

Curriculum für
CPSA Certified Professional for
Software Architecture®
– Advanced Level –

**Modul:
Architekturbewertung**



Version 1.5 (Februar 2015)

© (Copyright), International Software Architecture Qualification Board e. V.
(iSAQB® e. V.) 2009-2015

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

0	<u>EINLEITUNG: ALLGEMEINES ZUM ISAQB ADVANCED LEVEL</u>	5
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
1	<u>GRUNDLEGENDES ZUM MODUL ARCHITEKTURBEWERTUNG</u>	6
1.1	GLIEDERUNG DES LEHRPLANS FÜR ARCHITEKTURBEWERTUNG UND EMPFOHLENE ZEITLICHE AUFTEILUNG	6
1.2	DAUER, DIDAKTIK UND WEITERE DETAILS	7
1.3	VORAUSSETZUNGEN FÜR DAS MODUL „ARCHITEKTURBEWERTUNG“	7
1.4	GLIEDERUNG DES LEHRPLANS FÜR ARCHITEKTURBEWERTUNG	7
1.5	ERGÄNZENDE INFORMATIONEN, BEGRIFFE, ÜBERSETZUNGEN	7
1.6	CREDIT POINTS FÜR DIESE SCHULUNG	8
2	<u>EINFÜHRUNG IN DAS ISAQB ZERTIFIZIERUNGSPROGRAMM</u>	9
2.1	BEGRIFFE UND KONZEPTE	9
2.2	LERNZIELE	9
3	<u>GRUNDBEGRIFFE VON SOFTWARE-ARCHITEKTURBEWERTUNG</u>	10
3.1	BEGRIFFE UND KONZEPTE	10
3.2	LERNZIELE	10
3.3	REFERENZEN	11
4	<u>ANFORDERUNGEN IN DER ARCHITEKTURBEWERTUNG</u>	12
4.1	BEGRIFFE UND KONZEPTE	12
4.2	LERNZIELE	12
4.3	REFERENZEN	13
5	<u>VORGEHEN BEI DER BEWERTUNG</u>	14
5.1	BEGRIFFE UND KONZEPTE	14
5.2	LERNZIELE	14
5.3	REFERENZEN	15
6	<u>NACHARBEIT UND ERGEBNISVERWERTUNG</u>	16

6.1	BEGRIFFE UND KONZEPTE.....	16
6.2	LERNZIELE	16
6.3	REFERENZEN.....	16
7	<u>BEWERTUNG BESTEHENDER SYSTEM(-TEIL)E.....</u>	17
7.1	BEGRIFFE UND KONZEPTE.....	17
7.2	LERNZIELE	17
7.3	REFERENZEN.....	17
8	<u>BEISPIELE FÜR DIE BEWERTUNG VON SOFTWARE-ARCHITEKTUREN</u>	19
8.1	BEGRIFFE UND KONZEPTE.....	19
8.2	LERNZIELE	19
8.3	REFERENZEN.....	19
9	<u>QUELLEN UND REFERENZEN ZU SOFTWARE-ARCHITEKTURBEWERTUNG.....</u>	20

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.
- Architekturrelevante 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 Open Source Projekten)
- Aus- und Weiterbildung im Rahmen von iSAQB-Advanced Level Schulungen im Umfang von mindestens 7 Credit-Points aus mindestens zwei unterschiedlichen Kompetenzbereichen (detailliert geregelt in Abschnitt).
 - Bestehende Zertifizierungen (etwa Sun/Oracle Java-Architect, Microsoft-CSA o.ä.) können auf Antrag auf diese Credit-Points angerechnet werden.
 - 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 Architekturbewertung

1.1 Gliederung des Lehrplans für Architekturbewertung und empfohlene zeitliche Aufteilung

- Grundbegriffe von Softwarearchitekturbewertung (mind. 2h)
 - Anforderungen in der Architekturbewertung (mind. 2.5h)
 - Bewertungsworkshop (mind. 4h)
 - Nacharbeit und Ergebnisverwertung (mind. 1.5h)
 - Bewertung bestehender System(-teile) (2.5h)
 - Beispiele für die Bewertung von Softwarearchitekturen (mind. 1.5h)
- (Zeiten jeweils inklusive Übungen)



- Grundbegriffe von Software-Architekturbewertung
- Anforderungen in der Architekturbewertung
- Bewertungsworkshops
- Nacharbeit und Ergebnisverwertung
- Bewertung bestehender System(-teile)
- Beispiele für die Bewertung von Software-Architekturen

Detaillierte Gliederung mit den wichtigsten Themen:



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 „Architekturbewertung“

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 von Software, insbesondere der Architektur von Software- oder Software-nahen Systemen.

Hilfreich für das Verständnis einiger Konzepte sind darüber hinaus:

- Kenntnis typischer Herausforderungen beim Entwurf von Softwarearchitekturen:
 - Auswahl geeigneter Dokumentationsstrukturen, Notationen, Ergebnistypen (Stakeholderorientierung)
 - Gemeinsame Arbeit an grundlegenden Konzepten der zu erstellenden Software
 - Orientierung von Softwarearchitektur an Anforderungen und Rahmenbedingungen
 - Messung und Evaluation der Qualität von Entwürfen und Entscheidungen

1.4 Gliederung des Lehrplans für Architekturbewertung

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 selbstständig 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)	Davon 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 Software-Architekturbewertung

Dauer: 90 Min	Übungszeit: 30 Min
---------------	--------------------

3.1 Begriffe und Konzepte

Software-Architektur, Software-Architekturbewertung, übergreifende Konzepte/Aspekte, Architekturziele, nichtfunktionale und funktionale Anforderungen an Systeme, Einflussfaktoren, Bewertungsmethoden

3.2 Lernziele

3.2.1 Was sollen die Teilnehmer können?

3.2.1.1 Nutzen und Ziele von Software-Architektur

Eigenschaften von Architekturarbeit benennen

- Architekturarbeit ist grundlegend, hat im Projekt breite Auswirkungen und entsprechende Entscheidungen sind schwer zurückzunehmen
- Architekturarbeit sollte iterativ erfolgen und Feedbackschleifen vorsehen

Ziele von Software-Architekten und Software-Architekturen benennen und erklären

- Fokus von Software-Architektur liegt auf Qualitätsmerkmalen wie Langlebigkeit, Wartbarkeit, Änderbarkeit als Differenzierung gegenüber reiner Funktionalität.
- Architekturziele sind in der Regel langfristig, Projektziele oftmals kurzfristig

3.2.1.2 Nutzen und Ziele von Architekturbewertung

Nutzen von Architekturbewertung benennen und erklären

- Sicherstellung der Erreichung von zentralen Architekturzielen
- (Frühe) Erkennung von zentralen Risiken und Problemen bezüglich der geforderten Qualitätsmerkmale
- Methodische Absicherung von Architekturentscheidungen – Sicherung der langfristigen Nachvollziehbarkeit der Architektur
- Qualifiziertes Feedback zu Architekturentscheidungen
- Generell: Förderung von Kommunikation und Transparenz (Projektintern und mit externen Stakeholdern)

Positive Effekte von Architekturbewertungen auf das Vorgehen bei der Architekturentwicklung und die Architekturdokumentation benennen und erklären

3.2.1.3 Voraussetzungen für Architekturbewertung

Organisatorische Voraussetzungen benennen und erklären

- Notwendige Artefakte der Softwarearchitektur benennen und in konkreten Projektsituationen den notwendigen Detaillierungsgrad dieser Artefakte erkennen
- Rahmenbedingungen
- Architektur Anforderungen: konkrete qualitative Anforderungen
- Dokumentierte Architekturentscheidungen
- Architekturüberblick

3.2.1.4 Architekturbewertungsmethoden

Verschiedenen Methoden und Ansätze zur Architekturbewertung klassifizieren und erklären

- Szenarienbasierte Bewertungsmethoden
- Metrikbasierte Bewertung
- Expertenreviews (informell)

Besonderheiten von Szenarienbasierter Architekturbewertung benennen, insbesondere in Bezug auf die Ziele die Architekturbewertung.

Unterschied von qualitativen und quantitativen Techniken und die unterschiedliche Zielsetzung dieser Techniken erklären

3.2.2 Was sollen die Teilnehmer verstehen?

Software-Architekturen unterstützen die Erreichung nichtfunktionaler Qualitätsmerkmale.

Qualitätsanforderungen (Required constraints) wie Änderbarkeit, Effizienz, Sicherheit etc. beeinflussen Software-Architekturen in der Regel mehr als funktionale Anforderungen.

Software-Architekturen müssen aufgrund inhärenter Unsicherheit oftmals iterativ entwickelt werden. Die systematisch eingeholte Rückmeldung von Stakeholdern ist essentiell.

3.2.3 Was sollen die Teilnehmer kennen?

Wichtige Methoden zur Architekturbewertung (ohne genauen Ablauf)

- ATAM – Architecture Tradeoff Analysis Method
- CBAM – Cost-Benefit Analysis Method
- Vorgehen bei Codeanalyse
- Für mindestens eine Technologie: Werkzeuge zur Codeanalyse (Struktur- und Abhängigkeitsanalyse)

3.3 Referenzen

[Clements+2002]

[Kazman+ 2000]

[Bachmann 2000]

[Bass+2003]

[Clements+2003]

[Kruchten 1995]

[Starke 2008]

4 Anforderungen in der Architekturbewertung

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

4.1 Begriffe und Konzepte

Qualität, Qualitätsmerkmale, DIN/ISO 9126, Szenarien, Qualitätsbaum, Wechselwirkungen, Taktiken

4.2 Lernziele

4.2.1 Was sollen die Teilnehmer können?

4.2.1.1 Qualitative Anforderungen untersuchen und behandeln

Bedeutung von qualitativen Anforderungen für die Architektur erklären

Begriff der Qualität (angelehnt an DIN/ISO 9126) und von Qualitätsmerkmalen erklären

Qualitätsmodelle (wie etwa DIN/ISO 9126) erklären

Eigenschaften wichtiger Qualitätsmerkmale verstehen:

- Grundlegende Herangehensweise zur Behandlung erklären
- Typische Taktiken zur Erreichung erklären und anwenden
- Zusammenhang und Wechselwirkungen von Qualitätsmerkmalen erläutern

4.2.1.2 Qualitätsanforderungen detaillieren und beschreiben

Arten von Szenarien erklären und abgrenzen:

- Use-Case-Szenarien
- Grenzszenarien
- Stressszenarien

Erhebungstechniken für Szenarien erklären und anwenden:

- Brainstorming
- Interview
- Ableitung aus NFR-Spezifikation

Priorisierung von Szenarien für die Bearbeitung und die Bewertung erklären und durchführen

Kategorisierung von Szenarien nach Umsetzungsmöglichkeiten verstehen und anwenden:

- Szenarien als Akzeptanzkriterium (für einige wenige Anforderungen)
- Szenarien als Rahmenbedingung (für viele Entscheidungen)
- Szenarien als separat umsetzbare Qualitätsanforderung

Detaillierte Beschreibung von Szenarien: Mögliche Teile zur Verfeinerung und Konkretisierung erklären

Unterbringung von Szenarien in Architekturdokumentation erklären

Nutzen und Erstellung eines Qualitätsbaums erklären, eigene Qualitätsbäume erstellen

4.2.2 Was sollen die Teilnehmer verstehen?

- Ohne qualitative Anforderungen ist eine Architektur nicht sinnvoll bewertbar
- Qualitätsanforderungen widersprechen sich teilweise und werden typischer Weise von unterschiedlichen Stakeholdern unterschiedlich priorisiert

- Widersprüchliche Qualitätsanforderungen erzwingen Kompromisse in der Architektur. Diese Kompromisse transparent zu machen und angemessen zu entscheiden ist eine zentrale Aufgabe von Architekturbewertungen
- Konkrete und detaillierte Benennung der treibenden Qualitätsanforderungen ist die Voraussetzung für gute Architekturarbeit und Architekturbewertung
- Szenarien detaillieren Qualitätsanforderungen
- Qualitätsbäume strukturieren Szenarien und können bei der Priorisierung von Qualitätsanforderungen helfen.

4.2.3 Was sollen die Teilnehmer kennen?

Taxonomie Qualitätsmerkmale nach ISO 9126 (ohne deren exakte Definition)

Strukturierung von Qualitätsanforderungen nach Thomas Gilb

„Szenariobaukästen“ des SEI der CMU

4.3 Referenzen

[Bass+2003]

[Clements+2002]

[Kazman 2005]

[Barbacci 2003]

[Starke 2008]

Tom Gilb Referenz

5 Vorgehen bei der Bewertung

Dauer: 90 Min	Übungszeit: 150 Min
---------------	---------------------

5.1 Begriffe und Konzepte

ATAM, Kompromisse, Risiken, Nicht-Risiken, Sensitive Punkte, Entscheidungen, Rahmenbedingungen, Phasen von ATAM, Stakeholderbeteiligung, szenariobasierte Durchsprache, Bewertungsworkshop, Discovery Review, Zusammenspiel mit der Entwicklung

5.2 Lernziele

In diesem Teil lernen die Teilnehmer den Kern des methodischen Vorgehens zur qualitativen Bewertung von Software-Architekturen kennen.

5.2.1 Was sollen die Teilnehmer können?

5.2.1.1 ATAM – Architecture Tradeoff Analysis Method

- Bewertungsworkshops nach ATAM geeignet vorbereiten
- Die Phasen von ATAM erklären
- Die Schritte der Kernphasen von ATAM (Phase 1 und 2) erklären und auf die Bedürfnisse des eigenen Projekts anpassen.

5.2.1.2 Input für Workshops

- Anforderungen an Architekturdokumentation für Bewertungsworkshops verstehen und benennen:
- Detaillierte Szenarien
- Gesammelte Rahmenbedingungen
- Nachvollziehbar dokumentierte Entscheidungen
- Sichten zur Unterstützung des Überblicks bzw. Detaildiskussionen
- Source Code Teile
- Messergebnisse und erhobene Metriken
- Anforderungen an die Darreichungsform und Toolunterstützung
- Szenarien als Architekturelevant erkennen und entsprechende Kriterien benennen (Querschnittlichkeit der nötigen Entscheidungen, schwere Abbildbarkeit in Bestehendes, teure Umsetzung, etc.)

5.2.1.3 Workshops durchführen

Qualitative Bewertung von Software-Architekturen nach ATAM durchführen

- Vorgehen bei qualitativer Bewertung erklären und durchführen
- Erstellung von Szenarien und Qualitätsbäumen erklären und durchführen
- Analyse von Software-Architekturen hinsichtlich Szenarien und Identifikation entsprechender Risiken selbständig durchführen

5.2.1.4 Bewertung und Vorgehen

- Szenariobasierte Bewertungsmethoden in gängige Vorgehensmodelle einbetten
- Iterative Entwicklung

- Agile Vorgehensweisen
- In all diesen Vorgehensmodellen: das Zusammenspiel mit der Umsetzung verstehen und erklären.

5.2.2 Was sollen die Teilnehmer verstehen?

- Architekturbewertung ist als Teil des Entwicklungsprozesses (wiederkehrend) effektiver als ein Qualitätscheck vorhandener Software (nach erfolgter Implementierung)
- Die breite Beteiligung verschiedener Stakeholder ist wichtig für den Erfolg von Bewertungsworkshops
- Bewertungsworkshops und die szenariobasierte Methodik verbessern die Kommunikation im Projekt und die Transparenz gegenüber Umsetzern und Kunden.
- CBAM ist eine ergänzende Methodik, die bei Kompromissen eingesetzt werden kann. Nutzenkurven helfen beim fällen schwieriger Entscheidungen.
- Quantitative Techniken können zur Unterstützung von szenariobasierten Bewertungen eingesetzt werden.
- Die Einführung von Bewertungsworkshops im eigenen Projekt oder Unternehmen muss geplant werden und schrittweise erfolgen.

5.2.3 Was sollen die Teilnehmer kennen?

Unterschiede in der Durchführung von Bewertungsworkshops mit unterschiedlichen Teilnehmergruppen (interne und externe Stakeholder)

Unterschiede verschiedener szenariobasierter Methoden bei der Bewertung selbst (SAAM, ATAM, CBAM, LAAAM)

5.3 Referenzen

[Clements+2002]

[Bass+2003]

[Kazman+ 2000]

[Nord+ 2003]

[Kazman 2005]

[Starke 2008]

6 Nacharbeit und Ergebnisverwertung

Dauer: 45 Min	Übungszeit: 45 Min
---------------	--------------------

6.1 Begriffe und Konzepte

Risikocluster, Minderungsmaßnahmen, Kommunikation der Ergebnisse, Entscheidungsplanung, last responisble moment

6.2 Lernziele

6.2.1 Was sollen die Teilnehmer können?

- Die Ergebnisse von Bewertungsworkshops erarbeiten, dokumentieren und erklären:
- Risiken
- Nicht-Risiken
- Kompromisse
- Sensitive Punkte
- Allgemeine ToDo's
- Ergebnisse von Bewertungsworkshops entsprechend verwerten und verfolgen
- Kommunikationsmittel für die Vermittlung von Ergebnissen kennen und nutzen

6.2.2 Was sollen die Teilnehmer verstehen?

Risiken müssen aktiv angegangen werden.

Noch offene Entscheidungen müssen geplant werden.

6.2.3 Was sollen die Teilnehmer kennen?

Methoden zur Risikoeinschätzung und –behandlung.

6.3 Referenzen

[Clements+2002]

[Bass+2003]

[Bass+ 2006]

[Starke 2008]

7 Bewertung bestehender System(-teile)

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

7.1 Begriffe und Konzepte

Metriken, Messungen, Mathematische Modelle, Umsetzungsprüfung, Ist-Architektur, Soll-Architektur, Pain Points, Werkzeuge, Rekonstruktion

7.2 Lernziele

7.2.1 Was sollen die Teilnehmer können?

7.2.1.1 Szenariobasierte Bewertung

Szenarienbasierte Bewertung für bestehende Systeme:

- Unterschiede zur Bewertung im Architekturentwicklungsprozess erklären
- Unterschiede zur Umsetzungsüberprüfung mit Metriken und Messungen erklären

Szenarien aus vorhandenen Problemen ableiten. Mit entsprechenden Fragen Stakeholder zur Definition wichtiger Szenarien leiten.

7.2.1.2 Umsetzungsprüfung und Metriken

Ziele der Umsetzungsprüfung auf Basis von Metriken, Messungen und Bewertungstools nennen.

Bewertung von Software-Architekturen im Hinblick auf ihre Umsetzung durchführen, d.h. die Frage beantworten, in wie weit eine Architektur auch im Code umgesetzt wurde. Auch: Bewertung von Software-Architekturen im Hinblick auf allgemeine Best-Practices oder Prinzipien durchführen:

- Quellcode, bekannte Fehler in Quellcode und Fehlercluster analysieren
- Unterstützende Tools und Metriken kennen und anwenden.
- Operationalisierung entsprechender Überprüfungen im Prozess integrieren (beispielsweise als Teil des Check-In-Vorgangs, im Build etc.)

Konkrete Metriken für die Überprüfung von Design Best-Practices erklären. Insbesondere Abstraktion, Instabilität, Distanz, Fan-In / Fan-Out, zyklomatische Komplexität.

Verletzungen der Architekturvorgaben in der Implementierung behandeln

7.2.2 Was sollen die Teilnehmer verstehen?

Soll-Architektur und Ist-Architektur sind im Kontext der Umsetzungsprüfung sinnvolle Begriffe um zwischen den Architekturvorgaben und der De-facto-Architektur zu unterscheiden.

Die möglichen unterstützenden Artefakte für die Architekturüberprüfung bestehender Systemteile sind vielfältig.

7.2.3 Was sollen die Teilnehmer kennen?

Werkzeuge für die Umsetzungsprüfung mit Konformitätschecks und zur Metrikberechnung und entsprechender Auswertung (Bauhaus, Hello2morrow, Eclipse Metrics Plugin, Sonar, etc.)

Techniken zur Rekonstruktion von Architekturentscheidungen und Rekonstruktion der Ist-Architektur

7.3 Referenzen

[DeMarco 1986]

[Martin 2000]

<http://www.sonarsource.org/>

<http://www.hello2morrow.com/>

8 Beispiele für die Bewertung von Software-Architekturen

Dauer: 90 Min	Übungszeit: keine
---------------	-------------------

Dieser Abschnitt ist nicht prüfungsrelevant.

8.1 Begriffe und Konzepte

Innerhalb jeder akkreditierten Schulung muss mindestens ein Beispiel einer Software-Architektur vorgestellt, besprochen und bewertet 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

Vorstellung, eventuell Erarbeitung und Bewertung mindestens eines Beispiels einer Software-Architektur.

8.2.1 Was sollen die Teilnehmer können?

n.z.

8.2.2 Was sollen die Teilnehmer verstehen?

Den Ablauf von Bewertungsworkshops und die Bewertungsmöglichkeiten die sich bei realitätsnahen Architekturen ergeben.

8.2.3 Was sollen die Teilnehmer kennen?

Einige Beispiele (möglichst) realitätsnaher Bewertungsinputs und Bewertungsergebnisse

- Szenarien
- dokumentierte Rahmenbedingungen
- dokumentierte Entscheidungen
- Architekturüberblick
- Pain Points und sensitive Punkte
- dokumentierte Kompromisse, Risiken und Nicht-Risiken

8.3 Referenzen

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

9 Quellen und Referenzen zu Software-Architekturbewertung

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.

[Barbacci 2003]

Barbacci, M.R., Ellison, R., et al.: Quality Attribute Workshops (QAWs), Third Edition. Software Engineering Institute, CMU/SEI-2003-TR-016

[Bass+2003]

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

[Bass+ 2006]

Bass, L., Nord, R., et al.: Risk Themes Discovered Through Architecture Evaluations. Software Engineering Institute, CMU/SEI-2006-TR-012

C

[Clements+2002]

Clements, P., R. Kazman, M. Klein: Evaluating Software Architectures – Methods and Case Studies. Addison Wesley, 2002.

[Clements+2003]

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

D

[DeMarco 1986]

DeMarco, T.: Controlling Software Projects: Management, Measurement and Estimation. Prentice Hall, 1986.

K

[Kazman 2005]

Kazman, R., Bass, L.: Categorizing Business Goals for Software Architectures. Software Engineering Institute, CMU/SEI-2005-TR-021

[Kazman+ 2000]

Kazman, R., Klein, M., Clements, P.: ATAM: Method for Architecture Evaluation. Software Engineering Institute, CMU/SEI-2000-TR-004

[Kruchten 1995]

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

M

[Martin 2000]

Martin, R.C.: Design Principles and Design Patterns. White-Paper, 2000.

N

[Nord+ 2003]

Nord, R.L., Barbacci, M.R., et al.: Integrating the Architecture Tradeoff Analysis Method (ATAM) with the Cost Benefit Analysis Method (CBAM). Software Engineering Institute, CMU/SEI-2003-TN-038

S

[Starke 2008]

Starke, G. (2008): Effektive Software-Architekturen - Ein praktischer Leitfaden. 3. aktualisierte und erweiterte Auflage 2008, Carl Hanser Verlag, München.