Curriculum für

Certified Professional for Software Architecture (CPSA)®
Advanced Level

Modul
ARCEVAL

Architekturbewertung

Version 2015.1-DE; 2. September 2020

Inhaltsverzeichnis

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.

This version of this document has been produced with comments (like this one) enabled. It is **NOT** intended for public distribution or publication, but primarily for internal iSAQB purposes.

Verzeichnis der Lernziele

Einleitung: Allgemeines zum iSAQB Advanced Level

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.

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

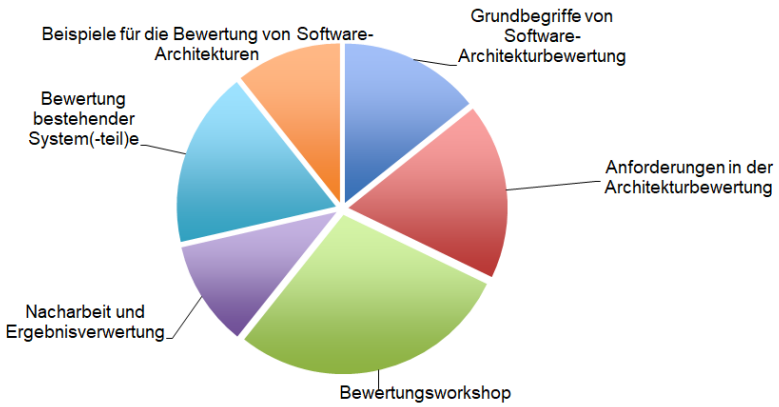
TBD

Hier bitte das Modul bzw. dessen Lerninhalte zusammenfassend in 5-8 Sätzen beschreiben. Dabei `Architekturbewertung` nicht entfernen, beim Zusammenbauen wird dieser Platzhalter mit dem Modulnamen ersetzt.

Struktur des Lehrplans und empfohlene zeitliche Aufteilung

Inhalt	Empfohlene Mindestdauer (min)
1. Grundbegriffe von Architekturbewertung	120
2. Anforderungen in der Architekturbewertung	150
3. Bewertungsworkshop	240
4. Nacharbeit und Ergebnisverwertung	90
5. Bewertung bestehender System(-teil)e	150
6. Beispiele für die Bewertung von Software-Architekturen	90
Summe	840 (14h)

Zeitliche Aufteilung der Themengebiete



Bitte sowohl die oben angegebene Tabelle als auch das beiliegende Excel-Dokument entsprechend anpassen und das Pie-Chart als "zeitaufteilung.png" nach `../images/01-basics` exportieren

= = =

Please adjust the table above as well as the excel document according to your curriculum and export the pie chart as "chronological_breakdown.png" to `../images/01-basics`.

Bitte die **?** durch die Anzahl der Tage sowie die erreichbaren Punkte ersetzen.

Dauer, Didaktik und weitere Details

Die unten genannten Zeiten sind Empfehlungen. Die Dauer einer Schulung zum Modul ARCEVAL 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 ARCEVAL 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

Teilnehmerinnen und 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-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 Software-Architekturen:
  - Auswahl geeigneter Dokumentationsstrukturen, Notationen, Ergebnistypen (Stakeholderorientierung)
  - Gemeinsame Arbeit an grundlegenden Konzepten der zu erstellenden Software
  - Orientierung von Software-Architektur an Anforderungen und Rahmenbedingungen
  - Messung und Evaluation der Qualität von Entwürfen und Entscheidungen

Kenntnisgruppen sowie Voraussetzungen bitte entsprechend ausformulieren!

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.

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.

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 Architekturbewertung

Dauer: 90 Min.	Übungszeit: 30 Min.

1.1. Begriffe und Konzepte

Softwareachitektur, Architekturbewertung, übergreifende Konzepte/Aspekte, Architekturziele, nichtfunktionale und funktionale Anforderungen an Systeme, Einflussfaktoren, Bewertungsmethoden.

Überschrift in 00-structure.adoc ersetzen

Sinnvolle Zeiten für Dauer und Übungszeit eintragen, vernünftige Begriffe aufzählen.

1.2. Lernziele

LZ 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.

LZ 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.

LZ 1-3: Voraussetzungen für Architekturbewertung

Organisatorische Voraussetzungen benennen und erklären:

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

LZ 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.

LZ 1-5: 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.

LZ 1-6: Was sollen die Teilnehmer kennen?

Wichtige Methoden zur Architekturbewertung (ohne genauen Ablauf):

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

Die einzelnen Lernziele müssen nicht als einfache Aufzählungen mit Unterpunkten aufgeführt werden, sondern können auch gerne in ganzen Sätzen formuliert werden, welche die einzelnen Punkte (sofern möglich) integrieren.

1.3. Referenzen

[Clements+2002], [Kazman+2000], [Bachmann+2000], [Bass+2003], [Clements+2003], [Kruchten 1995], [Starke 2011]

Eine Quelle wird über `[label]` referenziert. Dieses muss in `99-references/00-references.adoc` definiert sein.

= = =

A reference source is referenced via `[label]`. The label has to be defined in `99-references/00-references.adoc`.

2. Anforderungen in der Architekturbewertung

Dauer: 60 Min.	Übungszeit: 90 Min.

2.1. Begriffe und Konzepte

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

Überschrift in 00-structure.adoc ersetzen

Sinnvolle Zeiten für Dauer und Übungszeit eintragen, vernünftige Begriffe aufzählen.

2.2. Lernziele

LZ 2-1: 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.

LZ 2-2: 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.

LZ 2-3: 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.

LZ 2-4: 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.

Die einzelnen Lernziele müssen nicht als einfache Aufzählungen mit Unterpunkten aufgeführt werden, sondern können auch gerne in ganzen Sätzen formuliert werden, welche die einzelnen Punkte (sofern möglich) integrieren.

2.3. Referenzen

[Bass+2003], [Clements+2002], [Kazman+2005], [Barbacci+2003], [Starke 2011]

Eine Quelle wird über `[label]` referenziert. Dieses muss in `99-references/00-references.adoc` definiert sein.

= = =

A reference source is referenced via `[label]`. The label has to be defined in `99-references/00-references.adoc`.

3. Vorgehen bei der Bewertung

Dauer: 90 Min.	Übungszeit: 150 Min.

3.1. Begriffe und Konzepte

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

Überschrift in 00-structure.adoc ersetzen

Sinnvolle Zeiten für Dauer und Übungszeit eintragen, vernünftige Begriffe aufzählen.

3.2. Lernziele

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

LZ 3-1: 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.

LZ 3-2: 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 architekturrelevant erkennen und entsprechende Kriterien benennen (Querschnittlichkeit der nötigen Entscheidungen, schwere Abbildbarkeit in Bestehendes, teure Umsetzung, etc.)

LZ 3-3: 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.

LZ 3-4: 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.

LZ 3-5: 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 geplantwerden und schrittweise erfolgen.

LZ 3-5: 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).

Die einzelnen Lernziele müssen nicht als einfache Aufzählungen mit Unterpunkten aufgeführt werden, sondern können auch gerne in ganzen Sätzen formuliert werden, welche die einzelnen Punkte (sofern möglich) integrieren.

3.3. Referenzen

[Clements+2002], [Bass+2003], [Kazman+2000], [Nord+2003], [Kazman+2005], [Starke 2011]

Eine Quelle wird über `[label]` referenziert. Dieses muss in `99-references/00-references.adoc` definiert sein.

= = =

A reference source is referenced via `[label]`. The label has to be defined in `99-references/00-references.adoc`.

4. Nacharbeit und Ergebnisverwertung

Dauer: 45 Min.	Übungszeit: 45 Min.

4.1. Begriffe und Konzepte

Risikocluster, Minderungsmaßnahmen, Kommunikation der Ergebnisse, Entscheidungsplanung, Last Responisble Moment.

Überschrift in 00-structure.adoc ersetzen

Sinnvolle Zeiten für Dauer und Übungszeit eintragen, vernünftige Begriffe aufzählen.

4.2. Lernziele

LZ 4-1: Was sollen die Teilnehmer können?

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

LZ 4-2: Was sollen die Teilnehmer verstehen?

Risiken müssen aktiv angegangen werden. Noch offene Entscheidungen müssen geplant werden.

LZ 4-3: Was sollen die Teilnehmer kennen?

Methoden zur Risikoeinschätzung und -behandlung.

Die einzelnen Lernziele müssen nicht als einfache Aufzählungen mit Unterpunkten aufgeführt werden, sondern können auch gerne in ganzen Sätzen formuliert werden, welche die einzelnen Punkte (sofern möglich) integrieren.

4.3. Referenzen

[Clements+2002], [Bass+2003], [Bass+2006], [Starke 2011]

Eine Quelle wird über `[label]` referenziert. Dieses muss in `99-references/00-references.adoc` definiert sein.

= = =

A reference source is referenced via `[label]`. The label has to be defined in `99-references/00-references.adoc`.

5. Bewertung bestehender System(-teil)e

Dauer: 60 Min.	Übungszeit: 90 Min.

5.1. Begriffe und Konzepte

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

Überschrift in 00-structure.adoc ersetzen

Sinnvolle Zeiten für Dauer und Übungszeit eintragen, vernünftige Begriffe aufzählen.

5.2. Lernziele

LZ 5-1: 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.

LZ 5-2: 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

LZ 5-3: 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.

LZ 5-4: 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.

Die einzelnen Lernziele müssen nicht als einfache Aufzählungen mit Unterpunkten aufgeführt werden, sondern können auch gerne in ganzen Sätzen formuliert werden, welche die einzelnen Punkte (sofern möglich) integrieren.

5.3. Referenzen

[DeMarco+2006], [Martin 2000], [Sonarqube], [Hello2morrow]

Eine Quelle wird über `[label]` referenziert. Dieses muss in `99-references/00-references.adoc` definiert sein.

= = =

A reference source is referenced via `[label]`. The label has to be defined in `99-references/00-references.adoc`.

6. Beispiele

Dauer: 90 Min.	Übungszeit: keine

Dieser Abschnitt ist nicht prüfungsrelevant.

6.1. Begriffe und Konzepte

Innerhalb jeder lizenzierten Schulung muss mindestens ein Beispiel für ARCEVAL vorgestellt werden.

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

Sinnvolle Zeiten für Dauer und Übungszeit eintragen.

6.2. Lernziele

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

LZ 98-1: Was sollen die Teilnehmer verstehen?

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

LZ 98-2: 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

KURZE ERKLÄRUNG ZU DEN ZIELEN DIESER LERNEINHEIT

6.3. Referenzen

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

Eine Quelle wird über `[label]` referenziert. Dieses muss in `99-references/00-references.adoc` definiert sein.

= = =

A reference source is referenced via `[label]`. The label has to be defined in `99-references/00-references.adoc`.

Referenzen

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

Aufbau eines Eintrags-Ankers:

```
- [[[label,Text der erscheint]]]
```

ACHTUNG: Die Labels dürfen nur Buchstaben beinhalten, keine Zahlen oder Sonderzeichen

= = =

Structure of an anchor:

```
- [[[label,text that will be shown]]]
```

ATTENTION: labels have to be non-numeric.

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+2006] DeMarco, T.: Controlling Software Projects: Management, Measurement and Estimation. Prentice Hall, 1986.

H

- [Hello2morrow] Hello2Morrow, online: http://www.hello2morrow.com/

K

- [Kazman+2000] Kazman, R., Klein, M., Clements, P.: ATAM: Method for Architecture Evaluation. Software Engineering Institute, CMU/SEI-2000-TR-004.
- [Kazman+2005] Kazman, R., Bass, L.: Categorizing Business Goals for Software Architectures. Software Engineering Institute, CMU/SEI-2005-TR-021.
- [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

- [Sonarqube] Sonarqube, online: https://www.sonarqube.org/
- [Starke 2011] Starke, G. (2011): Effektive Software-Architekturen - Ein praktischer Leitfaden. 5. Auflage 2011, Carl Hanser Verlag, München.