

Curriculum für
CPSA Certified Professional for
Software Architecture®

– Advanced Level –

**Modul:
SOA-T**

**Service-Oriented
Architecture –
Technisch**



Version 1.3 (Februar 2015)

**© (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 nachfragen. Es müsste dann ein eigener Lizenzvertrag geschlossen werden.
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 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 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

<u>0</u>	<u>EINLEITUNG: ALLGEMEINES ZUM ISAQB-ADVANCED-LEVEL.....</u>	<u>5</u>
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
<u>1</u>	<u>GRUNDLEGENDES ZUM MODUL SERVICE-ORIENTED ARCHITECTURE - TECHNISCH (SOA-T)</u>	<u>6</u>
1.1	GLIEDERUNG DES LEHRPLANS FÜR DAS MODUL SERVICE-ORIENTED ARCHITECTURE - TECHNISCH (SOA-T).....	6
1.2	DAUER, DIDAKTIK UND WEITERE DETAILS.....	6
1.3	VORAUSSETZUNGEN FÜR DAS MODUL SERVICE-ORIENTED ARCHITECTURE - TECHNISCH (SOA-T).....	6
1.4	GLIEDERUNG DES LEHRPLANS FÜR SERVICE-ORIENTIERTE SOFTWARE-ARCHITEKTUR	7
1.5	ERGÄNZENDE INFORMATIONEN, BEGRIFFE, ÜBERSETZUNGEN	7
1.6	CREDIT POINTS FÜR DIESE SCHULUNG.....	7
<u>2</u>	<u>EINFÜHRUNG IN DAS ISAQB-ZERTIFIZIERUNGSPROGRAMM</u>	<u>8</u>
2.1	BEGRIFFE UND KONZEPTE.....	8
2.2	LERNZIELE	8
<u>3</u>	<u>EINFÜHRENDE SOA-GRUNDBEGRIFFE: KURZÜBERBLICK.....</u>	<u>9</u>
3.1	BEGRIFFE UND KONZEPTE.....	9
3.2	LERNZIELE	9
3.3	REFERENZEN.....	11
<u>4</u>	<u>TECHNISCHE KONZEPTE IN EINER SOA</u>	<u>12</u>
4.1	BEGRIFFE UND KONZEPTE.....	12
4.2	LERNZIELE	12
4.3	REFERENZEN.....	14
<u>5</u>	<u>SOA: TECHNOLOGIEN UND TECHNOLOGIESTANDARDS.....</u>	<u>15</u>
5.1	BEGRIFFE UND KONZEPTE.....	15
5.2	LERNZIELE	15
5.3	REFERENZEN.....	17
<u>6</u>	<u>TECHNISCHES SERVICE-DESIGN.....</u>	<u>18</u>
6.1	BEGRIFFE UND KONZEPTE.....	18

6.2	LERNZIELE	18
6.3	REFERENZEN.....	20
7	<u>ANWENDUNGSARCHITEKTUREN IN EINER SOA</u>	21
7.1	BEGRIFFE UND KONZEPTE.....	21
7.2	LERNZIELE	21
7.3	REFERENZEN.....	21
8	<u>WERKZEUGE IN EINER SERVICE-ORIENTIERTEN IT-LANDSCHAFT</u>	22
8.1	BEGRIFFE UND KONZEPTE.....	22
8.2	LERNZIELE	22
8.3	REFERENZEN.....	23
9	<u>SOA-FALLBEISPIEL(E)</u>	24
9.1	BEGRIFFE UND KONZEPTE.....	24
9.2	LERNZIELE	24
9.3	REFERENZEN.....	24
10	<u>QUELLEN UND REFERENZEN DES MODULS SERVICE-ORIENTED ARCHITECTURE - TECHNISCH (SOA-T)</u>	25

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-Systemen mittlerer bis hoher Kritikalität technische und inhaltliche Verantwortung übernehmen.
- Konzeption, Entwurf und Dokumentation von Maßnahmen zur Erreichung nicht-funktionaler 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 den drei unterschiedlichen Kompetenzbereichen.
 - 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.

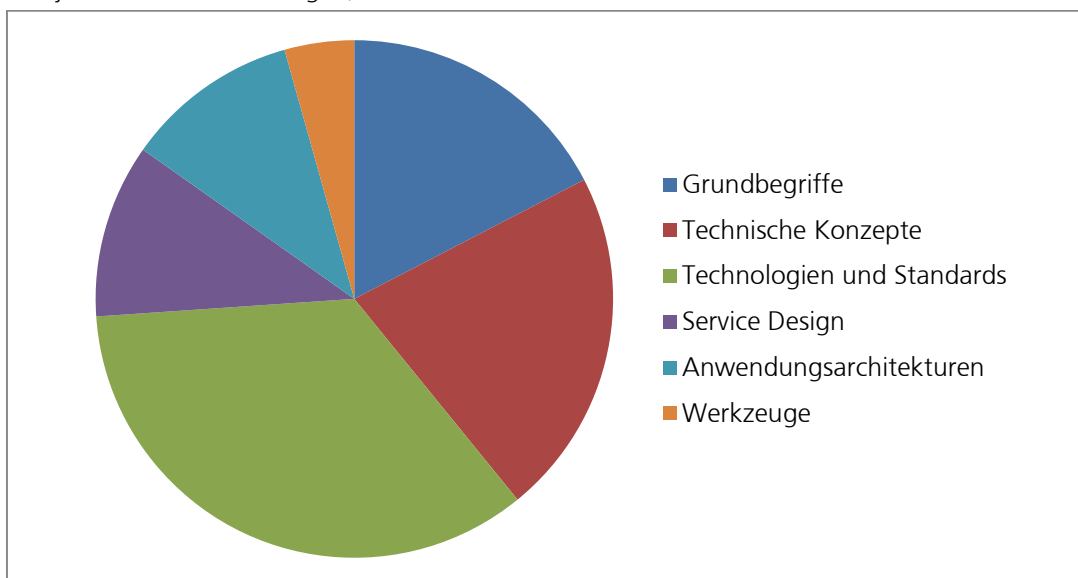


1 Grundlegendes zum Modul Service-Oriented Architecture - Technisch (SOA-T)

1.1 Gliederung des Lehrplans für das Modul Service-Oriented Architecture - Technisch (SOA-T)

- Einführende SOA-Grundbegriffe: Kurzüberblick (mind. 4h)
- Technische Konzepte in einer SOA (mind. 5h)
- SOA: Technologien und Technologiestandards (mind. 8h)
- Technisches Service-Design (mind. 2,5h)
- Anwendungsarchitekturen in einer SOA (mind. 2,5h)
- Werkzeuge in einer Service-orientierten IT-Landschaft (mind. 1h)
- SOA-Fallbeispiel(e) (mind. 1h)

(Zeiten jeweils inklusive Übungen)



1.2 Dauer, Didaktik und weitere Details

Die genannten Zeiten sind Empfehlungen. Die Dauer entsprechender Schulungen sollte mindestens 3 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 explizit offen.

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

1.3 Voraussetzungen für das Modul Service-Oriented Architecture - Technisch (SOA-T)

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.
- Grundlagen verteilter Systeme.
- Grundlagen der Prozessmodellierung.

Hilfreich für das Verständnis einiger Konzepte sind darüber hinaus:

- Kenntnisse über typische Herausforderungen bei Erstellung und Wartung (auch verteilter) kompositioneller Softwaresysteme:
 - Komponenten-Schnitt
 - Interface-Schnitt
 - Test
 - Fehleranalyse
- Kenntnis im Bereich „Enterprise Application Integration“

1.4 Gliederung des Lehrplans für Service-orientierte Software-Architektur

Die einzelnen Abschnitte des Lehrplans sind gemäß folgender Gliederung beschrieben:

1. **Begriffe/Konzepte:** Wesentliche Kernbegriffe dieses Themas.
2. **Unterrichts-/Übungszeit:** Legt die Unterrichts- und Übungszeit fest, die für dieses Thema bzw. dessen Übung in einer akkreditierten Schulung mindestens aufgewendet werden muss.
3. **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:
 - 3.1. 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.
 - 3.2. Was sollen die Teilnehmer **verstehen**? Diese Inhalte können geprüft werden.
 - 3.3. 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.

Zur besseren Gliederung und Nachvollziehbarkeit, werden ferner innerhalb der jeweiligen können/verstehen/kennen-Abschnitte inhaltliche Unterabschnitte gebildet.

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.

Gerade (auch) im Bereich SOA gibt es häufig mehrere Definitionen/Erläuterungen für Begriffe. Auf verwendete Referenzen wird in entsprechenden Abschnitten Bezug genommen.

1.6 Credit Points für diese Schulung

Vom iSAQB e. V. lizenzierte Schulungen nach diesem Lehrplan tragen zur Zulassung zur abschließenden Advanced-Level-Zertifizierungsprüfung folgende Punkte (Credit Points) bei:

Methodische Kompetenz:	10 Punkte
Technische Kompetenz:	20 Punkte

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 Levels,
- Randbedingungen und Vorgehen beim iSAQB-Zertifizierungsprogramm.

3 Einführende SOA-Grundbegriffe: Kurzüberblick

Dauer: 210 Min	Übungszeit: 30 Min
----------------	--------------------

3.1 Begriffe und Konzepte

SOA-Motivation, Intuitiver Service-Begriff, Grundprinzipien in einer SOA, Service-Orientierung, Prozess-Orientierung, Service-Klassifizierung, Service-Integration, Service-Komposition, Heterogene Systeme, Wiederverwendung, SOA-Governance, lose Kopplung.

3.2 Lernziele

Ziel dieses Abschnitts ist es, den Teilnehmern ein gemeinsames Verständnis für (vorwiegend technisch orientierte) Kernbegriffe einer SOA zu geben.

3.2.1 Was sollen die Teilnehmer können?

3.2.1.1 Intuitiver Service-Begriff & SOA-Definitionen

- Den Service-Grundbegriff „intuitiv“ als „Dienstleistung mit bestimmten Qualitätseigenschaften hinter einer klar trennenden Schnittstelle“ am Beispiel erklären können.

3.2.1.2 Grundprinzipien in einer SOA

- Wesentliche Grundprinzipien einer SOA verstehen und in Kurzform erklären können, besonders:
 - Geschäftsrelevanz (von Services und einer SOA),
 - Funktionale Zerlegung von Geschäftsbereichen in Form von Services,
 - Abstraktion technischer Details durch Services,
 - Service-Kontrakt,
 - Service-Auffindbarkeit,
 - Service-Komponierbarkeit,
 - Wiederverwendbarkeit von Services,
 - Grundbegriff der losen Kopplung von bzw. durch Services,
 - Statuslose vs. statusbehaftete Services.

3.2.1.3 Gegenüberstellung von fachlicher und technischer SOA

- Fachliche SOA: Ein Beispiel für fachliche Prozesse und Services in einer SOA erklären können.
- Technische SOA: Ein Beispiel für Prozesse und technisch implementierte Services in einer SOA erklären können.

3.2.1.4 Einfluss einer SOA auf IT-Prozesse und -Governance

- Mindestens einen typischen Service-Lebenszyklus verstehen und erklären können.
- Rollen und Skills im Rahmen der service-orientierten Software-Entwicklung benennen und erklären können.
- Den Begriff der SOA-Governance erklären können.
- Abgrenzung zwischen SOA-Governance, IT-Governance und Unternehmens-Governance darstellen können.
- Zusammenspiel von SOA-Governance, IT-Governance und Unternehmens-Governance erklären können.

3.2.1.5 Service-Grundbegriff

- Unterschied zwischen Geschäftsservice und technisch implementiertem Service erklären können.
- Grundaufbau eines technisch implementierten Service in Form von Service-Schnittstelle, Service-Implementierung und Service-Policies bzw. -Qualitätsmerkmalen erklären können.
- Service-Granularität verstehen und erklären können.

3.2.1.6 Service-Klassifizierung: Grundbegriffe

- Eine Klassifizierung von Services in „Basis-Services“ (Daten, Logik), „Composed-Services“ und „Prozess-Services“ verstehen und erklären können.

3.2.1.7 Service-Komposition: Grundbegriffe

- Das Prinzip der stufenweise Komposition von Services (zu „Composed Services“) verstehen und erklären können.
- Den Unterschied zwischen Orchestrierung und Choreographie von Services verstehen und erklären können.

3.2.1.8 Grundlegende (technische) SOA-Architekturen

- Das „fundamentale SOA-Dreieck“ (Service Consumer, Service Provider, Service Registry) verstehen und erklären können.

3.2.2 Was sollen die Teilnehmer verstehen?

3.2.2.1 Intuitiver Service-Begriff & SOA-Definitionen

- Verschiedene Ansätze zur Definition einer SOA kennen und erklären können.
- Den Unterschied zwischen klassischer Anwendung und Service-orientierter Anwendung verstehen.

3.2.2.2 Gegenüberstellung von fachlicher und technischer SOA

- Den Unterschied zwischen einer SOA in rein fachlicher Form vs. einer SOA mit technisch implementierten Services verstehen.

3.2.2.3 Einfluss einer SOA auf IT-Prozesse und -Governance

- Auswirkungen der Einführung einer SOA auf IT- und Unternehmensprozesse verstehen und wiedergeben können.

3.2.2.4 Service-Grundbegriff

- Abgrenzung Service vs. Web Service verstehen.
- Abgrenzung Service vs. Komponenten/Module verstehen.

3.2.2.5 Grundlegende (technische) SOA-Architekturen

- Erweiterungen des „fundamentale SOA-Dreiecks“ um typische Elemente wie Service Bus (technisch/abstrakt), Adapter, Security, Transaktionen, Orchestrierung und Management verstehen.

3.2.3 Was sollen die Teilnehmer kennen?

3.2.3.1 SOA-Motivation

- Verschiedene Motive für die Einführung von SOA kennen, wie u. a.: Heterogenität und historisches Wachsen bestehender IT-Systeme; Entkopplung der Geschäftsprozesse von IT-Systemen; Modularisierung von IT-Systemen; Flexibilität; Wiederverwendbarkeit.

3.2.3.2 Intuitiver Service-Begriff & SOA-Definitionen

- In Kurzform mehrere Definitionen einer SOA kennenlernen.

3.2.3.3 Service-Grundbegriff

- Die Teilnehmer sollen die Motivation für semantische Beschreibung von Services kennen, z. B. mittels Technologien wie OWL.

3.2.3.4 Service-Klassifizierung: Grundbegriffe

- In Kurzform einige weitere Klassifizierungen von Services kennenlernen.

3.3 Referenzen

[Beinhauer2008]
[Engels2008]
[Josuttis2008]
[Krafzig+2004]
[Liebhart2007]
[Mathas2007]
[Melzer2010]
[Oasis-SOA-2006]
[Starke2007]

4 Technische Konzepte in einer SOA

Dauer: 270 Min	Übungszeit: 30 Min
----------------	--------------------

4.1 Begriffe und Konzepte

Kommunikationsarten und Interaktionsmuster, Enterprise Service Bus, REST, Service-Beschreibung, Service Auffinden, Querschnittsaspekte, Service-Komposition, Service Registry, Service Repository, Abstrakte technische SOA-Referenzarchitekturen, SOA-typische technische Spezialkonzepte.

4.2 Lernziele

Ziel dieses Abschnitts ist es, den Teilnehmern wesentliche technische Konzepte in einer SOA zu vermitteln. Hierzu gehören zum Beispiel verschiedene Kommunikationsmechanismen, Service-Schnittstellenbeschreibungsvarianten, Quality of Service-Aspekte, Service Registry usw.

4.2.1 Was sollen die Teilnehmer können?

4.2.1.1 Kommunikation mittels Nachrichten, Ressourcen und Methodenaufrufen

- Die Grundkonzepte der Kommunikation mittels Nachrichten, Ressourcen und Methodenaufrufen verstehen und erklären können.
- Die Unterschiede zwischen der Kommunikation mittels Nachrichten, Ressourcen und Methodenaufrufen verstehen und erklären können.
- Vor- und Nachteile der Kommunikation mittels Nachrichten, Ressourcen und Methodenaufrufen vergleichen können und verstehen. Vergleiche sollen nach typischen Merkmalen wie Leichtgewichtigkeit, Security, Einfachheit der Implementierung u. a. m. erfolgen.

4.2.1.2 Kommunikation: Interaktionsmuster

- Verschiedene Arten der Interaktion (z. B. synchron oder asynchron) benennen und erklären können.
- Verschiedene Interaktionsmuster (z. B. RPC, Messaging, Event-Handling, ...) benennen und erklären können.
- Typische Arten von Nachrichtenaustauschmustern (Request-Response, One-Way, Request Callback, Notification, usw.) benennen und erklären können.
- Den Einsatz von einfachen und komplexen Datentypen in Interaktionsmustern benennen und die Auswirkungen ihres Einsatzes erklären können.
- Auswirkungen von Interaktionsmustern auf die Gestaltung von Service-Schnittstellen verstehen und erklären können.

4.2.1.3 Service Beschreibung

- Den Inhalt und den Zweck von Service-Beschreibungen benennen und erklären können.
- Die Service-Beschreibung in einen Kontext (z. B. Web Services, Rest, ...) einordnen und in diesem Kontext erklären können.
- Regeln (Namenkonventionen, Trennung der Fachlichkeit und Technik, Zustands-Konsistenz, Redundanz- und Überschneidungsfreiheit, ...) für Service-Schnittstellen benennen und erklären können.
- Den Einsatz von verschiedenen Datentypen für Schnittstellen (einfache Datentypen, komplexe Datentypen) diskutieren und erklären können.

4.2.1.4 Service Auffinden

- Das Konzept und die Funktionsweise von Service Registry und Service Repository verstehen und erklären können.
- Aufgaben, Ziele und Einsatzmöglichkeiten von Service Registry und Service Repository verstehen und erklären können.

4.2.1.5 Querschnittsaspekte

- Wesentliche technische Querschnittsaspekte in einer SOA verstehen und erklären können, mindestens:
 - Security
 - Sicherheitsaspekte in einer SOA (z. B. Identifizierung, Authentifizierung, Autorisierung) erklären können.
 - Den Einsatz von Sicherheitsmechanismen in verschiedenen Anwendungsszenarien (z. B. Single Sign On, ...) erklären und argumentieren können.
 - Transaktionen: Den Einsatz verschiedener Transaktionsarten erklären können:
 - Single Resource ACID-Transaktionen,
 - XA-Transaktionen,
 - Langlaufende Transaktionen.
 - Service Level Agreements (SLAs)
 - Die Notwendigkeit von Service Level Agreements (SLAs) verstehen und typische Inhalte (z. B. Vertragsparteien, rechtliche Regelung, Ausnahmen und Einschränkungen usw.) eines SLAs erklären können.
 - Service Level-Attribute erklären können, u. a. bzgl. Verfügbarkeit und Performance (Antwortzeiten, Transaktionsrate).
 - Das Konzept und die Rolle von Service Policies zur Beschreibung der nicht-funktionalen Eigenschaften von Services erklären können.

4.2.1.6 Service-Komposition

- Den Begriff der Service-Komposition verstehen und erklären können.
- Herausforderungen bei der Service-Komposition verstehen und erklären können.
- Den Zusammenhang zwischen Geschäftsprozessen und Service-Komposition benennen und erklären können.
- Den Unterschied zwischen Orchestrierung und Choreographie verstehen und erklären können.
- Die Vor-/Nachteile von Service-Komposition bewerten und beschreiben können.

4.2.1.7 Abstrakte technische SOA-Referenzarchitekturen

- Die Teilnehmer sollen verschiedene SOA-Referenzarchitekturen benennen und erklären können, mindestens: Enterprise Service Bus (ESB-artig), Ressourcenorientierte Integration (REST-artig).

4.2.2 Was sollen die Teilnehmer verstehen?

4.2.2.1 Kommunikation: Interaktionsmuster

- Typische Anwendungsszenarien für verschiedene Interaktionsmuster beschreiben und Argumente für die Auswahl der Interaktionsmuster für spezifische Anwendungsszenarien verstehen.
- Typische Anwendungsszenarien für den Einsatz von einfachen und komplexen Datentypen verstehen.

4.2.2.2 Service Auffinden

- Die Abgrenzung von Service Registry und Service Repository verstehen.

4.2.2.3 Querschnittsaspekte

- Konzepte zum Exception Handling in einer SOA verstehen,
- Konzepte zum Logging in einer SOA verstehen.

4.2.2.4 Service-Komposition

- Die Einsatzmöglichkeiten von Service Registry und Service Repository bei der Service-Komposition verstehen.
- Den Zusammenhang von Querschnittsaufgaben (z. B. Security, Transaktion, SLAs, Verfügbarkeit, Performance, ..) mit der Service-Komposition verstehen.

4.2.2.5 Integration bestehender Anwendungen

- Aspekte und Problemfelder bei der Integration der bestehenden Anwendungen verstehen und erklären können.
 - Kapselung bestehender Anwendungen.
 - Konzept von Adaptern bzw. Konnektoren.

4.2.2.6 SOA Clients

- Die Nutzung von Services in einer SOA mit verschiedenen Client-Typen verstehen, wie: Web Client, Rich Client u. a.,
- Session-Konzepte im Kontext einer SOA verstehen.

4.2.3 Was sollen die Teilnehmer kennen?

4.2.3.1 Abstrakte technische SOA-Referenzarchitekturen

- Weitere technische SOA-Referenzarchitekturen kennen, wie: Nachrichtenorientierte SOA-Architekturen und modellgetriebene SOA-Ansätze.

4.2.3.2 Integration bestehender Anwendungen

- Einige konkrete Beispiele verschiedener Adapter/Konnektoren kennen, wie: Java Enterprise Connector Architecture, Datenbankadapter u. a. m.

4.2.3.3 Spezialkonzepte in SOA-Umgebungen und SOA-Plattformen

- Typische Spezialkonzepte in einer SOA kennenlernen, wie:
 - Transformationsmechanismen,
 - Master Data Management,
 - Konzepte für die Übertragung großer Nachrichten mittels Services,
 - Interceptor,
 - Intelligent Routing,
 - Monitoring zur Überwachung von Services bzgl. Verfügbarkeit, Performanz, Einhaltung von Policies und SLAs u. a. m.,
 - Spezielle Mechanismen für Fehlertoleranz und Skalierbarkeit für technisch implementierte Services.

4.3 Referenzen

[Erl2008]

[Josuttis2008]

[Krafzig+2004]

[Liebhart2007]

[Mathas2007]

[Melzer2010]

[W3C]

5 SOA: Technologien und Technologiestandards

Dauer: 450 Min

Übungszeit: 30 Min

5.1 Begriffe und Konzepte

Web Services, WSDL, SOAP, XML, XSLT, HTTP, REST, JSON, UDDI, BPEL, BPMN, Java, .NET, ESB, CRUD.

5.2 Lernziele

5.2.1 Was sollen die Teilnehmer können?

5.2.1.1 Web Services

- SOAP zur Schnittstellenbeschreibung benennen und erklären können.
- WSDL zur technischen Schnittstellenbeschreibung benennen und erklären können.
- Technische Komposition von Web Services benennen und erklären können.
- Wichtige WS-* Spezifikationen benennen und überblicksartig erklären können.
 - WS-Security,
 - WS-Reliable Messaging,
 - WS-Transactions,
 - WS-Policy.

5.2.1.2 Kommunikationsparadigmen

- Technische Bestandteile eines ESB benennen und erklären können.
- Technische Nutzung eines ESB benennen und erklären können.
 - Service-Schnittstellen,
 - Services an den ESB anbinden und über den ESB aufrufen.
- Technische Umsetzung von REST benennen und erklären können.
 - Serviceoperationen (Create, Retrieve, Update, Delete: CRUD).
- Nutzung von XML als Nachrichtenformat am Beispiel benennen und erklären können.
 - XML Validierung mittels XSD,
 - XML Transformation mittels XSLT.

5.2.1.3 Geschäftsprozesse

- Technische Bestandteile einer Execution Engine für BPMN benennen und erklären können.
- Technische Nutzung einer Execution Engine für BPMN benennen und erklären können.
 - Integration der Execution Engine in einen SOA-Technologie-Stack.

5.2.2 Was sollen die Teilnehmer verstehen?

5.2.2.1 Web Services

- Abgrenzung zwischen SOAP und WSDL erklären können.
- Überblicksartig WS-* Spezifikationen als Sammlung von Standards im Web Service Umfeld benennen und an Beispielen erklären können.
- Vergleich SOAP-/WSDL-basierter Web Services vs. REST-basierte Services bzgl.
 - Anzahl Services, Anzahl Service-Operationen,
 - Technischen Qualitätsaspekten, besonders Security, Transaktionen, Skalierbarkeit.

5.2.2.2 Kommunikationsparadigmen

- Logische Bestandteile eines ESB benennen und erklären

- Logische Nutzung eines ESB als Kommunikationsplattform benennen und erklären
 - Rolle des ESB als Integrationsplattform
 - Zusammenspiel verschiedener ESBs
 - API-getriebene ESBs
 - Protokoll-getriebene ESBs
 - Sicherheitsaspekte bei der Nutzung eines ESB
- Logische Nutzung von REST als Kommunikationsparadigma und als Architekturmodell benennen und erklären
 - HTTP als Anwendungsprotokoll für REST
 - Sicherheitsaspekte bei REST
 - RESTful Web Services
 - JSON als Datenformat
- Nachrichten-basierte Kommunikation an Beispielen benennen und erklären
 - Beispiel: JMS
- Umgang mit XML als Nachrichtenformat benennen und erklären
 - XML-Schema (XSD) als Beschreibung der Datenstruktur
 - XSLT zur Nachrichtentransformation

5.2.2.3 Geschäftsprozesse

- Einsatz der BPMN zu Modellierung von Geschäftsprozessen benennen und erklären

5.2.3 Was sollen die Teilnehmer kennen?

5.2.3.1 Services

- Technische Verwaltung von Services mittels Service Registry verstehen
 - Universal Description, Discovery and Integration (UDDI)
- Fachliche Verwaltung von Services mittels Service Repository verstehen

5.2.3.2 Web Services

- Anwendungsbeispiele für Web Services kennen
- Beispiele für Referenzarchitekturen kennen
- Beispiele für die technische Umsetzung von Web Services kennen
 - Mittels Java (Standard und Enterprise Edition)
 - JAX-WS
 - Mittels .Net
 - Mittels REST und mittels RESTful Web Services
- Mehrere weitere Beispiele für SOA-Implementierungstechnologien im Kurzüberblick kennen
 - Mindestens drei weitere Beispiele wie: EAI Service Broker, CORBA, Generatoren für verschiedene Service-Technologien, Eigenentwicklungen.
- Möglichkeiten Web Services semantisch zu beschreiben
 - Web Service Description Language – Semantics (WSDL-S), Web Ontology Language for Web Services (OWL-S), Web Service Modelling Language (WSML), Semantic Web Service Language (SWSL), Semantic Web Service Ontology (SWSO), Unified Service Description Language (USDL).

5.2.3.3 Geschäftsprozesse

- BPEL als Execution Language für BPMN kennen
- BPEL Execution Engines beispielhaft kennen.

5.3 Referenzen

[Erl2008]
[Josuttis2008]
[Krafzig+2004]
[Liebhart2007]
[Mathas2007]
[Melzer2010]
[Starke2007]
[W3C]

6 Technisches Service-Design

Dauer: 150 Min	Übungszeit: 60 Min
----------------	--------------------

6.1 Begriffe und Konzepte

Service Life Cycle, Design, Implementierung, Test, Betrieb von Services.

6.2 Lernziele

Ziel dieses Abschnittes ist es, den Teilnehmern Vorgehensweisen beim Design von Services besonders unter technischen Gesichtspunkten zu vermitteln. Es gibt für „technisches Service-Design“ keine allgemeingültige, scharfe Begriffsdefinition und kein alleingültiges Vorgehensmodell gibt. Behandelt werden hier deshalb anhand eines Service Life Cycle typische, wesentliche Themen aus Beschreibung, Analyse, Design, Implementierung und Test von Services.

6.2.1 Was sollen die Teilnehmer können?

6.2.1.1 Überblick Service Life Cycle

- Verschiedene technische Artefakte eines Service benennen und erklären können:
 - Technische Service-Beschreibung,
 - Service-Implementierung,
 - Service-Tests,
 - Technische Aspekte einer Service-Nutzungsvereinbarung.

6.2.1.2 Analyse

- Überblicksartig die Geschäftsprozessmodellierung mit BPMN benennen und erklären können.

6.2.1.3 Design

- Technische Umsetzung der Anforderungen an einen Service:
 - Umsetzung des Offen-Geschlossen-Prinzips,
 - Technische Möglichkeiten verschiedene Granularitäten in den Service-Schnittstellen umzusetzen.

6.2.1.4 Implementierung

- Technische Aspekt der Service-Entwicklung benennen und erklären können:
 - Komponententests im Service-Umfeld,
 - Logging-Möglichkeiten im Service-Umfeld,
 - Integration eines ESB in eine Entwicklungsumgebung (Integration, Konfiguration)
 - Infrastruktur für REST.
- Einsatz der Code-Generierung im Entwicklungsprozess benennen und beschreiben.
- Technische Möglichkeiten und Voraussetzungen zur Konfiguration von Services
 - Beim Deployment,
 - Im laufenden Betrieb.
- Paketierung der Services und technische Vorbereitung für die Übergabe an den Betrieb.

6.2.1.5 Test

- Technische Aspekt des Testens von Services benennen und erklären können.
 - Berücksichtigung der Service-Nutzungsvereinbarung.
 - Frühzeitiges Testen gegen Mock-ups.
- Einsatz von Test-Frameworks für Services benennen und erklären können.

6.2.2 Was sollen die Teilnehmer verstehen?

6.2.2.1 Überblick Service Life Cycle

- Erzeugung, Bearbeitung und Bedeutung der Service-Artefakte für die einzelnen Phasen des Softwareentwicklungsprozesses verstehen.

6.2.2.2 Analyse

- Vorgehensweisen zur Identifikation von Services/Service-Kandidaten im Analyseprozess verstehen.

6.2.2.3 Design

- Einige Praktiken um einen Service-Schnitt zu definieren verstehen.
- Möglichkeiten, die Anforderungen potentieller Servicenutzer bei der Erstellung eines Services zu berücksichtigen, verstehen.
- Verantwortung für einen Service in Abgrenzung zur Verantwortung für ein Projekt
 - Nutzung eines bestehenden Services,
 - Erzeugung eines neuen Services im Projektrahmen.
- Beispiele für Randbedingungen im Design beim Einsatz von Codegenerierung verstehen.

6.2.2.4 Implementierung

- Technische Voraussetzungen und Möglichkeiten zur Überwachung der Service Qualität verstehen.
- Technische Konzepte, um mit Versionswechseln bei Services umgehen zu können, verstehen.
- Bedeutung der Service Registry als Hilfe für die technische Nutzung bestehender Services verstehen.

6.2.2.5 Test

- Möglichkeiten und Voraussetzungen für den Einsatz von Testgeneratoren verstehen.
- Technische Voraussetzungen für die Durchführung der verschiedenen Testphasen verstehen.
- Analysemöglichkeiten verstehen, um Fehlerursachen nachvollziehen zu können.

6.2.2.6 Betrieb

- Typische technische Möglichkeiten Services zu überwachen im Überblick verstehen.

6.2.3 Was sollen die Teilnehmer kennen?

6.2.3.1 Überblick Service Life Cycle

- Verschiedene fachliche Artefakte eines Services kennen
 - Fachliche Service-Beschreibung,
 - Fachliche Aspekte einer Service-Nutzungsvereinbarung.

6.2.3.2 Analyse

- Beispiele für Auswirkungen bzw. Rückkopplungen der Serviceorientierung auf ein Geschäftsprozessmodell kennenlernen.
- Fachliche Beschreibung der Anforderungen an Services/Service-Kandidaten kennen.

6.2.3.3 Design

- Beispiele für Service-Nutzungsvereinbarungen kennen.
- Beispiele für Vorgehensweisen um verschiedene technische Lösungen für die Interaktion der Services zu vergleichen, zu bewerten und die passende auszuwählen kennen.

6.2.3.4 Test

- Organisatorische Aspekte des Testens von Services kennen.
 - Integration vieler Beteiligter bis hin zu einer Partei pro Service.
 - Berücksichtigung des Service-Mix (Verschaltung von Umgebungen in verschiedenen Stadien – Entwicklung, Integration, Test, Wartung, Produktion).

6.2.3.5 Betrieb

- Exemplarisch mindestens einen Betriebsprozess, um Servicenutzung zu überwachen und Verletzung der Vereinbarung zu sanktionieren, kennen.
- Beispiele für fachliche und technische Artefakte, die von der Entwicklung an den Betrieb übergeben werden kennen.

6.3 Referenzen

[Erl2008]

[Josuttis2008]

[Liebhart2007]

[Mathas2007]

[Melzer2010]

7 Anwendungsarchitekturen in einer SOA

Dauer: 120 Min	Übungszeit: 90 Min
----------------	--------------------

7.1 Begriffe und Konzepte

Frontend, Datenquellen, Bestandssysteme, Integration, Geschäftslogik, SOA-Governance.

7.2 Lernziele

Kennenlernen von typischen Beispielen für Anwendungsarchitekturen mittels SOA. Entwurf einer eigenen Architektur als Kleingruppenarbeit.

7.2.1 Was sollen die Teilnehmer können?

- Bestandteile einer service-orientierten Anwendungsarchitektur benennen und erklären.
- Technische Integration verschiedener Arten von Clients in eine SOA.
- Beispiele für Anwendungsarchitekturen benennen und erklären, die die Geschäftslogik (Teile der Geschäftslogik) der Anwendung in unterschiedlichen Bereichen der Anwendung realisieren.
- Technische Integration von Datenquellen in eine Anwendungsarchitektur in einer SOA darstellen und erklären können.
- Eine SOA-Anwendungsarchitektur in groben Zügen entwerfen und erklären können.
- Verschiedene Möglichkeiten zur Integration von Bestandssystemen in eine Anwendungsarchitektur in einer SOA benennen und erklären können.

7.2.2 Was sollen die Teilnehmer verstehen?

- Vor- und Nachteile der Realisierung der Geschäftslogik in verschiedenen Teilen der Anwendung verstehen.
- Realisierung von Sicherheitsaspekten in verschiedenen Anwendungsarchitekturen verstehen.
- Aufgaben der SOA-Governance bei der Realisierung einer Anwendungsarchitektur benennen und erklären.
- Berücksichtigung von Querschnittsaspekten beim Entwurf der eigenen Architektur.

7.2.3 Was sollen die Teilnehmer kennen?

- Verschiedene Beispiele für Anwendungsarchitekturen in einer SOA,
- Beispiele für Technologien für die Erstellung von Frontends in einer SOA.

7.3 Referenzen

[Erl2008]

[Josuttis2008]

[Liebhart2007]

[Mathas2007]

[Melzer2010]

8 Werkzeuge in einer Service-orientierten IT-Landschaft

Dauer: 60 Min	Übungszeit: 0 Min
---------------	-------------------

8.1 Begriffe und Konzepte

Die folgenden Kategorien von Werkzeugen sind wichtige Arbeitsmittel für Service-orientierte Architekturen. Dieser Abschnitt gibt einen kurzen Eindruck über typische Werkzeuge in einer SOA. Es geht in diesem Abschnitt nur um ein erstes kennenlernen solcher Arten von Werkzeugen sowie einiger Beispiele solcher Werkzeuge. Die Auswahl der Werkzeugbeispiele liegt beim Schulungsanbieter, Sie sind nicht prüfungsrelevant.

8.2 Lernziele

8.2.1 Was sollen die Teilnehmer können?

- Wichtigste Kategorien und Werkzeuge sowie deren Stärken und Schwächen benennen und erklären.

8.2.2 Was sollen die Teilnehmer verstehen?

- Einfluss verschiedener Randbedingungen und Einflussfaktoren auf die Arbeitsumgebung und die Wahl der Arbeitsmittel verstehen.

8.2.3 Was sollen die Teilnehmer kennen?

- Typische wichtige Vertreter der oben genannten Kategorien kennen.

8.2.3.1 Modellierungswerkzeuge

Unterstützung bei den folgenden Aufgaben:

- Entwurf und Kommunikation von Service-orientierten Architekturen,
- Abbildung von Geschäftsprozessen,
- Vorgabe für detailliertes Design und Implementierung, eventuell auch Codegenerierung.

8.2.3.2 Applikationsserver

Unterstützung bei den folgenden Aufgaben:

- Technische Zusatzdienste wie zum Beispiel Autorisierung, Logging, Monitoring, Sicherheit, Zuverlässigkeit, Auditing, Transaktionsmanagement oder Verwaltung,
- Unterstützung bzw. Anbindung an BPEL-Engine zur Ausführung von Geschäftsprozessen.

8.2.3.3 Enterprise Service Bus

Unterstützung bei den folgenden Aufgaben:

- Kommunikation zwischen verschiedenen Applikationen,
- Integration verschiedener Applikationen,
- Unterstützung zusätzlicher Konzepte wie zum Beispiel Schnittstellenanpassungen, Datentransformationen, flexibles Routing oder Master Data Management.

8.2.3.4 SOA Registry und Repository

Unterstützung bei den folgenden Aufgaben:

- Steuerung und Kontrolle einer wachsenden SOA,
- Verwaltung von Schnittstellenbeschreibungen und zahlreicher weiterer Serviceinformationen,
- Unterstützung des gesamten Lebenszyklus von Prozessen, Policies und Services,
- Bereitstellung eines Verzeichnis oder Katalogs verfügbarer Dienste,
- Unterstützung für Design Time oder Business Intelligence,
- Speichern und Kategorisieren von Services mit ihren zugehörigen Artefakten,
- Regelung der Zusammenarbeit von Management- und Monitoring-Werkzeugen.

8.2.3.5 Werkzeuge zum Service-Management

Unterstützung bei den folgenden Aufgaben:

- Erfassung der organisatorischen Verantwortlichkeiten,
- Klassifizierung der Services,
- Bereitstellung von nachvollziehbaren Lebenszyklen,
- Planung der Einführung verschiedener Dienste,
- Verwaltung verschiedener Versionen desselben Service,
- Testen von Diensten.
- Zuordnen von Service Level Agreements.

8.2.3.6 SOA-Plattformen

- Beispiele für Plattformen, die Werkzeuge als ganzes Bündel für Entwurf, Entwicklung, Test und Betrieb von Service-orientierten Architekturen anbieten.

8.3 Referenzen

Keine.

9 SOA-Fallbeispiel(e)

Dauer: 60 Min	Übungszeit: 0 Min
---------------	-------------------

9.1 Begriffe und Konzepte

Innerhalb jeder akkreditierten Schulung muss mindestens ein technisches Beispiel einer größeren SOA vorgestellt und besprochen werden.

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

9.2 Lernziele

Vorstellung und eventuell Erarbeitung mindestens eines größeren Beispiels einer SOA.

9.2.1 Was sollen die Teilnehmer können?

Nicht zutreffend.

9.2.2 Was sollen die Teilnehmer verstehen?

Nicht zutreffend.

9.2.3 Was sollen die Teilnehmer kennen?

Die Teilnehmer sollen einen realitätsnahen Einblick in die wesentlichen technischen Aspekte einer größeren SOA anhand eines oder mehrerer konkreter praxisnaher Beispiele bekommen.

9.3 Referenzen

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

10 Quellen und Referenzen des Moduls Service-Oriented Architecture - Technisch (SOA-T)

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

B

[Beinhauer2008]

Beinhauer, W.: SOA für agile Unternehmen. Symposion Publishing, 2008.

E

[Engels2008]

Engels, G., Hess, A., Humm, B., Juwig, O., Lohmann, M., Richter, J.-P.: Quasar Enterprise: Anwendungslandschaften serviceorientiert gestalten. Dpunkt, 2008.

[Erl2008]

Erl, T.: SOA Design Patterns. Prentice Hall, 2008. Online: <http://soapatterns.org/>

J

[Josuttis2008]

Josuttis, N.: SOA in der Praxis - System-Design für verteilte Geschäftsprozesse. Dpunkt, 2008.

K

[Krafzig+2004] Dirk Krafzig, Karl Banke & Dirk Slama: Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall, 2004.

L

[Liebhart2007]

Liebhart, D.: SOA goes real: Service-orientierte Architekturen erfolgreich planen und einführen. Carl Hanser Verlag, 2007.

M

[Mathas2007]

Mathas, C.: SOA intern: Praxiswissen zu serviceorientierten IT-Systemen. Carl Hanser Verlag, 2007.

[Melzer2010]

Melzer, I.: Service-orientierte Architekturen mit Web Services: Konzepte - Standards – Praxis. Spektrum Akademischer Verlag, 2010.

O

[Oasis-SOA-2006]

MacKenzie, C. M., K. Laskey, F. McCabe, P. F. Brown, R. Metz. Reference Model for Service Oriented Architecture - OASIS Standard, 2006.

[online] <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf> (Zugriff September 2012)

S

[SOA Know-how]

Bitkom Servicegesellschaft mbH: SOA Know-how. Online: <http://soa-know-how.de>

[Starke2007]

Starke, G.: SOA-Expertenwissen - Praxis, Methoden und Konzepte serviceorientierter Architekturen. Dpunkt, 2007.

W

[W3C]

World Wide Web Consortium (W3C): Online: <http://www.w3.org/>