verschiedenen Aktivitäten rund um Architekturdokumentation unterstützen.
- Dokumentation sollte, analog zu Quellcode, versioniert und releasefähig verwaltet werden. Zu jedem definierten Stand der Software sollte der zugehörige Stand der (Architektur-) Dokumentation wieder hergestellt oder generiert werden können.

#### 6.2.3 Was sollen die Teilnehmer kennen?

- Jeweilige Stärken, Schwächen und typische Herausforderungen beim Einsatz und der Integration von verbreiteten Werkzeugenkategorien, konkret:
  - Wikis
  - Modellierungswerkzeuge
  - Zeichenprogramme
  - Textverarbeitungen
  - Versionsverwaltungen
  - Sonstige (etwa: Blogs, Issue-Tracker etc.)
- Konkrete vollständige Werkzeugketten, welche sämtliche Aktivitäten rund um Architekturdokumentation adressieren.

## 7 Dokumentation bewerten

Dauer: 75 Min	Übungszeit: ca. 45 Min
---------------	------------------------

### 7.1 Begriffe und Konzepte

Review, Checklisten, Fragenkataloge.

### 7.2 Lernziele

#### 7.2.1 Was sollen die Teilnehmer können?

- Zwischen inhaltlicher und formaler Begutachtung einer Architekturdokumentation differenzieren
- Review einer Dokumentation von der Bewertung der Architektur abgrenzen
- Unterschiedliche Ziele beim Review der Dokumentation erklären
- Angemessene Review-Schritte auswählen, planen und durchführen
- Ziele für ein Review festlegen und verbindlich an die Gutachter kommunizieren
- Checklisten und Fragenkataloge für Reviews erstellen
- Im Rahmen einer Review-Sitzung
  - In der Rolle des Autors mit Einwänden angemessen umgehen
  - In der Rolle eines Gutachters konstruktive Rückmeldungen an Autoren geben
  - In der Rolle des Moderators ein fachliches Review leiten

#### 7.2.2 Was sollen die Teilnehmer verstehen?

- Die Überprüfung der Dokumentation auf Gebrauchstauglichkeit ist ein essentieller Erfolgsfaktor.

#### 7.2.3 Was sollen die Teilnehmer kennen?

- IEEE 1028-2008 "IEEE Standard for Software Reviews and Audits"
- Verschiedene Beispiele von Architekturdokumentation.

## 8 Beispiele für Dokumentation von Software-Architekturen

Dauer: 60 Min	Übungszeit: keine
---------------	-------------------

Dieser Abschnitt ist nicht prüfungsrelevant.

### 8.1 Begriffe und Konzepte

Innerhalb jeder akkreditierten Schulung muss mindestens ein Beispiel einer dokumentierten Software-Architektur vorgestellt werden.

Art und Ausprägung der vorgestellten Beispiele können von der Schulung bzw. den Interessen der Teilnehmer abhängen und werden seitens iSAQB nicht vorgegeben.

### 8.2 Lernziele

Die Dokumentation (mindestens) einer realen Software-Architektur sehen und deren Vor- und Nachteile diskutieren.

#### 8.2.1 Was sollen die Teilnehmer können?

Erklären, wie methodische Dokumentation im Beispiel umgesetzt wurde, Stärken und Schwächen des vorgestellten Beispiels analysieren können.

#### 8.2.2 Was sollen die Teilnehmer verstehen?

Nicht zutreffend.

#### 8.2.3 Was sollen die Teilnehmer kennen?

Weitere Beispiele (möglichst) realitätsnaher Architekturdokumentationen.

### 8.3 Referenzen

Keine. Schulungsanbieter sind für die Auswahl und Beschreibung von Beispielen verantwortlich.

## **9 Quellen und Referenzen zu Softwarearchitektur-Dokumentation**

Dieser Abschnitt enthält Quellenangaben, die ganz oder teilweise im Curriculum referenziert werden.

### **B**

[Bachmann 2000]

Bachmann, F., L. Bass, et al.: Software Architecture Documentation in Practice. Software Engineering Institute, CMU/SEI-2000-SR-004.

[Bass+2003]

Bass, L., Clements, P. und Kazman, R. (2003): Software Architecture in Practice. Addison-Wesley, Reading, Mass

### **C**

[Clements+2003]

Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers et al: Documenting Software Architectures – Views and Beyond. Addison Wesley, 2003.

### **H**

[Hargis+2004]

Hargis, Gretchen et. al: Quality Technical Information: A Handbook for Writers and Editors. Prentice Hall, IBM Press, 2004..

### **K**

[Kruchten 1995]

Kruchten, P.: Architectural Blueprints – The 4-1 View Model of Architecture. IEEE Software November 1995; 12(6), p. 42-50.

### **S**

[Starke 2011]

Starke, G. (2011): Effektive Software-Architekturen - Ein praktischer Leitfaden. 5. Auflage 2011, Carl Hanser Verlag, München.

### **Z**

[Zörner 2012]

Zörner, Stefan: Softwarearchitekturen dokumentieren und kommunizieren, Hanser Verlag 2012