

Curriculum für
Certified Professional for
Software Architecture (CPSA)[®]
Advanced Level

**Modul
ADOC**

Architekturdokumentation

2021.1-DE-20210122



Inhaltsverzeichnis

Verzeichnis der Lernziele	2
Einführung: Allgemeines zum iSAQB Advanced Level	3
Was vermittelt ein Advanced Level Modul?	3
Was können Absolventen des Advanced Level (CPSA-A)?	3
Voraussetzungen zur CPSA-A-Zertifizierung	3
Grundlegendes	4
Was vermittelt das Modul „ADOC“?	4
Struktur des Lehrplans und empfohlene zeitliche Aufteilung	4
Dauer, Didaktik und weitere Details	4
Voraussetzungen	4
Gliederung des Lehrplans	5
Ergänzende Informationen, Begriffe, Übersetzungen	5
1. Grundbegriffe von Architekturdokumentation	6
1.1. Begriffe und Konzepte	6
1.2. Lernziele	6
2. Vorgehen bei Architekturdokumentation	8
2.1. Begriffe und Konzepte	8
2.2. Lernziele	8
3. Bestandteile von Architekturdokumentationen	10
3.1. Begriffe und Konzepte	10
3.2. Lernziele	10
4. Werkzeuge	11
4.1. Begriffe und Konzepte	11
4.2. Lernziele	11
5. Dokumentation bewerten	12
5.1. Begriffe und Konzepte	12
5.2. Lernziele	12
6. Beispiele für Dokumentation von Softwarearchitekturen	13
6.1. Lernziele	13
Referenzen	14

© (Copyright), International Software Architecture Qualification Board e. V. (iSAQB® e. V.) 2021

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

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 info@isaqb.org nachfragen. Es müsste dann ein eigener Lizenzvertrag geschlossen werden.
2. Sind Sie Trainer oder Trainingsprovider, ist die Nutzung der Dokumente und/oder Lehrpläne nach Erwerb einer Nutzungslizenz möglich. Hierzu bitte unter info@isaqb.org nachfragen. Lizenzverträge, die alles umfassend 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 info@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.

Wichtiger Hinweis

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.

Die Abkürzung "e. V." ist Teil des offiziellen Namens des iSAQB und steht für "eingetragener Verein", der seinen Status als juristische Person nach deutschem Recht beschreibt. Der Einfachheit halber wird iSAQB e. V. im Folgenden ohne die Verwendung dieser Abkürzung als iSAQB bezeichnet.

Verzeichnis der Lernziele

- LZ 1-1: Nutzen und Ziele von Architekturdokumentation verstehen
- LZ 1-2: Bedeutung von Architekturdokumentation für unterschiedliche Stakeholder verstehen
- LZ 1-3: Architekturdokumentation gegen andere Arten von Dokumentation abgrenzen
- LZ 1-4: Typische Notations- und Darstellungsmittel für Architekturdokumentation kennen
- LZ 1-5: Varianten von Entwicklungsvorgehen und deren Einfluss auf Architekturdokumentation kennen
- LZ 2-1: Architekturdokumentation erstellen
- LZ 2-2: Zielgruppen und Ziele der Architekturdokumentation ermitteln
- LZ 2-3: Situativ passende Struktur einer Architekturdokumentation bestimmen
- LZ 2-4: Situativ passendes Vorgehen zur Architekturdokumentation bestimmen
- LZ 2-5: Beispiele standardisierter Dokumentationsstrukturen (Templates) kennen
- LZ 3-1: Einflussfaktoren und Randbedingungen dokumentieren
- LZ 3-2: Architekturentscheidungen dokumentieren
- LZ 3-3: Architektursichten dokumentieren
- LZ 3-4: Übergreifende Konzepte dokumentieren
- LZ 3-5: Konsistenz zwischen Teilen der Dokumentation sicherstellen
- LZ 4-1: Passende Werkzeuge für Dokumentationsarbeit auswählen
- LZ 4-2: Anforderungen an Dokumentationswerkzeuge und -artefakte kennen
- LZ 5-1: Notwendigkeit der Überprüfung auf Gebrauchstauglichkeit erkennen
- LZ 5-2: Verschiedene Arten der Begutachtung differenzieren
- LZ 5-3: Review von Architektur und Dokumentation differenzieren
- LZ 5-4: Ziele von Dokumentations-Reviews ermitteln und erklären
- LZ 5-5: Checklisten und Fragenkataloge für Reviews erstellen
- LZ 5-6: Verschiedene Rollen bei Dokumentations-Reviews ausüben können
- LZ 5-7: Auf Basis von Review-Ergebnissen die Qualität von Dokumentation verbessern
- LZ 6-1: Beispiele praktischer Architekturdokumentation kennen

Einführung: Allgemeines zum iSAQB Advanced Level

Was vermittelt ein Advanced Level Modul?

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

- 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.

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

CPSA-A-Absolventen können:

- eigenständig und methodisch fundiert mittlere bis große IT-Systeme entwerfen
- in IT-Systemen mittlerer bis hoher Kritikalität technische und inhaltliche Verantwortung übernehmen
- Maßnahmen zur Erreichung von Qualitätsanforderungen konzeptionieren, entwerfen und dokumentieren sowie Entwicklungsteams bei der Umsetzung dieser Maßnahmen begleiten
- architekturrelevante Kommunikation in mittleren bis großen Entwicklungsteams steuern und durchführen

Voraussetzungen zur CPSA-A-Zertifizierung

- erfolgreiche Ausbildung und Zertifizierung zum Certified Professional for Software Architecture, Foundation Level® (CPSA-F)
- mindestens drei Jahre Vollzeit-Berufserfahrung in der IT-Branche; dabei Mitarbeit an Entwurf und Entwicklung von mindestens zwei unterschiedlichen IT-Systemen
 - Ausnahmen sind auf Antrag zulässig (etwa: Mitarbeit in Open-Source-Projekten)
- Aus- und Weiterbildung im Rahmen von iSAQB-Advanced-Level-Schulungen im Umfang von mindestens 70 Credit Points aus mindestens drei unterschiedlichen Kompetenzbereichen
 - bestehende Zertifizierungen (etwa Sun/Oracle Java-Architect, Microsoft CSA) können auf Antrag angerechnet werden
- erfolgreiche Bearbeitung der CPSA-A-Zertifizierungsprüfung



Grundlegendes

Was vermittelt das Modul „ADOC“?

ADOC vermittelt die Fähigkeiten:

- die Dokumentation von Softwarearchitekturen selbständig zu konzipieren und zu erstellen
- bestehende Dokumentation bezüglich ihrer Eignung und Angemessenheit zu bewerten
- bestehende Dokumentation zu verbessern.

Teilnehmende lernen, die wesentlichen Aspekte von Softwarearchitekturen (u. a. Entscheidungen, Strukturen, Konzepte, Qualitätsanforderungen und Sichten) systematisch zu beschreiben. Dazu stellt ADOC grundlegende Beschreibungs- und Notationsmittel sowie mögliche Werkzeuge zur praktischen Umsetzung vor.

Struktur des Lehrplans und empfohlene zeitliche Aufteilung

Inhalt	Empfohlene Mindestdauer (min)
1. Grundbegriffe	90
2. Vorgehen bei Architekturdokumentation	150
3. Bestandteile von Architekturdokumentationen	180
4. Werkzeuge	120
5. Dokumentation bewerten	120
6. Beispiele für Dokumentation	60
Summe	720 (12h)

Dauer, Didaktik und weitere Details

Die unten genannten Zeiten sind Empfehlungen. Die Dauer einer Schulung zum Modul ADOC 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 der Beispiele und Übungen lässt der Lehrplan komplett offen.

Lizenzierte Schulungen zu ADOC tragen zur Zulassung zur abschließenden Advanced-Level-Zertifizierungsprüfung folgende Credit Points) bei:

Methodische Kompetenz:	20 Punkte
Technische Kompetenz:	0 Punkte
Kommunikative Kompetenz:	0 Punkte

Voraussetzungen

Teilnehmende sollten folgende Kenntnisse und/oder Erfahrung mitbringen:

- Grundlagen der Beschreibung von Architekturen mit Hilfe verschiedener Sichten, übergreifender Konzepte, Entwurfsentscheidungen, Randbedingungen etc., wie es im CPISA- 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 o Inhaltsbezogene und formale Reviews von Dokumentation.

Gliederung des Lehrplans

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.

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. Grundbegriffe von Architekturdokumentation

Dauer: 70 Min.	Übungszeit: ca. 20 Min.
----------------	-------------------------

1.1. Begriffe und Konzepte

Softwarearchitektur, Dokumentation, Stakeholder, Notation, Vorgehensmodell, Lösungsentwurf, UML.

1.2. Lernziele

LZ 1-1: Nutzen und Ziele von Architekturdokumentation verstehen

Softwarearchitekt:innen wissen, dass Architekturdokumentation kein Selbstzweck ist.

Sie können den Nutzen und verschiedene Ziele von Architekturdokumentation erklären, beispielsweise:

- Architekturdokumentation unterstützt beim Lösungsentwurf, und bei der Kommunikation der Lösung dem Team und anderen (z. B. Auftraggebern) gegenüber.
- Unterstützung der Entwicklung durch Vermittlung architektur- und entwicklungsrelevanter Entscheidungen, Fakten und Gründe.
- Schriftliche Kommunikation relevanter struktureller und konzeptioneller Sachverhalte
- Konservierung von Wissen über Entwicklung und Architektur

Sie können diese Ziele und den Nutzen von Dokumentation gegenüber Stakeholdern überzeugend vertreten.

LZ 1-2: Bedeutung von Architekturdokumentation für unterschiedliche Stakeholder verstehen

Softwarearchitekt:innen verstehen, dass unterschiedliche Stakeholder i. d. R. verschiedene Anforderungen an Architekturdokumentation haben, beispielsweise hinsichtlich:

- Arten von Informationen, etwa zu Schnittstellen, Bausteinen, querschnittlichen Themen, Entscheidungen, technischen Risiken, grundlegenden Lösungsansätzen o. ä.
- Darstellung solcher Informationen, etwa als (grafische oder textuelle) Modelle, als Grafiken oder Text
- Detaillierungsgrad, etwa: Überblick, abstrakte oder detaillierte Darstellung
- technische Repräsentation etwa mit oder ohne Schreib-/Änderungsmöglichkeiten, als PDF, HTML, Wiki, Textverarbeitung o. ä.
- Grad der Aktualität.

LZ 1-3: Architekturdokumentation gegen andere Arten von Dokumentation abgrenzen

Softwarearchitekt:innen verstehen, dass Architekturdokumentation Berührungspunkte und teilweise Überdeckung zu anderen Arten von Dokumentation besitzt, beispielsweise:

- Anforderungs-/Requirements-Dokumentation
- Implementierungsdokumentation
- Managementdokumentation

- Testdokumentation
- Betriebsdokumentation

LZ 1-4: Typische Notations- und Darstellungsmittel für Architekturdokumentation kennen

Softwarearchitekt:innen kennen verschiedene Notations- und Darstellungsmittel für Architekturinformationen, beispielsweise:

- Modellierungsmethoden, wie UML, BPMN, ER-Diagramme, Ereignis-Prozessketten
- Textuelle Erläuterungen von Entscheidungen

Sie wissen, dass formale Notationen wie UML für Architekturdokumentation hilfreich sein können, aber nicht zwingend erforderlich sind.

LZ 1-5: Varianten von Entwicklungsvorgehen und deren Einfluss auf Architekturdokumentation kennen

Softwarearchitekt:innen wissen, dass

- in eher formalen Entwicklungsvorhaben (etwa *safety-critical*) hohe Ansprüche an Architekturdokumentation gelten
- in eher leichtgewichtigen Entwicklungsprozessen (agilen Prozesse) Teams eine dem System und dessen Umfeld angemessene Art von Dokumentation wählen sollten
- je nach Vorgehen und Rahmenbedingungen des Vorhabens sehr unterschiedliche Dokumentationsarbeiten zu leisten sind

2. Vorgehen bei Architekturdokumentation

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

2.1. Begriffe und Konzepte

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

2.2. Lernziele

LZ 2-1: Architekturdokumentation erstellen

Softwarearchitekt:innen können angemessene Architekturdokumentation für mittlere bis große IT-Systeme in beispielsweise folgenden 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.)

Sie können Zielgruppen und Ziele der Dokumentation ermitteln (siehe [LZ 2-2](#)).

Details zu den notwendigen methodischen Mitteln finden sich in den Lernzielen [LZ 3-1](#) bis [LZ 3-5](#).

LZ 2-2: Zielgruppen und Ziele der Architekturdokumentation ermitteln

Softwarearchitekt:innen können die Zielgruppen („Stakeholder“) für Architekturdokumentation ermitteln (beispielsweise Management, Entwicklungsteam, fachliche Stakeholder, betriebliche Stakeholder, Test/QS).

Sie können zusammen mit diesen Stakeholdern konkrete Ziele der Dokumentation ermitteln.

LZ 2-3: Situativ passende Struktur einer Architekturdokumentation bestimmen

Softwarearchitekt:innen können für gegebene Situationen und Systeme eine passende Struktur für die Architekturdokumentation auswählen oder erarbeiten.

Als Basis dafür dienen Zielgruppen und Ziele (siehe [LZ 2-2](#)) sowie Templates (siehe [LZ 2-5](#)).

LZ 2-4: Situativ passendes Vorgehen zur Architekturdokumentation bestimmen

Softwarearchitekt:innen finden für gegebene Situationen ein passendes Vorgehen zur Dokumentation. Sie können Stakeholdern und dem Entwicklungsteam die Vor- und Nachteile verschiedener Ansätze erläutern, etwa:

- Dokumentation entsteht **begleitend zur** Entwicklungsarbeit
- Dokumentation entsteht **vor** der Entwicklung/Implementierung
- Dokumentation entsteht **nach** der Entwicklung („Dokumentations-Sprint“)

- Dokumentation wird durch Personen **aus dem Entwicklungsteam** erstellt und gepflegt
- Dokumentation wird durch Personen **außerhalb des Entwicklungsteams** erstellt und gepflegt

Sie wissen, dass agile Entwicklung und Architekturdokumentation nicht im Widerspruch zueinander stehen, und können dies Stakeholdern gegenüber begründen.

LZ 2-5: Beispiele standardisierter Dokumentationsstrukturen (Templates) kennen

Softwarearchitekt:innen kennen Beispiele von Architekturtemplates, etwa:

- arc42
- C4 (Simon Brown)
- IEEE 1471 („Recommended Practice for Architecture Description of Software-Intensive Systems“)

3. Bestandteile von Architekturdokumentationen

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

3.1. Begriffe und Konzepte

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

3.2. Lernziele

LZ 3-1: Einflussfaktoren und Randbedingungen dokumentieren

Softwarearchitekt:innen können Faktoren, die Einfluss auf Softwarearchitektur haben, identifizieren und geeignet dokumentieren. Dazu zählen:

- Randbedingungen
- Qualitätsanforderungen und -ziele
- Technische Risiken

LZ 3-2: Architekturentscheidungen dokumentieren

Softwarearchitekt:innen können Architekturentscheidungen nachvollziehbar festhalten. Sie kennen beispielsweise *Architecture Decision Records* als eine feste Struktur für solche Entscheidungen.

LZ 3-3: Architektursichten dokumentieren

Softwarearchitekt:innen können verschiedene Architektursichten angemessen einsetzen und dokumentieren. Sie wissen, für welche Anwendungsfälle statische sowie dynamische Sichten am besten geeignet sind. Außerdem verstehen sie, wann welcher Detailgrad sinnvoll ist.

LZ 3-4: Übergreifende Konzepte dokumentieren

Softwarearchitekt:innen können Konzepte, die über Modulgrenzen hinaus oder querschnittlich durch die Software gehen, erkennen und entsprechend dokumentieren.

LZ 3-5: Konsistenz zwischen Teilen der Dokumentation sicherstellen

Softwarearchitekt:innen sind in der Lage, Konsistenz zwischen einzelnen Teilen der Dokumentation herzustellen und zu bewahren.

4. Werkzeuge

Dauer: 90 min	Übungszeit: 30 min
---------------	--------------------

4.1. Begriffe und Konzepte

Analoge und digitale Werkzeuge, Modellierungstools, Toolkette.

4.2. Lernziele

LZ 4-1: Passende Werkzeuge für Dokumentationsarbeit auswählen

Softwarearchitekt:innen können passenden Werkzeuge für die verschiedenen Aktivitäten auswählen:

- Erstellung und Pflege von Bestandteilen von Architekturdokumentation
- Verwaltung der Bestandteile
- Kommunikation von Inhalten mit Unterstützung der Bestandteile

Sie können:

- analoge und digitale Dokumentationswerkzeuge situations- und bedarfsgerecht einsetzen
- eine vollständige Werkzeugkette zur Architekturdokumentation anhand konkreter Anforderungen, Randbedingungen und weiterer Einflussfaktoren nachvollziehbar auswählen.

Sie kennen jeweilige Stärken, Schwächen und typische Herausforderungen beim Einsatz und der Integration verbreiteter Werkzeugenkategorien, beispielsweise:

- Wikis
- Modellierungswerkzeuge
- Zeichenprogramme
- Textverarbeitungen
- Versionsverwaltungen
- Sonstige (etwa: Blogs, Issue-Tracker etc.)

LZ 4-2: Anforderungen an Dokumentationswerkzeuge und -artefakte kennen

Softwarearchitekt:innen wissen, wie Werkzeuge bei verschiedenen Aufgaben rund um Architekturdokumentation unterstützen können.

Sie wissen, dass Dokumentation:

- ähnlich wie Quellcode versioniert und Release-fähig verwaltet werden sollte
- zu jedem definierten Stand der Software herstellbar oder generierbar sein sollte

5. Dokumentation bewerten

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

5.1. Begriffe und Konzepte

Review, Checklisten, Fragenkataloge.

Die Lernziele in diesem Abschnitt beschäftigen sich mit der Überprüfung von Architekturdokumentation auf ihre Gebrauchstauglichkeit. Für den praktischen Einsatz von Dokumentation ist dies ein wesentlicher Erfolgsfaktor.

5.2. Lernziele

LZ 5-1: Notwendigkeit der Überprüfung auf Gebrauchstauglichkeit erkennen

Softwarearchitekt:innen erkennen die Überprüfung auf Gebrauchstauglichkeit als einen wesentlichen Erfolgsfaktor der Dokumentation.

LZ 5-2: Verschiedene Arten der Begutachtung differenzieren

Softwarearchitekt:innen können zwischen inhaltlicher und formaler Begutachtung einer Architekturdokumentation differenzieren.

LZ 5-3: Review von Architektur und Dokumentation differenzieren

Softwarearchitekt:innen können Review/Begutachtung einer Dokumentation von der Bewertung der Architektur oder des Systems abgrenzen.

LZ 5-4: Ziele von Dokumentations-Reviews ermitteln und erklären

Softwarearchitekt:innen können mit den betroffenen Personen ("Stakeholdern") mögliche Ziele von Dokumentations-Reviews ermitteln und diese Ziele anderen Personen erklären.

LZ 5-5: Checklisten und Fragenkataloge für Reviews erstellen

Softwarearchitekt:innen können Checklisten und Fragenkataloge für Reviews von Dokumentation erstellen.

LZ 5-6: Verschiedene Rollen bei Dokumentations-Reviews ausüben können

Softwarearchitekt:innen können im Rahmen von Dokumentations-Reviews in der Rolle von

- Autor:innen mit Einwänden angemessen umgehen,
- Gutachter:innen konstruktive Rückmeldungen an Autor:innen geben,
- Moderator:innen ein fachliches Review leiten.

LZ 5-7: Auf Basis von Review-Ergebnissen die Qualität von Dokumentation verbessern

Softwarearchitekt:innen können auf Basis ermittelter Review-Ergebnisse vorhandene Architekturdokumentation weiterentwickeln und pflegen und dabei die Qualität dieser Dokumentation systematisch verbessern.

6. Beispiele für Dokumentation von Softwarearchitekturen

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

Innerhalb jeder akkreditierten Schulung muss mindestens ein Beispiel einer dokumentierten Softwarearchitektur vorgestellt werden.

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

6.1. Lernziele

LZ 6-1: Beispiele praktischer Architekturdokumentation kennen

Softwarearchitekten:innen haben an mindestens einem Beispiel Architekturdokumentation kennen gelernt.

Sie können Vor- und Nachteile vorgestellter Beispiele erklären.

Referenzen

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

- [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
- [Clements+2003] Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers et al: Documenting Software Architectures – Views and Beyond. Addison Wesley, 2003.
- [Hargis+2004] Hargis, Gretchen et. al: Quality Technical Information: A Handbook for Writers and Editors. Prentice Hall, IBM Press, 2004.
- [Kruchten 1995] Kruchten, P.: Architectural Blueprints – The 4-1 View Model of Architecture. IEEE Software November 1995; 12(6), p. 42-50.
- [Nygard] Nygard, Michael: Documenting Architecture Decision. <https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions>. See also <https://adr.github.io/>
- [Starke 2019] Starke, Gernot: Effektive Softwarearchitekturen - Ein praktischer Leitfaden. 9. Auflage 2019, Carl Hanser Verlag, München.
- [Starke+2016] Gernot Starke, Peter Hruschka: Communicating Software Architectures: lean, effective and painless documentation. Leanpub <https://leanpub.com/arc42inpractice>
- [UML] Universal Modeling Language, homepage: <http://uml.org/>. Provides both the standard plus a collection of resources for further information.
- [Zörner 2015] Stefan Zörner: Softwarearchitekturen dokumentieren und kommunizieren: Entwürfe, Entscheidungen und Lösungen nachvollziehbar und wirkungsvoll festhalten. 2. Auflage, Carl-Hanser Verlag 2015