

Curriculum for  
CPSA Certified Professional for  
Software Architecture®

– Advanced Level –

**Module:  
SWAM**

**Software Architecture for  
Mobile Devices**



Version 1.4 (October 2015)

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

The curriculum may only be used subject to the following conditions:

1. You wish to obtain the CPSA Certified Professional for Software Architecture Advanced Level® certificate. For the purpose of obtaining the certificate, it shall be permitted to use these text documents and/or curricula by creating working copies for your own computer. If any other use of documents and/or curricula is intended, for instance for their dissemination to third parties, for advertising etc., please write to [contact@isaqb.org](mailto:contact@isaqb.org) to enquire whether this is permitted. A separate license agreement would then have to be entered into.
2. If you are a trainer, training provider or training organizer, it shall be possible for you to use the documents and/or curricula once you have obtained a usage license. Please address any enquiries to [contact@isaqb.org](mailto:contact@isaqb.org). License agreements with comprehensive provisions for all aspects exist.
3. If you fall neither into category 1 nor category 2, but would like to use these documents and/or curricula nonetheless, please also contact the iSAQB e. V. by writing to [contact@isaqb.org](mailto:contact@isaqb.org). You will then be informed about the possibility of acquiring relevant licenses through existing license agreements, allowing you to obtain your desired usage authorizations.

We stress that, as a matter of principle, this curriculum is protected by copyright. The International Software Architecture Qualification Board e. V. (iSAQB® e. V.) has exclusive entitlement to these copyrights. The abbreviation "e. V." is part of the iSAQB's official name and stands for "eingetragener Verein" (registered association), which describes its status as a legal person according to German law. For the purpose of simplicity, iSAQB e. V. shall hereafter be referred to as iSAQB without the use of said abbreviation.

Table of contents

<b>0</b>	<b><u>INTRODUCTION</u></b>	<b>7</b>
0.1	WHAT DOES THE “SWAM” MODULE IMPART?	7
0.2	WHAT DOES AN ADVANCED LEVEL MODULE INCLUDE?	7
0.3	WHICH SKILLS DO GRADUATES OF THE ADVANCED LEVEL (CPSA-A) HAVE?	7
0.4	PREREQUISITES FOR THE CPSA ADVANCED CERTIFICATION	7
0.5	CURRICULUM STRUCTURE AND RECOMMENDED TEMPORAL BREAKDOWN.	8
0.6	DURATION, DIDACTICS AND FURTHER DETAILS	8
0.7	PREREQUISITES FOR THE SWAM MODULE	8
0.8	STRUCTURE OF LEARNING UNITS FOLLOWS LEARNING GOALS	9
<b>1</b>	<b><u>SPECIFIC CONSTRAINTS OF SOFTWARE ARCHITECTURE FOR MOBILE DEVICES</u></b>	<b>10</b>
1.1	TERMS AND CONCEPTS	10
1.2	LEARNING GOALS	10
1.3	REFERENCES	10
<b>2</b>	<b><u>REQUIREMENTS AND METHODOLOGY</u></b>	<b>11</b>
2.1	TERMS AND CONCEPTS	11
2.2	LEARNING GOALS	11
2.3	REFERENCES	12
<b>3</b>	<b><u>ARCHITECTURE AND IMPLEMENTATION</u></b>	<b>13</b>
3.1	TERMS AND CONCEPTS	13
3.2	LEARNING GOALS	13
3.3	REFERENCES	14
<b>4</b>	<b><u>TEST AND DEPLOYMENT</u></b>	<b>15</b>
4.1	TERMS AND CONCEPTS	15
4.2	LEARNING GOALS	15
4.3	REFERENCES	16
<b>5</b>	<b><u>TOOLS</u></b>	<b>17</b>
5.1	TERMS AND CONCEPTS	17
5.2	REFERENCES	17
<b>6</b>	<b><u>LEGAL QUESTIONS</u></b>	<b>18</b>

6.1	TERMS AND CONCEPTS.....	18
6.2	LEARNING GOALS.....	18
6.3	REFERENCES .....	19
<b>7</b>	<b><u>EXAMPLES OF MOBILE APPLICATIONS .....</u></b>	<b>20</b>
7.1	LEARNING GOALS.....	20
7.2	REFERENCES .....	20
<b>8</b>	<b><u>SOURCES AND REFERENCES FOR SWAM .....</u></b>	<b>21</b>

**LIST OF LEARNING GOALS**

LG 1-1: Being able to derive design decisions from constraints. .... 10

LG 1-2: Understand special influence of quality attributes on the architecture..... 10

LG 1-3: Understand that mobile architectures must solve special problems..... 10

LG 2-1: Being able to capture and implement special requirements of mobile applications.  
..... 11

LG 2-2: Being able to estimate difficulties and efforts to implement special “mobile” features  
and properties. .... 11

LG 2-3: Understand that mobile applications place higher demands on interaction design  
than desktop applications..... 11

LG 2-4: Know different application area and types of mobile apps. .... 11

LG 2-5: Know various software libraries for selected platforms..... 12

LG 2-6: Know different development methods and being aware that your own approach  
must be adapted..... 12

LG 3-1: Being able to give a high-level description of the architecture of mobile operating  
systems..... 13

LG 3-2: Being able to harmonise the architecture of the mobile application and of the back-  
end..... 13

LG 3-3: Understand various alternatives of network communication and use them  
appropriately. .... 13

LG 3-4: Understand the advantages and disadvantages of various implementation languages.  
..... 13

LG 3-5: Understand the differences between mobile and desktop front-ends. .... 14

LG 3-6: Know typical solutions for problem areas of mobile applications. .... 14

LG 3-7: Know solutions to send push notifications to devices. .... 14

LG 4-1: Being able to coordinate testing strategies for mobile applications. .... 15

LG 4-2: Know different testing methods and tools..... 15

LG 4-3: Know examples of cloud testing providers..... 15

LG 4-4: Being able to develop a mobile application update plan..... 15

LG 4-5: Understand different ways of application distribution and installation. .... 15

LG 4-6: Understand that not all users have to run the same version of an application. .... 15

LG 4-7: Understand how different app stores operate and how to use them. .... 15

LG 4-8: Understand the necessity and the benefits of signing applications..... 15

LG 4-9: Understand how to collect feedback. .... 15

LG 4-10: Know tools to get statistics of user behaviour..... 16

LG 4-11: Be aware of the large variation of browsers on mobile devices..... 16

LG 5-1: Being able to choose appropriate tools for given requirements. .... 17

LG 5-2: Understand how build servers and build tools support the creation of mobile  
applications. .... 17

LG 5-3: Know the features of various IDEs..... 17

LG 5-4: Know different profilers and their characteristics. .... 17

LG 6-1: Know technical means to implement security concepts and being able to choose  
among options..... 18

LG 6-2: Know some common security risks..... 18

LG 6-3: Being able to do trademarks and patents research..... 18

LG 6-4: Being able to identify critical areas with legal impact. .... 18

LG 6-5: Understand that it is not possible to close all security loopholes. .... 18

LG 6-6: Understand that data protection and security cost money and must not be neglected.  
..... 18

LG 6-7: Know important issues of data protection. .... 18

LG 6-8: Understand that architecture and design decisions may have legal impact that has to  
be assessed..... 19

LG 7-1: Being able to name industries and fields of use for mobile applications. ....	20
LG 7-2: Try out various apps on different devices. ....	20

## 0 Introduction

### 0.1 What does the “SWAM” module impart?

The module provides a comprehensive overview of topics related to application development for mobile devices from an architect’s perspective. The module emphasises the differences to other technical environments. Upon completion of this module the architect is capable to ask questions relevant for the project and solve them in an adequate manner.

### 0.2 What does an Advanced Level module include?

- The iSAQB Advanced Level offers a modular approach to education in three different domains with flexible, customisable training routes. It takes individual tendencies and priorities into account.
- The certification is done in homework. The evaluation and the oral exam is done by experts named by the iSAQB.

### 0.3 Which skills do graduates of the Advanced Level (CPSA-A) have?

CPSA-A graduates are able to:

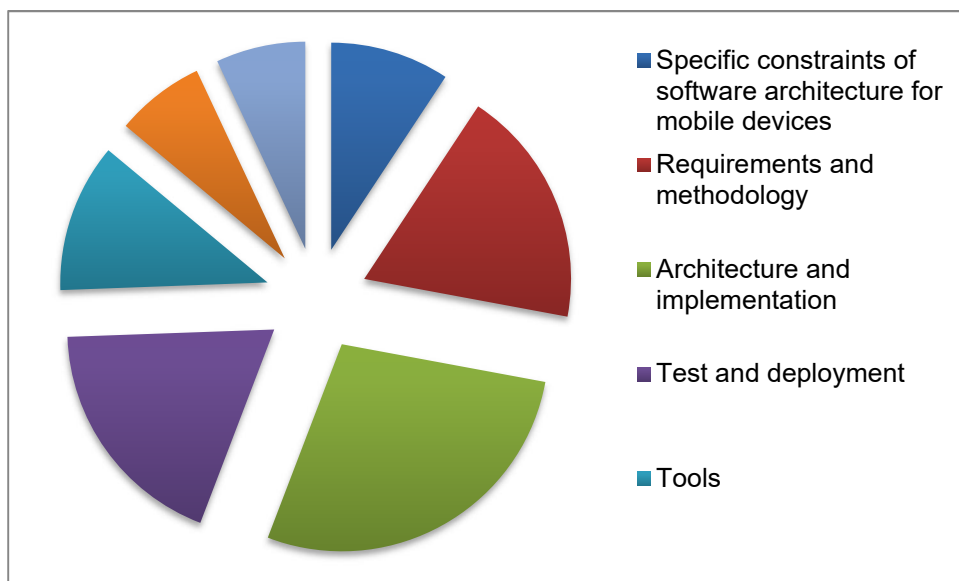
- Autonomously and methodically design medium to large size IT systems.
- Take technical and contentual responsibility for medium to highly critical IT systems.
- Identify, design and document actions to reach non-functional requirements. They are able to assist development teams with the implementation of these actions.
- Manage and perform the communication of system’s architecture with medium to large development teams.

### 0.4 Prerequisites for the CPSA Advanced Certification

- Successful education and certification as CPSA-F (Certified Professional for Software Architecture, Foundation Level).
- At least three years of full-time professional experience in the IT industry working on the design and implementation of at least two different IT-systems.
  - Exceptions allowed on request (for example: working on open-source projects)
- Completing iSAQB Advanced Level trainings with a minimum of 70 credit points from all three different areas of competence (for detail see the iSAQB web site).
- Successful completion of the CPSA-A exam.

## 0.5 Curriculum structure and recommended temporal breakdown.

Topic	Recommended minimum duration
Specific constraints of software architecture for mobile devices	120 min
Requirements and methodology	240 min
Architecture and implementation	360 min
Test and deployment	240 min
Tools	150 min
Legal basics	90 min
Examples of mobile architectures	90 min
Total (3 days at 430 min)	1,290 min = 21.5h



## 0.6 Duration, didactics and further details

The proposed durations for the parts of the curriculum are recommendations only. The duration of an SWAM training course should be at least three days, but may as well be longer. Providers may vary in their approach to duration, teaching methods, the type and structure of exercises as well as the detailed course outline. In particular, the type of examples and exercises are left completely open by the curriculum.

Licensed trainings of the SWAM module contribute the following points (credit points) to the permission for the final Advanced Level certification exam:

Methodical skills:	10 points
Technological skills:	20 points
Communicative skills:	0 points

## 0.7 Prerequisites for the SWAM module

Participants should have the following knowledge and/or experience:

- Foundations of the description of architectures using different views, cross-cutting concepts, design decisions, constraints, etc., as taught in the CPSA-F (Foundation Level).



- Practical experience with the handling of mobile devices.
- Knowledge about the creation of mobile applications.
- Basic experience with the specific problems of mobile application development.

Furthermore, the following will be useful for understanding certain concepts:

- Methodical approach to create an architecture documentation.
- Foundations of architecture evaluation.
- Being able to build an architecture and design using UML, design- and architectural patterns.

### **0.8 Structure of learning units follows learning goals**

The sections of the curriculum are described according to the following structure:

- Terms/concepts: essential core terms of the topic.
- Teaching/exercise time: Specifies the teaching and exercise time that at least has to be spent in this topic or its exercise in an accredited course.
- Learning goal: describes the contents to be taught including their key terms and concepts.

The chapters of the curriculum are structured according to learning goals.

If required the learning goals include references to further literature, standards, or other sources.



# 1 Specific Constraints of Software Architecture for Mobile Devices

Duration: 90 min, Exercises: ca. 30 min

## 1.1 Terms and concepts

- Different monetisation models
- User experience, haptics, and responsive design
- Handling of limited resources
- Life-cycle of program components in a running application
- Penetration and fragmentation

## 1.2 Learning goals

**LG 1-1: Being able to derive design decisions from constraints.**

- Consider the impact of different monetisation models.
- Consider the importance of user experience, responsive design, and usability.
- Deal with the effects of limited resources.
- Choose frequency and type of feedback/evaluation from application users.
- Analyse penetration and fragmentation of operating systems and hardware variants for focused implementation.

**LG 1-2: Understand special influence of quality attributes on the architecture.**

- Performance
- Energy consumption
- Memory requirements
- Security
- User experience

**LG 1-3: Understand that mobile architectures must solve special problems.**

- Various operating systems, versions, hardware platforms.
- Life-cycle.
- Interface design with respect to performance and compatibility.
- Data synchronisation with the server.
- Diversity of data sources and protocols.
- GUI design and haptics.
- Mobile phone as personal companion.
- Update behaviour.

## 1.3 References

- [Fling 2009]

## 2 Requirements and Methodology

Duration: 90 min, Exercise: ca. 30 min

### 2.1 Terms and concepts

- Influence of tailored development methods on achieving specific quality goals, e. g.,
  - Scrum
  - Mobile-D
  - Mobile/Contract first
  - Mobile only
- Features specific for mobile applications, e. g.,
  - Location Detection
  - Augmented Reality
  - Device-to-Device-Communication e. g., NFC, Wi-Fi-direct, Bluetooth
- Tools for requirements specification, e. g.,
  - Device-, Social-, User-Context
  - Application Sketches

### 2.2 Learning goals

Participants know specific requirements of mobile applications and have an overview of alternatives to include them in the architecture development process.

**LG 2-1: Being able to capture and implement special requirements of mobile applications.**

- Create the extended system context.
- Create application sketches.

**LG 2-2: Being able to estimate difficulties and efforts to implement special “mobile” features and properties.**

For example:

- Synchronisation and off-line processing.
- Processing of sensor data.
- Location detection.
- Augmented reality.

**LG 2-3: Understand that mobile applications place higher demands on interaction design than desktop applications.**

- Mobile applications are controlled with touch sensitive surfaces and gestures.
- Mobile applications have to be rendered on displays with small physical dimensions.
- Design guidelines of platform providers constrain the UI design to achieve common look and feel for all applications.

**LG 2-4: Know different application area and types of mobile apps.**

For example:

- Business apps

- eLearning
- Scientific apps
- Medical apps
- Games
- Lifestyle apps
- Communication and social networks
- Industrial apps

**LG 2-5: Know various software libraries for selected platforms.**

For example, libraries for:

- Communication
- Database access
- Dependency injection

**LG 2-6: Know different development methods and being aware that your own approach must be adapted.**

### **2.3 References**

[VanZandt 2008]  
[Schwaber 2004]  
[Sutherland, 2014]  
[da Cunha et al 2011]  
[Spataru 2010]  
[VTT 2006]  
[Schiller 2013]  
[Lee 2013]

## 3 Architecture and Implementation

Duration: 360 min, Exercise: ca. 60 min

### 3.1 Terms and concepts

- Basic understanding of different operating system architectures.
- Solutions for recurring problems.
- Basics of GUI design
- Choosing a suitable implementation language.
- Distinguish between native and hybrid applications.
- Connection between mobile application and server processes.

### 3.2 Learning goals

**LG 3-1: Being able to give a high-level description of the architecture of mobile operating systems.**

- For example, iOS and Android
- General structure of an app.
- Features of the respective OS:
  - Threads
  - Database access
  - “native” code
  - File system access
  - Services

**LG 3-2: Being able to harmonise the architecture of the mobile application and of the back-end.**

- For example, with respect to:
  - Interface design
  - Scaling
  - Updates
  - Compatibility even in case of software defects

**LG 3-3: Understand various alternatives of network communication and use them appropriately.**

- Synchronous vs. asynchronous communication
- Pull/push-requests
- Messaging e. g., JMS or MQTT
- Protocols like http, REST, SOAP
- Websockets
- encoding e.g. binary, xml, json
- Physical transport medium, e. g., Wi-Fi, Bluetooth, 3G

**LG 3-4: Understand the advantages and disadvantages of various implementation languages.**

For example:

- Java vs. C++
- Objective-C vs. Swift
- HTML 5, CSS 3, Java Script

- Native and hybrid development

**LG 3-5: Understand the differences between mobile and desktop front-ends.**

To connect a mobile application to a server requires more effort than just replacing the presentation layer. Possible reasons: the granularity of the interface requires too many resources, the communication protocol has to be changed, the amount of transferred data has to be re-organised (paging, lazy loading) etc.

**LG 3-6: Know typical solutions for problem areas of mobile applications.**

- Application life cycle, initialisation and destruction
- Navigation within and between different applications, workflows.
- Background processes and concurrency.
- Communication of different components within an application.
- Communication between applications on the same device.
- Use of multimedia.
- Handling of notifications and exceptions.
- On-line/off-line operation and network connection outage.

**LG 3-7: Know solutions to send push notifications to devices.**

For example:

- WebSockets
- Google Cloud Messaging
- Apple Push Notification Service

### **3.3 References**

[Apple 2014]

[Google]

## 4 Test and Deployment

Duration: 180 min, Exercise: ca. 60 min

### 4.1 Terms and concepts

- Types and levels of test
- Test environments
- Testing tools and testing methods
- Continuous Integration (CI), Continuous Delivery (CD)
- Deployment and installation
- Feedback from users and from installed applications

### 4.2 Learning goals

**LG 4-1: Being able to coordinate testing strategies for mobile applications.**

- Unit testing, integration testing, acceptance testing, load testing
- Testing environments, e. g., own hardware, emulators, cloud providers
- Consider specifics of the testing environments, e. g., latency of cloud based environments, assessment types, cost

**LG 4-2: Know different testing methods and tools.**

- For example, mocks, GUI-testing tools, test monkeys, instrumentation

**LG 4-3: Know examples of cloud testing providers.**

**LG 4-4: Being able to develop a mobile application update plan.**

- Data migration on the device.
- Update schedule, e. g., by geographic regions.

**LG 4-5: Understand different ways of application distribution and installation.**

- Public and private app stores, via e-mail, download links.

**LG 4-6: Understand that not all users have to run the same version of an application.**

- Beta-programs, invited-only programs.
- Variants (free/paid/freemium).

**LG 4-7: Understand how different app stores operate and how to use them.**

- For example, Apple AppStore and Google Play.

**LG 4-8: Understand the necessity and the benefits of signing applications.**

**LG 4-9: Understand how to collect feedback.**

- Crash reports

- User feedback
- Usage statistics
- Analytics engines

LG 4-10: Know tools to get statistics of user behaviour.

LG 4-11: Be aware of the large variation of browsers on mobile devices.

### **4.3 References**

[Hockeyapp]  
[Testdroid]  
[Monkey]  
[Espresso]  
[Selendroid]



## 5 Tools

Duration: 100 min, Exercises: ca. 50 min

### 5.1 Terms and concepts

- IDEs
- Profiler
- Build tools
- Build server

**LG 5-1: Being able to choose appropriate tools for given requirements.**

- Hardly any tool is suitable for all use cases.
- Every tool comes with its specific advantages and drawbacks.

**LG 5-2: Understand how build servers and build tools support the creation of mobile applications.**

- For example, Jenkins' "Matrix Project Plugin" supports multiple testing environments.
- For example, Scaling Gradle: script length, diversity, and compile times.

**LG 5-3: Know the features of various IDEs.**

For example:

- Android Studio
- Xcode
- Eclipse ADT
- Titanium Studio

**LG 5-4: Know different profilers and their characteristics.**

For example, to measure:

- Update rates
- GUI nesting levels
- Resource utilization (memory, CPU, network, mass storage, energy)

### 5.2 References

[Eclipse]  
[Xcode]  
[Android Studio]  
[Titanium]  
[Maven]  
[Gradle]  
[Jenkins]

## 6 Legal Questions

Duration: 90 min

### 6.1 Terms and concepts

- Security
- Data protection and privacy
- Signature
- Patent and trademark protection

### 6.2 Learning goals

App development takes place in a partly regulated, partly open market. International law and patent law have an important role.

**LG 6-1: Know technical means to implement security concepts and being able to choose among options.**

- Grant permissions
- Sandboxes
- Opening apps for other apps.

**LG 6-2: Know some common security risks.**

Example:

- Surreptitiously establishing ad hoc connections e.g. via Bluetooth, NFC, Wi-Fi direct, or the like

**LG 6-3: Being able to do trademarks and patents research.**

**LG 6-4: Being able to identify critical areas with legal impact.**

For example:

- Payment transactions
- Private data, login information
- Imprint
- Content included from third parties

**LG 6-5: Understand that it is not possible to close all security loopholes.**

**LG 6-6: Understand that data protection and security cost money and must not be neglected.**

**LG 6-7: Know important issues of data protection.**

For example:

- Protection of confidential data

**LG 6-8:** Understand that architecture and design decisions may have legal impact that has to be assessed.

### **6.3 References**

[Solmecke et al, 2013]

[Baumgartner et al, 2012]

## **7 Examples of Mobile Applications**

Duration: 90 min

Remark: choosing examples of mobile applications is at the trainer's or training provider's discretion.

Any accredited training must give at least one example of the development process of a mobile application.

### **7.1 Learning goals**

Gain insight in potential usage scenarios of mobile applications and run through an example of the creation of a mobile application from an architect's perspective.

**LG 7-1: Being able to name industries and fields of use for mobile applications.**

**LG 7-2: Try out various apps on different devices.**

### **7.2 References**

[Apple Store]

[Google Play]

## 8 Sources and References for SWAM

This section contains the sources that are referenced, whole or in part, in this curriculum.

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](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/iPhone-OSTechOverview/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'Reilly 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](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: Agile Project Management with Scrum. Microsoft Professional, 2004

[Schiller 2013]

Schiller, Nicholas: The Mobile App Design Process: A Tube Map Infographic.

<http://acrl.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 twice 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](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/>