

Curriculum für
CPSA Certified Professional for
Software Architecture®

– Advanced Level –

**Modul:
SWAM**

**Software-Architektur für
mobile Geräte**



Version 1.4 (Oktober 2015)

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

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 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 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 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

0	<u>EINLEITUNG.....</u>	7
0.1	WAS VERMITTELT DAS MODUL "SWAM"?	7
0.2	WAS VERMITTELT EIN ADVANCED-LEVEL-MODUL?	7
0.3	WAS KÖNNEN ABSOLVENTEN DES ADVANCED LEVEL (CPSA-A)?	7
0.4	VORAUSSETZUNGEN ZUR CPSA-ADVANCED-ZERTIFIZIERUNG	7
0.5	GLIEDERUNG DES LEHRPLANS UND EMPFOHLENE ZEITLICHE AUFTEILUNG	8
0.6	DAUER, DIDAKTIK UND WEITERE DETAILS	8
0.7	VORAUSSETZUNGEN FÜR DAS MODUL SWAM	9
0.8	AUFBAU DER LERNEINHEITEN AUS LERNZIELEN	9
1	<u>SPEZIELLE RAHMENBEDINGUNGEN VON SOFTWARE-ARCHITEKTUR FÜR MOBILE DEVICES</u>	10
1.1	BEGRIFFE UND KONZEPTE	10
1.2	LERNZIELE	10
1.3	REFERENZEN	11
2	<u>ANFORDERUNGEN UND METHODIK.....</u>	12
2.1	BEGRIFFE UND KONZEPTE	12
2.2	LERNZIELE	12
2.3	REFERENZEN	13
3	<u>ARCHITEKTUR UND UMSETZUNG</u>	14
3.1	BEGRIFFE UND KONZEPTE	14
3.2	LERNZIELE	14
3.3	REFERENZEN	15
4	<u>TEST UND DEPLOYMENT.....</u>	16
4.1	BEGRIFFE UND KONZEPTE	16
4.2	LERNZIELE	16
4.3	REFERENZEN	17
5	<u>WERKZEUGE.....</u>	18
5.1	BEGRIFFE UND KONZEPTE	18
5.2	REFERENZEN	18
6	<u>RECHTLICHE FRAGEN.....</u>	20

6.1	BEGRIFFE UND KONZEPTE.....	20
6.2	LERNZIELE.....	20
6.3	REFERENZEN	21
7	<u>BEISPIELE FÜR MOBILE ANWENDUNGEN.....</u>	22
7.1	LERNZIELE.....	22
7.2	REFERENZEN	22
8	<u>QUELLEN UND REFERENZEN ZU SWAM</u>	23

VERZEICHNIS DER LERNZIELE

LZ 1-1: Aus den speziellen Rahmenbedingungen Entscheidungen ableiten können 10

LZ 1-2: Besonderen Einfluss von Qualitätsmerkmalen auf die Architektur verstehen 10

LZ 1-3: Verstehen, dass mobile Architekturen besondere Probleme lösen müssen. 10

LZ 2-1: Spezielle Anforderungen für Mobile Anwendung aufnehmen und umsetzen können 12

LZ 2-2: Schwierigkeiten und Aufwand zur Umsetzung spezieller „mobiler“ Features und Eigenschaften einschätzen können 12

LZ 2-3: Verstehen, dass mobile Anwendungen höhere Anforderungen an das Interaktions-Design stellen als Desktop-Applikationen 12

LZ 2-4: Unterschiedliche Einsatzgebiete und Arten von Apps kennen 13

LZ 2-5: Verschiedene Libraries für ausgewählte Plattformen kennen..... 13

LZ 2-6: Verschiedene Vorgehensmodelle kennen und wissen, dass das eigene Vorgehen geeignet angepasst werden muss 13

LZ 3-1: Grundlegend die Architektur von mobilen Betriebssystemen beschreiben können 14

LZ 3-2: Die Architekturen der mobilen Anwendung und des Backends miteinander abstimmen können..... 14

LZ 3-3: Die verschiedenen Möglichkeiten der Netzwerkkommunikation verstehen und zielgerichtet einsetzen können..... 14

LZ 3-4: Vor- und Nachteile verschiedener Sprachen verstehen 15

LZ 3-5: Unterschied zwischen mobilem und nicht-mobilem Frontend verstehen 15

LZ 3-6: Typische Lösungen für Problembereiche mobiler Anwendungen kennen... 15

LZ 3-7: Möglichkeiten zum Versand von Push Notifications kennen..... 15

LZ 4-1: Teststrategie für mobile Anwendungen abstimmen können 16

LZ 4-2: Verschiedene Testmethoden und Tools kennen 16

LZ 4-3: Beispielhafte Testanbieter in der Cloud kennen 16

LZ 4-4: Update Plan für mobile Anwendungen entwickeln können..... 16

LZ 4-5: Verstehen, welche verschiedenen Arten der Installation der Anwendung es gibt..... 16

LZ 4-6: Verstehen, dass nicht alle Benutzer die selbe Variante der Anwendung benutzen müssen 16

LZ 4-7: Verstehen, wie verschiedene AppStores funktionieren und wie sie benutzt werden 16

LZ 4-8: Verstehen der Notwendigkeit und des Nutzens Anwendungen zu signieren 16

LZ 4-9: Verstehen, wie Feedback gesammelt werden kann..... 16

LZ 4-10: Tools um Statistiken zum Nutzerverhalten zu bekommen kennen..... 17

LZ 4-11: Sich der Tatsache bewusst sein, dass es auf mobilen Geräten eine sehr große Browservielfalt gibt..... 17

LZ 5-1: Werkzeuge aufgrund von Anforderungen auswählen können	18
LZ 5-2: Verstehen, wie Buildserver und Buildtools beim Bau mobiler Anwendungen unterstützen.....	18
LZ 5-3: Funktionsumfang von mehreren IDEs kennen.....	18
LZ 5-4: Verschiedene Profiler und deren Eigenschaften kennen	18
LZ 6-1: Technische Maßnahmen zur Umsetzung von Sicherheitskonzepten kennen und auswählen können	20
LZ 6-2: Einige häufige Sicherheitslücken kennen	20
LZ 6-3: Marken-, Warenzeichen- und Patentrecherche durchführen können.....	20
LZ 6-4: Für rechtliche Fragen kritische Stellen identifizieren können.....	20
LZ 6-5: Verstehen, dass es ist nicht möglich ist, alle Sicherheitslücken zu schließen	20
LZ 6-6: Verstehen, dass Datenschutz und Sicherheit Geld kosten und nicht vernachlässigt werden dürfen	20
LZ 6-7: Wichtige Fragestellungen des Datenschutzes kennen	20
LZ 6-8: Verstehen, dass Entwurfs- und Designentscheidungen juristische Konsequenzen haben können, die abgeschätzt werden müssen.....	21
LZ 7-1: Branchen und Einsatzgebiete von Apps nennen können	22
LZ 7-2: Ausprobieren verschiedenster Apps auf verschiedenen Devices.....	22

0 Einleitung

0.1 Was vermittelt das Modul "SWAM"?

Das Modul gibt einen umfassenden Überblick über die Themen mobiler Anwendungsentwicklung aus der Sicht des Architekten. Insbesondere werden die Unterschiede zu anderen technologischen Umgebungen betont. Mit Abschluss dieses Moduls ist der Architekt in der Lage, die für sein Projekt relevanten Fragen zu stellen und fachgerecht zu lösen.

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.

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

CPSA-A-Absolventen können:

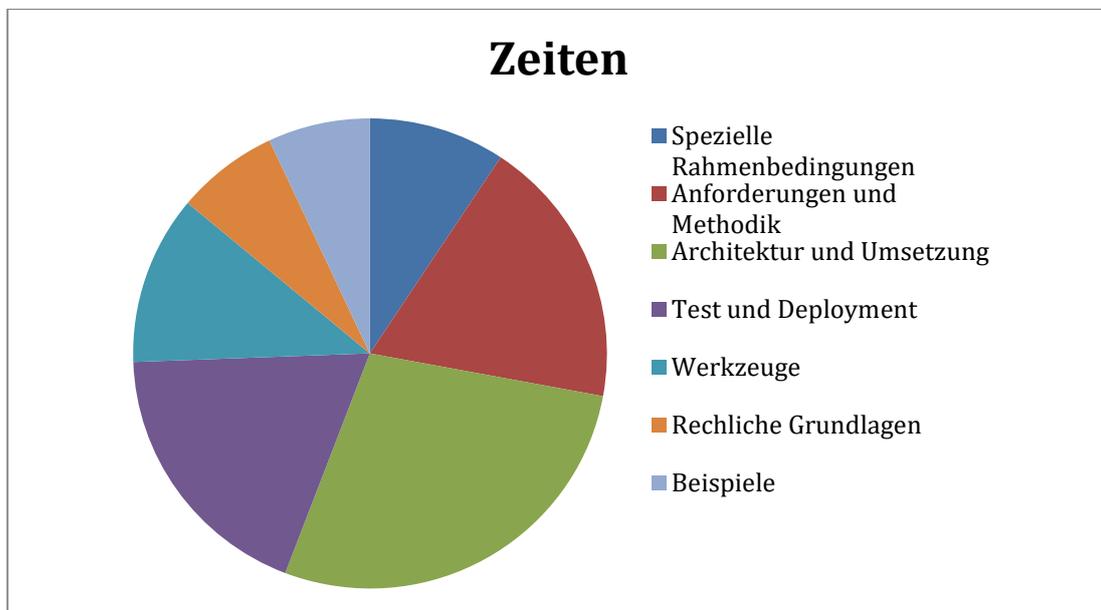
- Eigenständig und methodisch fundiert mittlere bis große IT-Systeme entwerfen.
- In IT-Systemen mittlerer bis hoher Kritikalität technische und inhaltliche Verantwortung übernehmen.
- Maßnahmen zur Erreichung nichtfunktionaler Anforderungen konzeptionieren, entwerfen und dokumentieren. Sie können Entwicklungsteams 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
Spezielle Rahmenbedingungen von Software-Architektur für mobile Devices	120min
Anforderungen und Methodik	240 min
Architektur und Umsetzung	360 min
Test und Deployment	240 min
Werkzeuge	150 min
Rechtliche Grundlagen	90 min
Beispiele für mobile Architekturen	90 min
Gesamt (3 Tage à 430 min)	1.290 min = 21,5h



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 SWAM 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 SWAM-Modul tragen bezüglich der Zulassung zur abschliessenden Advanced-Level-Zertifizierungsprüfung folgende Punkte (Credit Points) bei:

Methodische Kompetenz: 10 Punkte

Technische Kompetenz: 20 Punkte

Kommunikative Kompetenz: 0 Punkte

0.7 Voraussetzungen für das Modul SWAM

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
- Erfahrung im Umgang mit mobilen Devices
- Kenntnisse in der Erstellung von mobilen Anwendungen
- Erste Erfahrungen mit den speziellen Problemen bei der Erstellung von mobilen Anwendungen

Hilfreich für das Verständnis einiger Konzepte sind darüber hinaus:

- Vorgehen zum Erstellen einer Architekturdokumentation
- Grundlagen der Architekturbewertung
- Fähigkeiten zum Erstellen von Architektur und Design mit UML und Design- und Architektur-Patterns

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 **Spezielle Rahmenbedingungen von Software-Architektur für mobile Devices**

Dauer: 90 min, Übungszeit: ca. 30 Minuten

1.1 **Begriffe und Konzepte**

- Verschiedene Monetarisierungsmodelle
- User Experience, Haptik, Responsive Design
- Umgang mit begrenzten Ressourcen
- LifeCycle von Programmkomponenten der laufenden App
- Penetranz und Fragmentierung

1.2 **Lernziele**

LZ 1-1: Aus den speziellen Rahmenbedingungen Entscheidungen ableiten können

- Auswirkungen verschiedener Monetarisierungsmodelle berücksichtigen
- Wichtigkeit von User Experience, Responsive Design und Usability einbeziehen
- Einfluss von Ressourcenknappheit Rechnung tragen
- Frequenz und Art von Bewertung und Feedback durch den Anwender richtig bemessen
- Analysieren von Penetranz und Fragmentierung von Betriebssystemen und Hardwarevarianten um die Umsetzung zu fokussieren

LZ 1-2: Besonderen Einfluss von Qualitätsmerkmalen auf die Architektur verstehen

- Performance
- Stromverbrauch
- Speicherplatz
- Sicherheit
- User Experience

LZ 1-3: Verstehen, dass mobile Architekturen besondere Probleme lösen müssen

- verschiedene Betriebssysteme, Versionen, Hardware
- LifeCycle
- Schnittstellendesign in Bezug auf Performance und Kompatibilität
- Datensynchronisation von Daten mit dem Server
- Diversität von Datenquellen und Protokollen
- GUI Design und Haptik
- Mobiltelefon als persönlicher Begleiter

- Update-Verhalten

1.3 Referenzen

[Fling 2009]

2 Anforderungen und Methodik

Dauer: 90 min, Übungszeit: ca. 30 min

2.1 Begriffe und Konzepte

- Einfluss angepasster Vorgehensmodelle auf die Realisierung spezieller Qualitätsmerkmale, z. B.
 - Scrum
 - Mobile-D
 - Mobile/Contract first
 - Mobile only
- Spezielle Features von mobilen Anwendungen z. B.
 - Location Detection
 - Augmented Reality
 - Device-to-Device-Communication z. B. NFC, WiFi-direct, Bluetooth
- Werkzeuge, um Anforderungen zu beschreiben z. B.
 - Device-, Social-, User-Context
 - Application Sketches

2.2 Lernziele

Teilnehmer kennen die speziellen Anforderungen von mobilen Anwendungen und haben einen Überblick über Möglichkeiten, diese in den Architektur-Entwicklungsprozess einfließen zu lassen.

LZ 2-1: Spezielle Anforderungen für Mobile Anwendung aufnehmen und umsetzen können

- Erweiterten Systemkontext erstellen können
- Application Sketches erstellen können

LZ 2-2: Schwierigkeiten und Aufwand zur Umsetzung spezieller „mobiler“ Features und Eigenschaften einschätzen können

Zum Beispiel:

- Synchronisation und Offlinebetrieb
- Verarbeitung von Sensordaten
- Location Detection
- Augmented Reality

LZ 2-3: Verstehen, dass mobile Anwendungen höhere Anforderungen an das Interaktions-Design stellen als Desktop-Applikationen

- Mobile Anwendungen werden über Touch-Oberflächen und Gesten gesteuert
- Mobile Anwendungen müssen auf physisch kleinen Bildschirmen dargestellt werden

- Design-Guidelines von Plattform Providern machen Vorgaben für die Gestaltung der UI um sie über Anwendungen hinweg zu vereinheitlichen

LZ 2-4: Unterschiedliche Einsatzgebiete und Arten von Apps kennen

Zum Beispiel:

- Business Apps
- eLearning
- wissenschaftliche Apps
- Apps aus dem Bereich Medizin
- Spiele
- Lifestyle Apps
- Kommunikation und soziale Netzwerke
- Industrielle Anwendungen

LZ 2-5: Verschiedene Libraries für ausgewählte Plattformen kennen

Zum Beispiel Bibliotheken für:

- Kommunikation
- Datenbankzugriffe
- Dependency Injection

LZ 2-6: Verschiedene Vorgehensmodelle kennen und wissen, dass das eigene Vorgehen geeignet angepasst werden muss

2.3 Referenzen

[VanZandt 2008]

[Schwaber 2004]

[Sutherland, 2014]

[da Cunha et al 2011]

[Spataru 2010]

[VTT 2006]

[Schiller 2013]

[Lee 2013]

3 Architektur und Umsetzung

Dauer: 360 min, Übungszeit: ca. 60 min

3.1 Begriffe und Konzepte

- Grundlegendes Verständnis verschiedener Betriebssystem-Architekturen
- Lösungen für wiederkehrende Probleme
- Grundlagen zum GUI Design
- Auswahl der passenden Sprache
- Unterscheiden zwischen nativen und hybriden Anwendungen
- Verbindung von mobiler Anwendung und Servern

3.2 Lernziele

LZ 3-1: Grundlegend die Architektur von mobilen Betriebssystemen beschreiben können

- z. B.: iOS und Android
 - Grundsätzlicher Aufbau einer App
 - Features der jeweiligen Betriebssysteme
 - Threads
 - Datenbankzugriffen
 - „nativer“ Code
 - Dateisystemzugriffe
 - Dienste

LZ 3-2: Die Architekturen der mobilen Anwendung und des Backends miteinander abstimmen können.

- z. B. in Bezug auf:
 - Schnittstellen-Design
 - Skalierung
 - Updates
 - Kompatibilität auch bei Fehlern im Programm

LZ 3-3: Die verschiedenen Möglichkeiten der Netzwerkkommunikation verstehen und zielgerichtet einsetzen können

- synchrone vs. asynchrone Kommunikation
- pull/push-Anfragen
- Messaging z. B. JMS oder MQTT

- Protokolle wie http, REST, SOAP
- Websockets
- Kodierung z. B. binär, xml, json
- Medium z. B. WiFi, Bluetooth, 3G

LZ 3-4: Vor- und Nachteile verschiedener Sprachen verstehen

Zum Beispiel:

- Java vs. C++
- Objective-C vs. Swift
- HTML 5, CSS 3, Java Script
- native und hybride Entwicklung

LZ 3-5: Unterschied zwischen mobilem und nicht-mobilem Frontend verstehen

Eine mobile App an einen Server anzuschließen erfordert mehr Aufwand als nur die Präsentationsschicht auszutauschen, z. B. weil die Granularität der Schnittstelle zu viele Ressourcen benötigt, weil das Protokoll geändert werden muss, die Datenmengen anders abgestimmt werden muss (paging, lazy load) etc.

LZ 3-6: Typische Lösungen für Problembereiche mobiler Anwendungen kennen

- Lebenszyklus der Anwendung, Initialisierung und Zerstörung
- Navigation innerhalb der Anwendung und zwischen Anwendungen, Workflows
- Hintergrundprozesse und Nebenläufigkeit
- Kommunikation verschiedener Teile einer Anwendung miteinander
- Kommunikation verschiedener Anwendungen auf einem Gerät
- Einsatz von Multimedia
- Umgang mit Benachrichtigungen und Ausnahmesituationen
- Online/Offline-Betrieb und Ausfall von Netzwerkverbindungen

LZ 3-7: Möglichkeiten zum Versand von Push Notifications kennen

Zum Beispiel:

- WebSockets
- Google Cloud Messaging
- Apple Push Notification Service

3.3 Referenzen

[Apple 2014]

[Google]

4 Test und Deployment

Dauer: 180 min, Übungszeit: ca. 60 min

4.1 Begriffe und Konzepte

- Arten und Ebenen von Tests
- Testumgebungen
- Testtools und Testmethoden
- Continuous Integration (CI) , Continuous Delivery (CD)
- Deployment und Installation
- Feedback von Anwendern und der installierten Anwendung

4.2 Lernziele

LZ 4-1: Teststrategie für mobile Anwendungen abstimmen können

- Unit-Tests, Integrationstest, Akzeptanztests, Lasttests
- Jeweilige Testumgebung, z. B. eigene Hardware, Emulator, Cloud-Anbieter
- Besondere Eigenschaften der Testumgebungen einbeziehen, z. B. Latenz bei Cloud-Anbietern, Auswertungsarten, Kosten

LZ 4-2: Verschiedene Testmethoden und Tools kennen

- z. B. Mocks, GUI-Testtools, Testmonkeys, Instrumentation

LZ 4-3: Beispielhafte Testanbieter in der Cloud kennen

LZ 4-4: Update Plan für mobile Anwendungen entwickeln können

- Datenmigration auf dem Gerät
- Update-Zeitpunkt und -reihenfolge, z. B. geografisch

LZ 4-5: Verstehen, welche verschiedenen Arten der Installation der Anwendung es gibt

- Öffentliche und private AppStores, via Mail, Downloadlink

LZ 4-6: Verstehen, dass nicht alle Benutzer die selbe Variante der Anwendung benutzen müssen

- Beta-Programme, invited-only Programme
- Varianten (Free/Paid/Freemium)

LZ 4-7: Verstehen, wie verschiedene AppStores funktionieren und wie sie benutzt werden

- z. B. Apple AppStore und Google Play

LZ 4-8: Verstehen der Notwendigkeit und des Nutzens Anwendungen zu signieren

LZ 4-9: Verstehen, wie Feedback gesammelt werden kann

- Crashreports

- Anwenderfeedback
- Benutzungsstatistiken
- Analytics-Engine

LZ 4-10: Tools um Statistiken zum Nutzerverhalten zu bekommen kennen

LZ 4-11: Sich der Tatsache bewusst sein, dass es auf mobilen Geräten eine sehr große Browservielfalt gibt

4.3 Referenzen

[Hockeyapp]

[Testdroid]

[Monkey]

[Espresso]

[Selendroid]

5 Werkzeuge

Dauer: 100 min, Übungszeit: ca. 50 min

5.1 Begriffe und Konzepte

- IDEs
- Profiler
- Buildtools
- Buildserver

LZ 5-1: Werkzeuge aufgrund von Anforderungen auswählen können

- Kaum ein Werkzeug ist für jeden Anwendungsfall geeignet.
- Jedes Werkzeug bringt spezifische Vor- und Nachteile mit sich.

LZ 5-2: Verstehen, wie Buildserver und Buildtools beim Bau mobiler Anwendungen unterstützen

- z. B. Jenkins beim Umgang mit vielen Testumgebungen mit dem Matrix Project Plugin
- z. B. Gradle Skalierung: Skriptlänge, Diversität, Compilezeit

LZ 5-3: Funktionsumfang von mehreren IDEs kennen

Zum Beispiel:

- Android Studio
- xCode
- Eclipse ADT
- Titanium Studio

LZ 5-4: Verschiedene Profiler und deren Eigenschaften kennen

Zum Beispiel zum Messen von:

- Update-Raten
- GUI-Schachtelungstiefen
- Ressourcennutzung (Speicher, CPU, Netzwerk, Massenspeicher, Batterie)

5.2 Referenzen

[Eclipse]

[Xcode]

[Android Studio]

[Titanium]

[Maven]

[Gradle]

[Jenkins]

6 Rechtliche Fragen

Dauer: 90 min

6.1 Begriffe und Konzepte

- Security
- Datenschutz
- Signatur
- Patent-, Marken- und Warenzeichenschutz

6.2 Lernziele

App-Entwicklung findet in einem rechtlich teilweise geregelten, teilweise offenen Markt statt.

Internationales Recht sowie Patentrecht spielt eine bedeutende Rolle.

LZ 6-1: Technische Maßnahmen zur Umsetzung von Sicherheitskonzepten kennen und auswählen können

- Berechtigungen vergeben
- Sandboxen
- Apps für andere Apps öffnen

LZ 6-2: Einige häufige Sicherheitslücken kennen

Zum Beispiel:

- Unerwünschter Aufbau von AdHoc-Kommunikationsverbindungen z. B. über Bluetooth, NFC, WiFi direct, o. ä.

LZ 6-3: Marken-, Warenzeichen- und Patentrecherche durchführen können

LZ 6-4: Für rechtliche Fragen kritische Stellen identifizieren können

Zum Beispiel:

- Bezahlvorgänge
- Persönliche Daten, Anmeldeinformationen
- Impressum
- Von Dritten eingebundene Inhalte

LZ 6-5: Verstehen, dass es ist nicht möglich ist, alle Sicherheitslücken zu schließen

LZ 6-6: Verstehen, dass Datenschutz und Sicherheit Geld kosten und nicht vernachlässigt werden dürfen

LZ 6-7: Wichtige Fragestellungen des Datenschutzes kennen

Zum Beispiel:

- Schutz vertraulicher Daten

LZ 6-8: Verstehen, dass Entwurfs- und Designentscheidungen juristische Konsequenzen haben können, die abgeschätzt werden müssen

6.3 Referenzen

[Solmecke et al, 2013]

[Baumgartner et al, 2012]

[Datenschutzanforderungen für Apps]

7 Beispiele für mobile Anwendungen

Dauer: 90 min

Anmerkung: Beispiele für mobile Systeme können durch Trainer und Schulungsanbieter individuell ausgesucht werden.

Innerhalb jeder akkreditierten Schulung muss mindestens ein Beispiel für den Entwicklungsprozess zur Erstellung einer Mobilen App durchgespielt werden.

7.1 Lernziele

Vorstellung bekommen, wo Apps ihren möglichen Einsatz haben und an einem Beispiel die Erstellung aus architektonischer Sicht durchspielen

LZ 7-1: Branchen und Einsatzgebiete von Apps nennen können

LZ 7-2: Ausprobieren verschiedenster Apps auf verschiedenen Devices

7.2 Referenzen

[Apple Store]

[Google Play]

8 Quellen und Referenzen zu SWAM

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

A

[Android Studio]

<https://developer.apple.com/xcode/>

[Apple Store]

<http://store.apple.com/de>

[Artisan 2008]

Artisan: Model Driven Development. http://www.omg.org/news/meetings/workshops/Real-time_WS_Final_Presentations_2008/Tutorials/00-T5_VanZandt-Mraidha_Part1.pdf, 2008

[Apple 2014]

About the iOS Technologies:

<https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>, 2014

C

[da Cunha et al 2011]

da Cunha, T.F.V.; Dantas, V.L.L. ; Andrade, R.M.C:SLeSS: A Scrum and Lean Six Sigma Integration Approach for the Development of Software Customization for Mobile Phones. Software Engineering (SBES), 2011 25th Brazilian Symposium on, 2011

E

[Eclipse]

<https://eclipse.org/>

[Espresso]

<https://code.google.com/p/android-test-kit/wiki/Espresso>

F

[Fling 2009]

Fling, Brian: Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps. O'Reilley 2009

G

[Google Play]

<https://play.google.com/store>

[Google Android]

Google: Application Fundamentals: <http://developer.android.com/guide/components/fundamentals.html>

[Gradle]

<https://gradle.org/>

H

[Hockeyapp]
<http://hockeyapp.net/features/>

J
[Jenkins]
<http://jenkins-ci.org/>

L
[Lee 2013]
Lee, David: http://randomwire.com/?attachment_id=7682, 2013

M
[Maven]
<http://maven.apache.org/>
[Monkey]
<http://developer.android.com/tools/help/monkey.html>

U
[Baumgartner et al, 2012]
Ulrich Baumgartner, Konstantin Ewald , Apps und Recht, Verlag C. H. Beck 2012

S
[Schwaber 2004]
Schwaber, Ken: Agiles Projektmanagement mit Scrum. Microsoft Professional, 2004
[Schiller 2013]
Schiller, Nicholas: The Mobile App Design Process: A Tube Map Infographic.
<http://acla.ala.org/techconnect/?p=2882>, 2013
[Selendroid]
<http://selendroid.io/>
[Solmecke et al, 2013]
Solmecke, Christian; Taeger, Jürgen; Feldmann, Thorsten: Mobile Apps: Rechtsfragen und rechtliche Rahmenbedingungen. Berlin/Boston, Verlag de Gruyter 2013
[Spataru 2010]
Spataru, Andrei Cristian: Agile Development Methods for Mobile Applications.
<https://www.inf.ed.ac.uk/publications/thesis/online/IM100767.pdf> 2010
[Sutherland, 2014]
Jeff Sutherland: The Art of Doing thice the work in half the Time, Crown Business

T
[Testdroid]
<https://cloud.testdroid.com/>

V
[VanZandt 2008]
VanZandt, Lonnie: http://www.omg.org/news/meetings/workshops/Real-time_WS_Final_Presentations_2008/Tutorials/00-T5_VanZandt-Mraidha_Part1.pdf. Washington 2008
[VTT 2006]

VTT: <http://agile.vtt.fi/index.html>.2006

X

[Xcode]

<https://developer.apple.com/xcode/>