

Curriculum für  
CPSA Certified Professional for  
Software Architecture®

– Advanced Level –

**Modul:  
AGILA**

**Agile Software Architektur**



Version 1.2 (Februar 2015)

**© (Copyright), International Software Architecture Qualification Board e. V.  
(iSAQB®) 2014**

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 [contact@isaqb.org](mailto:contact@isaqb.org) nachfragen. Sie müssten in diesem Fall einen Lizenzvertrag mit dem iSAQB e. V. schließen.
2. Sind Sie Trainer, Anbieter oder Trainingsorganisator, ist die Nutzung der Dokumente und/oder Lehrpläne nach Erwerb einer Nutzungslizenz möglich. Hierzu bitte unter [contact@isaqb.org](mailto:contact@isaqb.org) nachfragen. Lizenzverträge, die solche Nutzung regeln, sind vorhanden.
3. Fallen Sie weder unter die Kategorie 1. noch unter die Kategorie 2. fallen, aber dennoch die Dokumente und/oder Lehrpläne nutzen möchten, nehmen Sie bitte ebenfalls Kontakt unter [contact@isaqb.org](mailto:contact@isaqb.org) zum iSAQB® e. V. auf. Sie werden dort über die Möglichkeit des Erwerbs entsprechender Lizenzen im Rahmen der vorhandenen Lizenzverträge informiert und können die gewünschten Nutzungsgenehmigungen erhalten.

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

## Inhaltsverzeichnis

<b>0</b>	<b><u>EINLEITUNG.....</u></b>	<b>6</b>
0.1	WAS VERMITTELT DAS MODUL "AGILA"?	6
0.2	WAS VERMITTELT EIN ADVANCED-LEVEL-MODUL?	6
0.3	WAS KÖNNEN ABSOLVENTEN DES ADVANCED LEVEL (CPSA-A)?	6
0.4	VORAUSSETZUNGEN ZUR CPSA-ADVANCED-ZERTIFIZIERUNG	6
0.5	GLIEDERUNG DES LEHRPLANS UND EMPFOHLENE ZEITLICHE AUFTEILUNG	7
0.6	DAUER, DIDAKTIK UND WEITERE DETAILS	7
0.7	VORAUSSETZUNGEN FÜR DAS MODUL AGILA	8
0.8	AUFBAU DER LERNEINHEITEN AUS LERNZIELEN	8
<b>1</b>	<b><u>GRUNDLAGEN AGILER SOFTWARE-ARCHITEKTUR.....</u></b>	<b>9</b>
1.1	BEGRIFFE UND KONZEPTE	9
1.2	LERNZIELE	9
<b>2</b>	<b><u>AGILES ARCHITEKTURVORGEHEN .....</u></b>	<b>11</b>
2.1	BEGRIFFE UND KONZEPTE	11
2.2	LERNZIELE	11
<b>3</b>	<b><u>ARCHITEKTURANFORDERUNGEN IN AGILEN PROJEKTEN .....</u></b>	<b>12</b>
3.1	BEGRIFFE UND KONZEPTE	12
3.2	LERNZIELE	12
<b>4</b>	<b><u>ARCHITEKTUREN IM TEAM ENTWERFEN UND WEITERENTWICKELN .....</u></b>	<b>14</b>
4.1	BEGRIFFE UND KONZEPTE	14
4.2	LERNZIELE	14
<b>5</b>	<b><u>REFLEXION UND FEEDBACK ZU ARCHITEKTURARBEIT IM AGILEN KONTEXT.....</u></b>	<b>16</b>
5.1	BEGRIFFE UND KONZEPTE	16
<b>6</b>	<b><u>BEISPIELE FÜR AGILE ARCHITEKTURARBEIT .....</u></b>	<b>17</b>
<b>7</b>	<b><u>QUELLEN UND REFERENZEN ZU AGILA .....</u></b>	<b>18</b>

## VERZEICHNIS DER LERNZIELE

LZ 1-1: Die Bedeutung agiler Ideen für Architekturarbeit kennen und erläutern können .....	9
LZ 1-2: Aufgaben der Architekturentwicklung und deren Abwandlung im agilen Umfeld kennen .....	9
LZ 1-3: Architekturaufwand angemessen an Problemstellung und Projekt ausrichten können.....	10
LZ 1-4: Agile Werkzeuge für die Architekturarbeit kennen .....	10
LZ 1-5: Möglichkeiten Architektur als Querschnittsaspekt in agilen Organisationen zu verankern kennen.....	10
LZ 2-1: Architekturarbeit iterativ und agil gestalten können .....	11
LZ 2-2: Rollenmodelle für Architekten in agilen Projekten kennen.....	11
LZ 2-3: Möglichkeiten Stakeholder in Architekturarbeit zu involvieren kennen ..	11
LZ 3-1: Qualitätsanforderungen zielgruppengerecht formulieren können .....	12
LZ 3-2: Agile Konzepte für Architektur Anforderungen nutzen können.....	12
LZ 3-3: Iterative Ansätze zur stetigen Erhebung von Architektur Anforderungen anwenden können.....	12
LZ 3-4: Gemeinsame Anforderungspflege, -schätzung und -priorisierung effektiv gestalten können.....	13
LZ 3-5: Dringlichkeit als Treiber für Architekturarbeit kennen und erläutern können .....	13
LZ 4-1: Methoden zum Treffen von Entscheidungen in Gruppen anwenden können .....	14
LZ 4-2: Gruppen und Teams bei Entscheidungen begleiten können .....	14
LZ 4-3: Rahmenbedingungen für Teamentscheidungen schaffen können.....	14
LZ 4-4: Architekturkonzepte zur Förderung lokaler Entscheidungsmöglichkeiten kennen .....	15
LZ 4-5: Methoden für Just-In-Time-Architekturentscheidungen kennen .....	15
LZ 4-6: Möglichkeiten Architekturentscheidungen in agilen Projekten zu kommunizieren kennen .....	15
LZ 5-1: Techniken zur gemeinsamen Reflexion von Architekturentscheidungen kennen .....	16
LZ 5-2: Ursachen für Architekturprobleme gezielt finden können .....	16
LZ 5-3: Feedback-Möglichkeiten aus der Umsetzung kennen und Ergebnisse auf Architekturziele zurückführen können .....	16
LZ 6-1: Beispiele für Entscheidungsverfahren in agilen Projekten kennen und nachvollziehen .....	17
LZ 6-2: Beispiele für agil ausgeprägte Architektur Anforderungen kennen und nachvollziehen .....	17
LZ 6-3: Physische Ausprägungen agiler Kommunikationsansätze kennen.....	17
LZ 6-4: Verschiebung von Architekturentscheidungen nachvollziehen können .	17

LZ 6-5: Beispiele für agil organisierte Architekturgruppen kennen und nachvollziehen ..... 17

## 0 Einleitung

### 0.1 Was vermittelt das Modul "AGILA"?

Teilnehmer lernen hier, Softwaresysteme und -architekturen nach agilen Prinzipien zu entwerfen und weiterzuentwickeln. Dabei geht es einerseits darum agile Prinzipien und Ideen auf Architekturarbeit zu übertragen, andererseits darum, Architekturpraktiken sinnvoll in agiles Vorgehen zu verankern.

Wird Architektur in Projekten entwickelt, die auf selbstbestimmte Teams oder gemeinsam wahrgenommene Verantwortung setzen, sind neue Fähigkeiten von Entwicklern und Architekten gefragt. Diese umfassen sowohl technische, als auch methodische und kommunikative Aspekte. Sie werden hier theoretisch und praktisch behandelt.

### 0.2 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. Details im Web unter <http://isaqb.org>.

### 0.3 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 und Verbesserung nichtfunktionaler Anforderungen konzeptionieren, entwerfen und dokumentieren. Sie können Entwicklungsteams bei der Umsetzung dieser Maßnahmen begleiten.
- Architekturelevante Kommunikation in mittleren bis großen Entwicklungsteams steuern und durchführen.

### 0.4 Voraussetzungen zur CPSA-Advanced-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 allen drei unterschiedlichen Kompetenzbereichen (Details siehe iSAQB-Website).
- Erfolgreiche Bearbeitung der CPSA-A-Zertifizierungsprüfung.

### 0.5 Gliederung des Lehrplans und empfohlene zeitliche Aufteilung

Inhalt	Empfohlene Mindestdauer
Grundlagen	90min
Agiles Architekturvorgehen	150min
Architekturanforderungen in agilen Projekten	240min
Architekturen im Team entwerfen	300min
Reflexion und Feedback	180min
Beispiele für agile Architekturarbeit	120min
Gesamt (3 Tage à 360min)	1080 min = 18h

### 0.6 Dauer, Didaktik und weitere Details

Die genannten Zeiten für die einzelnen Teile des Lehrplans stellen lediglich Empfehlungen dar. Die Dauer einer Schulung zu AGILA sollte mindestens 3 Tage betragen, kann aber durchaus 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 zum AGILA-Modul tragen bezüglich der Zulassung zur abschließenden Advanced-Level-Zertifizierungsprüfung folgende Punkte (Credit Points) bei:

Methodische Kompetenz:	20 Punkte
Technische Kompetenz:	0 Punkte
Kommunikative Kompetenz:	10 Punkte

## **0.7 Voraussetzungen für das Modul AGILA**

Teilnehmer sollten folgende Kenntnisse und/oder Erfahrung mitbringen:

- Praktische Erfahrung in Entwurf und Entwicklung kleiner bis mittelgroßer Softwaresysteme.
- Praktische Erfahrung im Umgang mit Architekturentscheidungen – deren Erarbeitung, deren Dokumentation und Kommunikation.
- Erste praktische Erfahrung in agilen Software-Projekten.

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

- Kenntnisse rund um agile Vorgehensmodelle:
  - Das agile Manifest [AgileM+2001]
  - Scrum [Schwaber+2013]
  - Lean / Kanban [Anderson2010], [Kniberg2011].
- Kenntnisse und erste praktische Erfahrung in der Erarbeitung und Definition von Architekturanforderungen sowie dem Umgang mit unterschiedlichen Stakeholdern von Entwicklungsprojekten.
- Erste praktische Erfahrungen oder Kenntnisse im Bereich der automatisierten Integration und Auslieferung von Software:
  - Kenntnisse und erste Erfahrungen in automatisiertem Testen auf Unit und Integrationsebene
  - Grundkenntnisse der Laufzeitanalyse von Software, etwa Profiling, Tracing, Log-Analyse, Datenanalyse
  - Grundkenntnisse in automatisiertem Build, Integration und Auslieferung von Software.

## **0.8 Aufbau der Lerneinheiten aus Lernzielen**

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.

Die Kapitel des Lehrplans sind anhand von Lernzielen gegliedert.

Bei Bedarf enthalten die Lernziele Verweise auf weiterführende Literatur, Standards oder andere Quellen.



# 1 Grundlagen agiler Software-Architektur

Dauer: 90 min

## 1.1 Begriffe und Konzepte

- Software-Architektur allgemein
- Agiles Manifest: Werte und Prinzipien
- Agile Vorgehensmodelle inkl. Scrum und Lean
- Agiler Umgang mit Software-Architektur
- Software-Architektur in agilen Praktiken/Werkzeugen verankern

## 1.2 Lernziele

### **LZ 1-1: Die Bedeutung agiler Ideen für Architekturarbeit kennen und erläutern können**

- Definition von Architekturarbeit und Abgrenzung von Design
- Agiles Gedankengut in Beziehung zur Arbeit an Software-Architektur:
  - Agile Philosophie im agilen Manifest, bzw. Scrum Guide
  - Relevante Prinzipien des agilen Manifests
  - Konzepte und Säulen von Lean
- Agile Vorgehensmodelle und deren Auswirkungen auf klassische Konzepte des Architekturvorgehens

### **LZ 1-2: Aufgaben der Architekturentwicklung und deren Abwandlung im agilen Umfeld kennen**

- Klärung wie die folgenden, im CPSA-Foundation-Lehrplan definierten, Aufgaben der Architekturentwicklung im agilen Umfeld wahrgenommen werden:
  - Anforderungen und Randbedingungen klären, hinterfragen und bei Bedarf verfeinern.
  - Strukturentscheidungen hinsichtlich Systemzerlegung und Baueinstruktur treffen, dabei Abhängigkeiten und Schnittstellen zwischen den Bausteinen festlegen.
  - Übergreifende technische Konzepte entscheiden (beispielsweise Persistenz, Kommunikation, GUI) und bei Bedarf umsetzen.
  - Software-Architektur auf Basis von Sichten, Architekturmustern und technischen Konzepten kommunizieren und dokumentieren.
  - Umsetzung und Implementierung der Architektur begleiten, Rückmeldungen der beteiligten Stakeholder bei Bedarf in die Architektur einarbeiten, Konsistenz von Quellcode und Software-Architektur prüfen und sicherstellen.
  - Software-Architektur bewerten, insbesondere hinsichtlich Risiken bezüglich der geforderten Qualitätsmerkmale.

- Wenig Änderung an den Themen der Architekturarbeit, jedoch häufig Anpassung der Arbeitsweise bei der Bearbeitung von Architekturfragen (Aufgabenverteilung, Entscheidungsfindung, Ergebniskommunikation etc.).
- Mapping der wichtigsten Veränderungen auf die Inhalte dieses Lehrplans.

**LZ 1-3: Architekturaufwand angemessen an Problemstellung und Projekt ausrichten können**

- Spannungsfeld zwischen schneller Reaktionsfähigkeit und guter Planbarkeit bzw. Steuerbarkeit.
  - Im negativen Extrem: Spannungsfeld zwischen geringer Reaktionsfähigkeit (Starre) und schwerer Beherrschbarkeit (Chaos)
- Gerade genug Architekturarbeit und unterstützende Konzepte - Arbeitseinsatz auf relevante Probleme beschränken:
  - Fokussierung von Aufwand nach Architekturerelevanz
  - Fokussierung von Kommunikation
- Lean Waste in Architekturaufgaben

**LZ 1-4: Agile Werkzeuge für die Architekturarbeit kennen**

- Taskverwaltung und -priorisierung
- Agile Alternativen zu Modellierungstools
  - Kompromiss zwischen einfacher Erstellbarkeit von Diagrammen und nicht durchsuchbaren oder nicht verknüpften Modellelementen
- Stetige Codeprüfung mit Architektureigenschaften verknüpfen

**LZ 1-5: Möglichkeiten Architektur als Querschnittsaspekt in agilen Organisationen zu verankern kennen**

Architektur als Basis-Fähigkeit von Entwicklern etablieren. Beispielsweise durch:

- Offene, projektübergreifende Kommunikation von Best-Practices
- Communities-of-Practice zu Architektur aufbauen und lebendig halten
- Architektur-Katas als Mittel Architekturwissen zu verbreitern
- Architekturverantwortung kommunizieren und übertragen

## 2 Agiles Architekturvorgehen

Dauer: 150 min

### 2.1 Begriffe und Konzepte

- Iterativ, inkrementelle Architekturentwicklung: Risikogetriebene Architekturarbeit und Release-Planung
- Architekturarbeit mit Scrum und Kanban
- Agile Architekturrollen
- Stakeholder-Einbindung

### 2.2 Lernziele

#### **LZ 2-1: Architekturarbeit iterativ und agil gestalten können**

- Risikogetriebene Architekturarbeit: Architektonisches Risiko als bestimmender Faktor für den Einsatz von Architekturpraktiken
- Dringlichkeit von Architekturfragestellungen einschätzen und Entscheidungen verschieben
- Architekturarbeit in Scrum einordnen
- Iteration 0 und nötige Vorarbeit bestimmen
- Architekturarbeit bei der Release-Planung berücksichtigen
- Architekturarbeit mit Kanban steuern

#### **LZ 2-2: Rollenmodelle für Architekten in agilen Projekten kennen**

- Rollenmodelle für Architekten kontextabhängig auswählen. Beispielsweise:
  - Architecture Owner/unterstützender Architekt
  - Architekturagenten
  - Arbeit ohne benannten Architekten
- Verantwortlichkeiten von Scrum-Rollen
- Zusammenspiel der drei Scrum-Rollen (Product Owner, Scrum Master, Entwicklungsteam) zur Erledigung von Architekturaufgaben

#### **LZ 2-3: Möglichkeiten Stakeholder in Architekturarbeit zu involvieren kennen**

- Methodische Andockpunkte für die Zusammenarbeit schaffen
- Stakeholder motivieren zur Architektur beizutragen
- Architektur Aspekte und -auswirkungen für fachliche Ansprechpartner verständlich machen

## **3    Architekturanforderungen in agilen Projekten**

Dauer: 240 min

### **3.1    Begriffe und Konzepte**

- Zusammenarbeit mit Kunden und anderen Stakeholdern: Kommunikationsebenen, Architekturprobleme zielgruppenorientiert ausdrücken, Anforderungen und Wünsche in Architekturkonzepte übersetzen
- Qualitätsanforderungen in agilen Prozessen
- Technische Schulden
- Architektonische Risiken inkl. Erhebung
- Anforderungspflege, Grooming, Schätzung und Priorisierung von Architekturanforderungen
- Dringlichkeit von Architekturthemen und der letzte vernünftige Moment

### **3.2    Lernziele**

#### **LZ 3-1: Qualitätsanforderungen zielgruppengerecht formulieren können**

- Abstraktionsstufen von Qualitätsszenarien
- Architektonische Probleme und technische Kompromisse in qualitativen bzw. fachlichen Aussagen ausdrückbar machen

#### **LZ 3-2: Agile Konzepte für Architekturanforderungen nutzen können**

- Stories mit qualitativen Aspekten erweitern (Akzeptanzkriterien)
- Architekturanforderungen in Backlogs verankern
- Architekturaufgaben schneiden und verteilen

#### **LZ 3-3: Iterative Ansätze zur stetigen Erhebung von Architekturanforderungen anwenden können**

- Laufende Zusammenarbeit mit Kunden(-vertretern):
  - Nutzen für fachliche Ansprechpartner herausstellen
  - Techniken zur effektiven, regelmäßigen Einbindung fachlicher Ansprechpartner
- Technische Schulden als Architekturanforderungen nutzen
- Techniken um Risiken zu erheben: alleine und in Gruppen
- Tests und statische Analyse als Quelle für Architekturanforderungen nutzen

**LZ 3-4: Gemeinsame Anforderungspflege, -schätzung und -priorisierung effektiv gestalten können**

- Architekturanforderungen regelmäßig priorisieren, ggf. trennen und konkretisieren (z. B. innerhalb des Backlog-Grooming)
- Gemeinsame Schätzung von architektonischen Fragestellungen
- Priorisierungstechniken für Architekturanforderungen

**LZ 3-5: Dringlichkeit als Treiber für Architekturarbeit kennen und erläutern können**

- Späte Architekturentscheidungen und offene Optionen als Wert (Real Options Theorie)
- Ansätze, um dringend notwendige Architekturentscheidungen erkennen zu können
- Kommunikation mit dem Product Owner und anderen priorisierungsrelevanten Rollen

## **4 Architekturen im Team entwerfen und weiterentwickeln**

Dauer: 300 min

### **4.1 Begriffe und Konzepte**

- Gruppenentscheidungsverfahren: Konsensieren, Veto-Verfahren
- Moderation von Entscheidungsverfahren
- Konzeptionelle Integrität bei mehreren Entscheidern
- Vertikale Architekturstile (im Gegensatz zu systemweiter Schichtung)
- Just-In-Time-Architekturentscheidungen
- Architekturprinzipien
- Architekturwand

### **4.2 Lernziele**

#### **LZ 4-1: Methoden zum Treffen von Entscheidungen in Gruppen anwenden können**

- Methodische Grundlagen um (gleichberechtigte) Gruppenmitglieder effektiv zu Entscheidungen zu führen:
  - Wahl- und Veto-Verfahren
  - Konsensieren/Konsens
  - Aufteilung von Entscheidungsvorbereitung und Entscheidung
  - Dynamik der Delegation von Entscheidungskompetenz verstehen
  - Anwendung im Zusammenspiel mit agilen Vorgehen (Scrum)

#### **LZ 4-2: Gruppen und Teams bei Entscheidungen begleiten können**

Um in Gruppen effektiv entscheiden zu können, sind methodische und kommunikative Aspekte wichtig:

- Gruppen in Entscheidungsprozessen moderieren
- Mit Einwänden und Widerständen richtig umgehen
- Techniken, um breite Diskussionen auf ein Entscheidungsziel zu lenken

#### **LZ 4-3: Rahmenbedingungen für Teamentscheidungen schaffen können**

- Verantwortlichkeiten klären und effektiv übertragen
- Prinzipien als Mittel für konsistente Entscheidungen und hohe konzeptionelle Integrität einsetzen
- Technische Systemauslegung, um lokale Entscheidungen zu fördern

**LZ 4-4: Architekturkonzepte zur Förderung lokaler Entscheidungsmöglichkeiten kennen**

Die technische Architektur beeinflusst auch, wie sehr typische Architekturthemen Entwickler beeinflussen. Ist der technologische Stack etwa nicht zentral vorgegeben, sondern durch Teams bestimmt, sind Änderungen am Team-Stack weniger weitreichend, die Kommunikationslast zwischen den Teams sinkt und Entscheidungen sind einfacher auf Teams und Entwickler verteilbar.

- Aktuelle architektonische Konzepte zur Förderung lokaler Entscheidungsfähigkeit einordnen: Vertikale Architekturstile wie (Micro-)Services oder CQRS, Polyglotte Architekturansätze etc.
- Voraussetzungen für vertikale Architekturansätze (organisatorisch und technisch)
- Kompromisse bei der Flexibilisierung von Technologieentscheidungen (z. B. Richtung Entwickler Know-How, Betrieb oder Wartung)
- Verbindung zu den Prinzipien des agilen Manifests (selbstorganisierte Teams, schnellere Reaktionsfähigkeit etc.)
- Risiken der Team-lokalen Optimierung kennen und abwägen können (beispielsweise Redundanz, Verlust konzeptioneller Integrität, übertriebene Vielfalt)

**LZ 4-5: Methoden für Just-In-Time-Architekturentscheidungen kennen**

- Konzept des letzten vernünftigen Moments
- Technische Möglichkeiten, um architektonische Fragestellungen zu verschieben
- Architektonische Fragestellungen in Teilentscheidungen trennen
- Techniken um dringende und risikoreiche Entscheidungen zu bearbeiten (Set based Design)

**LZ 4-6: Möglichkeiten Architekturentscheidungen in agilen Projekten zu kommunizieren kennen**

- Stand-Ups und Feedbackrunden zu Entscheidungen
- Leichtgewichtige Architektur-Bewertungsworkshops (siehe auch Kapitel 5)
- Architekturwand (Informativer Arbeitsplatz)
- Architekturkonzepte mit Prinzipien in Teams kommunizieren
- Notwendigkeit expliziter Entscheidungskriterien für manche (große, riskante) Entscheidungen erkennen

## **5 Reflexion und Feedback zu Architekturarbeit im agilen Kontext**

Dauer: 180 min

### **5.1 Begriffe und Konzepte**

- Leichtgewichtige Architekturbewertung und andere Feedbackmechanismen
- Root-Cause-Analyse
- Testen qualitativer Systemeigenschaften
- Qualitätsindikatoren und Metriken

#### **LZ 5-1: Techniken zur gemeinsamen Reflexion von Architekturentscheidungen kennen**

- Architekturbewertungstechniken leichtgewichtig ausprägen
  - Moderation von kleinen Bewertungsworkshops
  - Richtiges Feedback zu Entwürfen und Entscheidungsvorlagen (Feedbackregeln)
- Metriken und Tools zur stetigen Architekturüberprüfung einsetzen
- Realitätscheck für Architekturziele

#### **LZ 5-2: Ursachen für Architekturprobleme gezielt finden können**

- Techniken für die Ursachenforschung in Gruppen (z. B. Root-Cause Analyse)
- Zusammenhang zwischen Test- bzw. Messergebnissen und Architekturentscheidungen herstellen

#### **LZ 5-3: Feedback-Möglichkeiten aus der Umsetzung kennen und Ergebnisse auf Architekturziele zurückführen können**

- Testen qualitativer Eigenschaften: Techniken, Werkzeuge und Einbindung in die Entwicklungsaktivitäten
- Statische Qualitätsindikatoren auswählen und nutzen
- Trendanalysen auf Qualitätsindikatoren durchführen und Erkenntnisse ableiten
- Vorteile von Continuous Integration und Continuous Deployment in diesem Zusammenhang



## 6 Beispiele für agile Architekturarbeit

Dauer: 120 min

Anmerkung: Beispiele für reale Ausprägungen agiler Architekturarbeit können durch Trainer und Schulungsanbieter individuell ausgesucht werden.

Aufgrund der bei realen IT-Systemen oftmals geltenden Geheimhaltungs- oder Vertraulichkeitsregelungen stellt der iSAQB e. V. Schulungsanbietern und Trainern frei, Beispiele abstrahiert oder auszugsweise in Schulungen zu zeigen.

### **LZ 6-1: Beispiele für Entscheidungsverfahren in agilen Projekten kennen und nachvollziehen**

Teilnehmer sollen an mindestens zwei Beispielen nachvollziehen können, wie agile Projekte unterschiedlicher Größe Architekturentscheidungen treffen und kommunizieren. Dazu sollten in Schulungen die relevanten Meetings, Rollen und Verantwortlichkeiten beschrieben werden, sowie die konkrete Einbettung in den agilen Prozess des jeweiligen Projekts deutlich werden.

### **LZ 6-2: Beispiele für agil ausgeprägte Architektur Anforderungen kennen und nachvollziehen**

- Teilnehmer sollen mindestens ein Beispiel für die Formulierung von Architektur Anforderungen und deren Einbettung in agile Anforderungsartefakte (wie Stories oder Backlogs) nachvollziehen können.
- Teilnehmer sollen Beispiele nachvollziehen, die gemeinsame Arbeit an Architektur Anforderungen in realen Projekten zeigen (Grooming, Planning etc.).

### **LZ 6-3: Physische Ausprägungen agiler Kommunikationsansätze kennen**

Teilnehmer sollen in Beispielen sehen und nachvollziehen können wie agile Projekte Architekturinformationen und -entscheidungen kommunizieren (Fotos von informativen Arbeitsplätzen, Layouts von Informationsradiatoren etc.).

### **LZ 6-4: Verschiebung von Architekturentscheidungen nachvollziehen können**

Teilnehmer sollen in einem realen Beispiel die Verschiebung von Architekturentscheidungen kennenlernen und nachvollziehen können.

### **LZ 6-5: Beispiele für agil organisierte Architekturgruppen kennen und nachvollziehen**

Teilnehmer sollen in mindestens zwei realen Beispielen die Organisation von Architekturcommunities oder -gilden kennenlernen und nachvollziehen können, sowie die Aktivitäten und Techniken kennenlernen die Unternehmen einsetzen um diese Gruppen zu etablieren und aktiv zu halten.

## **7 Quellen und Referenzen zu AGILA**

### **A**

[AgileM+2001] agilemanifesto.org – Das agile Manifest.  
<http://agilemanifesto.org/iso/de/principles.html>

[Anderson2010] David Anderson: "Kanban", Blue Hole Press, 2010, Sequim, WA

[Appelo2010] Jurgen Appelo: "Management 3.0: Leading Agile Developers, Developing Agile Leaders", Addison Wesley 2010

### **F**

[Fairbanks2010] George Fairbanks: "Just Enough Software Architecture: A Risk-Driven Approach", Marshall & Brainerd 2010

### **K**

[Kniberg2011] Henrik Kniberg: "Lean from the Trenches: Managing Large-Scale Projects with Kanban", Pragmatic Bookshelf 2011

### **L**

[Larman+2008] Craig Larman, Bas Vodde: "Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum", Addison Wesley 2008

### **R**

[Richardson+2005] Jared Richardson, William A. Gwaltney: "Ship It!: A Practical Guide to Successful Software Projects", Pragmatic Bookshelf 2005

### **S**

[Schwaber+2013] Ken Schwaber, Jeff Sutherland: "The Definitive Guide to Scrum: The Rules of the Game", <http://www.scrumguides.org>

### **T**

[Toth2015] Stefan Toth: „Vorgehensmuster für Softwarearchitektur: Kombinierbare Praktiken in Zeiten von Agile und Lean“, 2. Auflage, Hanser Verlag 2015

### **V**

[Vigenschow+2012] Uwe Vigenschow, Björn Schneider, Ines Meyrose: „Soft Skills für IT-Berater: Workshops durchführen, Kunden methodisch beraten und Veränderungen aktiv gestalten“, dpunkt.verlag 2012