MOOC TO GO

Jan Renz, Thomas Staubitz and Christoph Meinel
Hasso Plattner Institute, University of Potsdam, Potsdam, Germany

ABSTRACT

Massive Open Online Courses (MOOCs) have been one of the major trend topics of the last years within the e-learning community. Many companies, such as Coursera, edX and Udacity, launched MOOCs offering a broad range of topics. In this paper, the authors will take a look at the mobile support of different MOOC providers. Use cases and benefits of mobile access to a MOOC platform—both online and offline—will be shown. Finally, openSAP's solution to address this task will be demonstrated.
In this context, key technical decisions, which can serve as a blueprint for other MOOC providers will be discussed.

KEYWORDS

MOOC, Mobile Learning, Ubiquitous Learning, Responsive Web, Learning Apps

1. INTRODUCTION

Massive Open Online Courses (MOOCs) have been in the focus of the worldwide e-learning Community since at latest in early 2012, when Sebastian Thrun founded Udacity, based on the success of the experiences he made in 2011 at Stanford University offering a course to 160,000 students. Other players like Cousera and edX followed soon within the very same year. Soon, millions of users worldwide joined one or more of these free courses.

While much money and manpower was invested to build these platforms, the business models behind these new founded companies and organizations have been subject to frequent changes, as Schulmeister (2013) has been shown. In a type of a gold-rush mood the MOOC Providers battled to be first to market.

Despite the amount of scientific research on Mobile and Ubiquitous Learning when these companies launched their first courses, the impact of this research on the MOOC providers was minimal, as the authors will discuss in section two. While there have been MOOCs dealing with the topic of mobile learning, as described by de Waard (2011a), mobile technology as a part of the learning experience hasn't been in the focus of these MOOC providers.

The authors will discuss the demand and supply of mobile solutions for MOOCs in general in section two, while section three will focus on the current development of a mobile app at openSAP. Finally, section four and five will show limitations and problems, and highlight topics for future research.

2. MOOCS IN THE CONTEXT OF UBIQUITIOS LEARNING

2.1 Comparison of Mobile Device Support on Major MOOC Platforms

Following Hwang (2008) ubiquitous learning can be defined as "learning anywhere and anytime." Mobile learning therefore can be considered as a subpart of ubiquitous learning that includes mobile devices and wireless communication. In section two we will discuss the mobile support of selected MOOC platforms and take a look at the offline and slow bandwidth usage.

Mobile and Ubiquitous Learning (MUL) has been a trend topic long before MOOC became one of the hot topics in e-learning, as shown by Hwang and Tsai (2011). However, when the first MOOCs became available all platforms of the major MOOC providers lacked a proper support of MUL. As de Waard (2011b) states,

based on a survey of the participants of the MobiMOOC (a MOOC focusing on mobile learning and provided through a Wiki and an accompanying Google Group), 55% of the survey participants answered that they think that it is possible to follow a MOOC entirely through mobile devices.

Despite that fact, at the time of writing, none of the other major MOOC platforms offered a dedicated mobile app or website. However, research in the social media channels of Udacity, Coursera, and edX has shown that all of these platforms at least have unofficially confirmed that they have plans to offer better support for mobile devices.

It is not very astounding that My Open Courses [1], an Indian MOOC provider and official partner of the government programs NMEICT [2] and NPTEL [3], is a step ahead in delivering its contents via mobile devices as mobile technology has been leapfrogging the distribution of PCs and laptops in developing countries [4].

Table 1. Mobile Support of major MOOC providers, last updated at 11/25/2013

	edX	Coursera	Udacity	iversity
URL	edx.org	coursera.org	udacity.org	iversity.org
Responsive Website	Yes, min. 1024px width	No	Yes, min. 800px width	Yes
Own Mobile App	No	No	No	No
Mobile App planned	Yes	Yes	Yes	-
Third party solutions	iOS	iOS & Android	-	Android

The evaluation of the MOOC providers' mobile support in table 1 is based on statements in their social media channels, in-platform forums, and FAQ's. edX states on Facebook "[...] At this time, we do not have full support for tablet or mobile browsers. (We're working on this!) [...]" (edX Facebook (2013))

Questions on mobile support in Udacity's discussion forum are answered rather vaguely: "For the moment we don't support mobile platforms such as Android or the iPad. We are working on having support for them, however I don't have an ETA for when it will happen."(Udacity (2013)) An article about a new round of funding for Coursera in Forbes Online states, "Coursera has started building up a mobile devices team." (Forbes (2013a))

Third party apps are available for most of the platforms (see Table 1), demonstrating the users' demand for mobile solutions.

2.2 The Demand for Mobile Solutions for Ubiquitous Learning

On openSAP, an enterprise MOOC portal described in section three, a special type of a discussion forum[5] is presented to the users at the end of each course. It animates the users to articulate their praise and critique for the current course and their wishes for future courses. The following statement represents a wish that has been articulated by many users:

"I wish there would be stronger support for mobile devices: [...] Better touch friendly navigation for mobile devices. [...] I wish openSAP would re-consider a use case for mobile consumers: [...] I'd prefer a dedicated openSAP app to deliver a more usable, fluid user experience [...]"(openSAP Forum (2013))

The most common motivation for these wishes is the, currently missing, possibility to attend the courses offline on the commute. Another use case is demonstrated by the following discussion-post on openHPI: "Dear openHPI-Team, I'm currently in Spain and follow the lectures in a beach bar via a hotspot; I've got my iPad but no laptop with me [...]" (openHPI Forum (2013))

Often, participants cannot complete a course because they have left for holidays during the time of the course's final phase.

Identifying the need for MUL from analytics tools, such as Piwik [6], is difficult. These tools detect mobile devices based on the device information, which is retrieved from the requesting browser. This artificially

[1] http://myopencourses.com/
[2] National Mission on Education through ICT (http://www.nmeict.iitkgp.ernet.in/signalmain.htm)
[3] National Program on Technology Enhanced Learning (http://nptel.iitm.ac.in/)
[4] see e.g.: http://www.economist.com/node/10650775
[5] The I like, I wish format is a method, which originates from the Design Thinking process (http://dschool.stanford.edu/wp-content/themes/dschool/method- cards/i-like-i-wish-what-if.pdf)
[6] http://www.piwik.org

simplifies the complexity of the user scenarios to be expected in a MOOC learning environment. The used device is only one of the aspects that should be considered.

Table two shows that another important aspect is the availability of high speed Internet, providing enough bandwidth (on reasonable conditions).

Table 2. MOOC-Usage scenarios based on device and Internet connectivity

	Broadband	Slow Internet	No Internet
Mobile Device (Tablet)	Responsive Solution	Dedicated app to watch videos that have been downloaded earlier (At a point of time when the internet connection was faster, e.g. at home while connected via WLAN vs. on the commute while connected via UMTS). Textual content can be accessed. Quizzes and homework assignments can be submitted.	Dedicated app to watch videos that have been downloaded earlier. Quizzes and homework assignments cannot be submitted (Optionally, they could be stored on the mobile device and synchronized when a connection has been established again. Currently the authors are reluctant to add such a feature due to the complexity of the resulting issues).
Notebook or Desktop	Normal Website	Videos can only be watched if previously downloaded. Quizzes and homework assignments can be submitted. Participation in the discussion forum is possible.	Videos can only be watched if previously downloaded. No other form of participation is possible.

Research in openSAP's user tracking tool showed that, currently, mobile devices hold a share of 8 % of the visits and 5 % off all actions, such as page views. The iPad holds a share of 29.6 % of those visits. This number does not appear to be essential, but there are a couple of aspects that have to be considered:

• The number does not include those users who download the teaching videos on their desktop computer to watch them later on their mobile devices.

• Even on a web application that explicitly does not provide an optimized solution for mobile devices, one out of twelve visits comes from a mobile device (Higher numbers might be expected after providing a fully responsive version).

3. MOBILE SUPPORT AT OPENSAP

SAP, one of the world's largest software companies (see Forbes (2013b)), was one of the first global enterprises that launched its own MOOC portal: openSAP[7] offers free MOOCs to an interested audience. openSAP is a partner platform of openHPI (see Meinel & Willems (2013)), which delivers MOOCs since September 2012. The Hasso-Plattner-Institute at the University of Potsdam, Germany, provides openSAP with scientific guidance.

openSAP started in June 2013 with the course "Introduction to Software Development on SAP HANA." In November 2013 (the time of writing), 77,679 registered users enrolled 134,648 times in a total of 5 courses. 41,411 of these enrolled users can be considered as active participants (See Willems *et al.* (2013)) for the definition of active users at openHPI and openSAP). Two of these courses had been concluded at the time of writing. In these courses, a total of 14,810 Records of Achievement (RoA) and 17,817 Confirmations of Participation (CoP) have been issued to the course participants.

Courses on openSAP follow a traditional MOOC concept, so they have a defined start and end date and are structured into weeks. Each week consists of short video lectures, followed by diagnostic e-assessments (self-tests).

Currently, the openSAP website provides only limited support for tablet devices and is not optimized to be used on smartphones. To better support mobile devices—both online and offline—openSAP decided to create a native app, which is shown in Fig. 1. Currently, only iPads—running iOS 6 or higher—are supported. Smartphones and tablets running a different operating system are not supported by now. Table

[7] https://open.sap.com/

three shows a list of all major functionalities and their availability in respect to the available Internet connectivity, which are covered by the first release of the openSAP app. It is implemented as a hybrid app as described in Willnecker *et al.* (2012).

Thus, it features a mix of native views, which access the openSAP platform's data through a provided API, and web views, which load the platform's HTML pages into an embedded browser component.

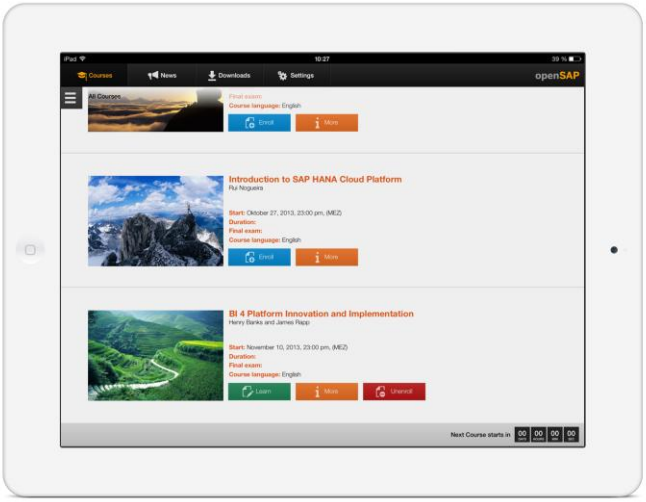

Figure 1. The app with the native course list for

Table 3. Offline availability of functionalities in the openSAP app

Function	Offline	Online
Register on platform	No	Yes
Enroll to course	No	Yes
Browse courses	Yes	Yes
Watch a video	Yes, if downloaded	Yes
Take a quiz	No	Yes

3.1 Optimizing API-Requests

The API was developed following the principles of the REST architecture as described in Masse (2011). The API Design is coherent to the usage of entities within the mobile app and does not rely on implementation details within the current platform.

The app can access the following content through the API:

- Courses, sections and items (items include videos, content pages and quiz pages)
- Announcements (such as global and course specific news)
- Enrollments (the courses of a specific user)
- Progress (the learners progress and visited items)

Following a strict REST design approach may lead to complex sequences of API calls for, otherwise, rather simple use cases. For example, the workflow to get a user's course progress would look like:

- Get the sections of a course
- (For every section) get all the items
- (For every item) get the progress of that item

This would result in many HTTP-requests from the app to the API-server, a large HTTP and data overhead, and therefore, a significant slower app performance.

To solve this problem, two solutions have been discussed. One possibility is to keep the API design strict to the REST paradigm and introduce a middleware layer, to cache requests and optimize access for the app.

As this solution results in higher maintenance efforts, the REST paradigm was softened where it appeared to be reasonable. It was decided to provide API Calls that enable access to the data in a more convenient way. In the given example this results in a single API call. Thereby, loading times within the affected parts of the app have been reduced from a few seconds to parts of a second.

3.2 Optimized Views

To improve the maintainability and reduce the required amount of effort for the implementation of the app, so-called web-views were used whenever it appeared to be reasonable. These web-views are loading certain HTML-pages from the platform into an embedded browser view component. Some parts of the loaded HTML pages have to be adjusted for the usage within the mobile app to provide an optimal user experience. For example, the header of the course page needs to be hidden, as the mobile app provides a header of its own. Furthermore, certain buttons to trigger functionalities—which either are not offered or handled differently in the context of the app—had to be hidden as well.

A first approach to handle this problem was to process these adjustments on the mobile device. The app code downloads the HTML, parses it, and modifies it as needed. This approach requires the app to be aware of the structure of the downloaded HTML.

Let us examine the following example. Assumed we have the following header:

```
<div id="header" class="no-print␣no-user">
  <img id="topbg" src="/plugins/.../bg.png" class="w100" height="106" alt="">[...]
  <a href="/login">Login</a></li> </div> [...]
 </div>
```

The app could either remove the HTML node with the id 'header' or inject custom CSS to hide that node. This approach introduces a severe maintainability issue, however. Assumed the website receives a facelift or gets updated to use HTML5 as in the listing below:

```
<header><img id="topbg" src="...bg.png" class="w100" height="106" alt="">
<div id="header-inner" class="main">[...] <a href="/login">Login</a></li>[...] </div>
</header>
```

This update of the website would break this feature of the mobile app (until an update of the app), as now the header cannot be detected and hidden or removed anymore.

Another approach that has been discussed was to add classes that serve this specific purpose to the page's HTML code. The app could now inject CSS to target these classes.

```
<div id="header" class="no-print␣no-user␣hide-me-in-ipad-app">
 [...]
</div>
```

This approach seems more feasible in terms of maintainability; still, yet another scenario needs to be reflected. The possibility that future development requires additional HTML elements, which will only be delivered to views that are requested from within the context of the mobile app, is reasonably high. Also, different versions of the app might require different versions of HTML to be delivered. Therefore, it was decided that the app should send a customized HTTP header with each HTTP request.

```
Example : User-Platform : OS APPIDENTIFIER APPVERSION
          User-Platform : iOS open.sap.com 1.0
```

Figure 2. The app with the native course list for

This header will be recognized by the platform, which in return will deliver the customized views to be displayed in the mobile app's web-views. Thus, the bigger part of the maintenance has been moved from the mobile client to the server, where it can be handled much easier and does not require the user to download a new version of the mobile app whenever the code on the platform has been updated (see section 3.3).

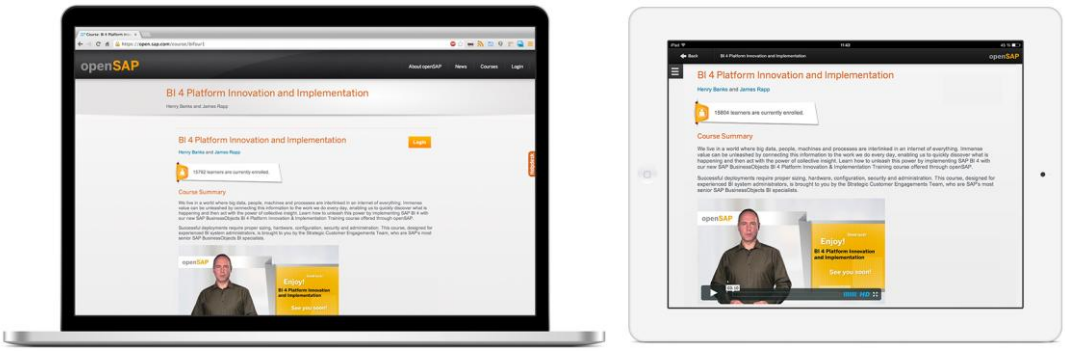

Figure 3. The course view in the platform and within the mobile app

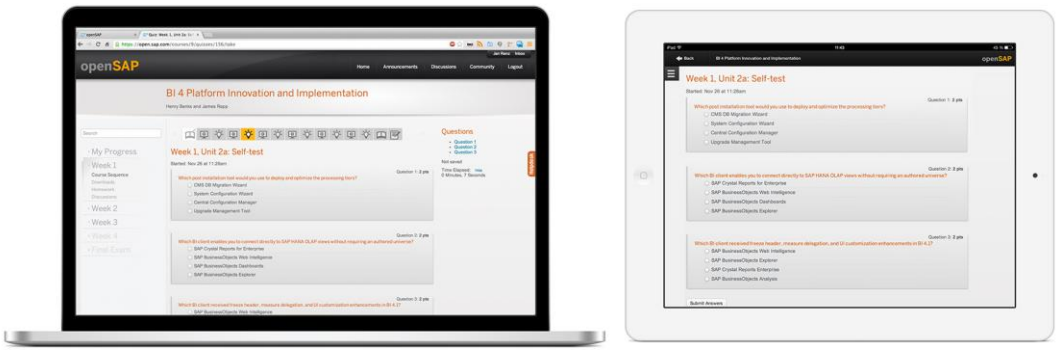

Figure 4. The quiz view in the platform and within the mobile app

As shown in Fig 3 and in Fig 4 an adapted web-view might omit some elements of the original web page. In this specific example the header(s), the left-hand navigation, and the login and help-desk buttons have been removed from the web-view, as these functionalities are natively provided by the mobile app.

3.3 Update and Release Cycles of the Mobile App

In web development, there is a trend towards continuous integration and rapid deployment cycles as described in Fowler & Foemmel (2006). After an update of the web platform all users will automatically access the newest version. In comparison to the easy rollout of a new version of the openSAP platform, updating the openSAP app is a more complex and time-consuming procedure. After creating and testing a new build of the app, it must be submitted to Apple. Only after a successful review by Apple it will be distributed to the user through Apple's App Store [8]. Even if a new version of the app is finally rolled out to the app store, there is no guarantee that all users will update to the new version. Especially those who disabled the automatic update function within iOS might stick with older versions of the app. This will lead to fragmentation and higher complexity in maintenance and user support.

3.4 Public API Specs to Encourage Third party Apps

Given the target audience of openSAP, which includes developers and other people interested in SAP's technologies, it can be assumed that many of the course participants have a high level of technical knowledge. By publishing the API documentation to the public, everyone would have the chance to use the API to send and retrieve data from openSAP within her own tool or environment.

This would enable users and third party developers to create their own native apps, as an alternative to existing ones or for currently unsupported platforms. Apps could be tailored to their own needs (e.g. an XBMC 8 Plugin showing all videos from the courses in which the user is enrolled). Shifting app development

[8] https://itunes.apple.com/de/genre/ios/id36

from the MOOC creators to 3rd party developers may lead to a faster release of apps, as limited resources of MOOC creators can be bypassed. As openSAP decides, which data will be published, and this decision by design reflects the user's access level, security and privacy issues are no concern.

There is a serious risk for the brand image, however. Badly designed, broken, or unmaintained 3rd party apps could reflect negatively on open Sap's public perception and image. Therefore, a strategy to deal with a possibly unsatisfying user experience of 3rd party apps is a core requirement before steps in this direction can be taken.

4. CONCLUSION

As shown in this paper, the support for MUL within the MOOC context needs to be improved. Although the users request native apps, they are not highest priority for MOOC providers today. MOOC platforms are new, and many other tasks with higher priority seem to be on the MOOC providers' agenda. Thus, currently, there are only very rare offerings of mobile apps by MOOC providers as they leave 3rd party providers to fill the gap. Owed to the tight time-boxing of MOOCs with fixed start and end dates, weekly deadlines, and the important role of the social context (see Grünewald *et al.* (2013)), users demand access to the platform on a regular basis, no matter where they are or what type of devices they can access. Mobile apps, such as the openSAP app, can provide the users with this possibility and, therefore, provide a ubiquitous learning experience.

Native apps are not necessarily required anymore to access MOOCs that adhere to the principles of responsive design. However, they can offer functionalities that cannot be provided within the browser environments today (for example downloading videos to watch them later offline at some point in the future). The openSAP app shows how the advantages of reusing components such as responsive web-views can be combined with native functionalities to find a balance between maintainability, costs, and user experience. Only native apps (including hybrid ones) allow a seamless usage in situations where a sufficient broadband connection may not be available (e.g. triggered by a change of the user's location).

To provide a ubiquitous user experience and to satisfy the users' requests, it is required to fill the offline gap in a mobile learners MOOC usage. The openSAP app addresses important parts of this gap (primarily the offline access to the videos). However, not all use-cases of taking a MOOC can be offered on mobile devices and even without a broadband connection (see table three). This may be resulting from technical issues (lab environment cannot be started on tablets hardware), the required amount of work to port server side components to the client (like quiz environments), or components that cannot be cached and provided offline (like communication and discussion tools). To address these issues, further development and research is required to bring MOOCs and MUL even closer together.

5. FUTURE RESEARCH

5.1 Ubiquitous Learning Based on HTML5 Features

The current approach of a hybrid app has several disadvantages. One of the biggest is the support of only a single platform. Minimizing the native part and shifting functionality to the WebViews and HTML would make it easier to rollout multi-platform support.

HTML in its version 5 supports new features like native video support, local databases, and better graphic support. One of the new features is limited support of offline storage (see Hickson (2011)). Storage capacity comes with a standard quota of 5 MB per domain, so while this might not be usable to data intense applications like a MOOC, at least some web browsers allow to turn off this limit and allow unlimited offline storage (limited by local resources). Currently, only a few users are aware of these new possibilities, so the browser will not be the first place where they will look for offline capable functions. However, these possibilities could be used to solve issues like using a device with non-steady Internet connection, for example in a train or a car.

Based on these features HTML5 adds one more possible solution in addition to native and hybrid apps that might be considered from MOOC providers.

5.2 Evaluation of App-Usage

After the app is launched, further research will be needed to evaluate the usage and impact of the app. Questions to be asked are:

- How will users learning behavior change given the additional mobile and offline access?
- How many users will use these new possibilities?
- And for those users using the new app, will there be a relevant impact on their learning achievements, both in comparison to previous courses they visited and in comparison to the rest of the users accessing the course through the web browser only.

ACKNOWLEDGEMENT

The authors would like to thank the teams at openSAP and the 3 Screen Solutions GmbH & Co. KG.

REFERENCES

de Waard, I. et al., 2011a. Using mLearning and MOOCs to understand chaos, emergence, and complexity in education. The International Review of Research in Open and Distance Learning, [S.l.], v. 12, n. 7, p. 94-115, oct. 2011. ISSN 1492-3831. Available at: http://www.irrodl.org/index.php/irrodl/article/view/1046/2026. [Online; last accessed: 01/28 /2014].

de Waard, I., Koutropoulos, A., Özdamar K., N., Abajian, S. C., Hogue, R., Rodriguez, C. O., & Gallagher, M. S., 2011b. Exploring the MOOC format as a pedagogical approach for mLearning. Proceedings from mLearn.

edX Facebook, 2013. *https://www.facebook.com/EdxOnline/posts/374074342723934*. [Online; last accessed 11/25/2013].

Forbes. 2013a. *Coursera Hits 4 Million Students – And Triples Its Funding, http://onforb.es/15w7FJt*. [Online; last accessed 11/25/2013].

Forbes, 2013b. The Worlds Biggest Public Companies. *(Forbes) http://goo.gl/Tw8rrc*. [Online; last accessed 11/25/2013].

Fowler, M., & Foemmel, M., 2006. Continuous Integration. *(Thought-Works) http://www.thoughtworks.com/ ContinuousIntegration.pdf* . [Online; last accessed 11/25/2013].

Grünewald, F., Mazandarani, E., Meinel, C., Teusner, R., Totschnig, M., & Willems, C., 2013. openHPI: Soziales und Praktisches Lernen im Kontext eines MOOC. *DeLFI 2013–Die 11. e-Learning Fachtagung Informatik*.

Hickson, I., 2011. Web storage. *Draft, W3C, October 4th 2011, http://dev.w3.org/html5/webstorage/* . [Online; last accessed 11/25/2013].

Hwang, G. J., Tsai, C. C., & Yang, S. J. H., 2008. Criteria, strategies and research issues of context-aware ubiquitous learning. Educational Technology & Society, 11(2), pp 81-91.

Hwang, G. J., Tsai, C. C., 2011. Research trends in mobile and ubiquitous learning: A review of publications in selected journals from 2001 to 2010 in British Journal of Education Technology 42 (4), pp E65-E60

Masse, M., 2011. *REST API design rulebook*. O'Reilly.

Meinel, C., & Willems, C., 2013. *openHPI : the MOOC offer at Hasso Plattner Institute, Hasso-Plattner-Institut für Softwaresystemtechnik: Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam ; Nr. 80*. Potsdam, Germany: University Press.

openHPI Forum, 2013. *https://openhpi.de/courses/7/discussion topics/2669 (translated from German original)*. [Online; last accessed 11/25/2013].

openSAP Forum, 2013. *https://open.sap.com/courses/3/discussion topics/3177*. [Online; last accessed 11/25/2013].

Schulmeister, R., 2013. Das Beginn und das Ende von Open, pp 17-59 in MOOCs – Massice Open Online Courses, Offene Bildung oder Geschäftsmodell?, Waxmann Verlag GmbH

Udacity, 2013. Forum. *http://forums.udacity.com/questions/4000069/course-content-on-android*. [Online; last accessed 11/25/2013].

Willems, C., Renz, J., Staubitz, T., & Meinel, C., 2013. Reflections on Enrollment Numbers and Success Rates at the openHPI MOOC Platform. *In: submitted*.

Willnecker, F., Ismailovic, D., & Maison, W., 2012. Architekturen mobiler Multiplattform-Apps. *pp 403–417 in: Smart Mobile Apps*. Springer.