

Evaluation der Akzeptanz mobiler Lernszenarien auf Basis plattformunabhängiger Webtechnologien

Jan Eberwein, Irawan Nurhas, Stefan Steinwasser, Marc Jansen

Institut Informatik
Hochschule Ruhr West
Tannenstr. 43
46240 Bottrop

{jan.eberwein, irawan.nurhas, stefan.steinwasser}@stud.hs-ruhrwest.de
marc.jansen@hs-ruhrwest.de

Abstract: In den letzten Jahren nahm die Nutzung mobiler Geräte stark zu, was auch zu einer Verbreitung mobiler Geräte in mobilen Lernszenarien geführt hat. Aufgrund der Heterogenität der am Markt befindlichen mobilen Geräte wird zur Softwareentwicklung häufig ein plattformunabhängiger Ansatz gewählt. Hierzu gibt es im Wesentlichen zwei unterschiedliche Ansätze, die beide auf Webtechnologien basieren. Entweder es werden Applikationen mit spezifischen UI Frameworks für mobile Geräte oder mit responsive Ansätzen, parallel für mobile und stationäre Geräte, entwickelt. Um beide Ansätze vergleichen zu können, stellt dieses Papier eine Evaluation der Akzeptanz mobiler Lernszenarien auf Basis plattformunabhängiger Webtechnologien dar.

1 Einleitung

Neue Frameworks und Technologien zur Softwareentwicklung im mobilen Umfeld gibt es immer mehr. Daher stellt sich die Frage, ob die Implementierung bevorzugt mittels einer auf mobile Plattformen ausgelegten Technologie durchgeführt wird, oder ob ein responsive Ansatz für die Umsetzung praktikabler ist. Mobile Endgeräte, wie Smartphones und Tablets, verbreiten sich immer weiter. Aufgrund ihrer unterschiedlichen Betriebssysteme ist ein nativer Entwicklungsprozess in vielen Fällen allerdings nicht zielführend, da die Software ansonsten jeweils spezifisch für alle gängigen Plattformen entwickelt werden müsste oder einzelne Plattformen nicht mit der Anwendung versorgt werden können. Von daher werden heutzutage häufig plattformunabhängige Ansätze im mobilen Umfeld gewählt. Hier gibt es im Wesentlichen noch die Unterscheidung zwischen der Entwicklung spezifisch für mobile Plattformen und sogenannte responsive Ansätze, die die Softwareentwicklung für mobile und stationäre Systeme parallel erlauben. Um diese beiden Ansätze miteinander zu vergleichen, wurde auf Basis dessen eine App zur Durchführung einer QOC-Modellierung¹ [ML91] zur Diskussionsunter-

¹ Zu einer Frage (Question) werden mindestens 2 Optionen (Options) angeboten und mithilfe von Kriterien (Criteria) sowie Gewichtungsfaktoren soll die beste Option zum Beantworten dieser bestimmt werden.

stützung bzw. Entscheidungsfindung implementiert. Beide Versionen wurden im Rahmen einer Lehrveranstaltung durch 12 Teilnehmer evaluiert. In den folgenden Abschnitten wird zuerst der aktuelle Stand der Technik in Bezug auf die beiden Ansätze dargestellt. Darauf folgt eine kurze Beschreibung der Implementierung, gefolgt von den Ergebnissen der Evaluation und einer kurzen Diskussion. Abschließend wird ein Ausblick auf kommende Arbeiten präsentiert.

2 State-of-Art

Smartphones und Tablets, allgemein bezeichnet als Smartdevices [Ja06], sind zu einem festen Bestandteil des alltäglichen Lebens geworden. Nach BITKOM besaßen im August 2013 40% aller Bundesbürger über 14 Jahre ein Smartdevice [BI13]. Die primär vertretenen Smartdevice-Betriebssysteme sind iOS von Apple, Android von Google und Windows Phone von Microsoft [ID14], weshalb plattformunabhängige Technologien benötigt werden, um die Benutzer mit Applikationen zu versorgen und gleichzeitig keine parallelen Entwicklungsstränge betreiben zu müssen. Im Bereich der Webapplikationen (kurz WebApp) werden daher Frameworks verwendet, um mit HTML5, JavaScript und CSS3 [Ba11] plattformunabhängige WebApps zu entwickeln. Auch im Bereich mobiler Lernszenarien gibt es bereits Initiativen zur plattformunabhängigen Implementierung. In [GJ11] beschreiben die Autoren ein plattformunabhängiges System zur Integration von Quizze aus einem Moodle System in eine mobile Applikation. In [ALY11] wurde über einen längeren Zeitraum eine mobile App für verschiedene Plattformen zum Erlernen der chinesischen Sprache für internationale Studierende evaluiert. Das Resultat dieser Evaluation zeigt eine Sinnhaftigkeit solcher Apps. Des Weiteren gibt es Ansätze zur Evaluation bestehender Apps, z.B. in Richtung des Benutzerverhaltens sowie der Akzeptanz und Nützlichkeit der App [Ku14]. Da die Entwicklung solcher Smartdevice-Applikationen zu einem wichtigen Standfuß in der Wirtschaft geworden ist, existieren einige Bücher, wie z.B. [NB13], die App-Entwicklern wichtige Aspekte der Applikationsentwicklung bei mobilen Geräten aufzeigt.

3 Implementation

Um beide relevanten Entwicklungsansätze vergleichen zu können, wurde die QOC-App sowohl mit einem responsive Web Framework als auch mit einem mobilen UI Framework implementiert, um festzustellen, welcher der plattformunabhängigen Ansätze bei Benutzern eine höhere Akzeptanz erfährt.

3.1 Architektur

Die Hosts der Benutzer sind Smartdevices, welche über einen Webbrowser auf die WebApp zugreifen. Als Frontend wird entweder das responsive Web oder mobile UI Framework verwendet. Die Applikation greift über eine ReST basierte Middleware auf eine Datenbank zu. Bei beiden Implementationen wird als Datenbank MongoDB

verwendet, um die JSON-Objekte der jeweiligen App abzuspeichern. Die Architektur beider Implementierungen ist in Abbildung 1 dargestellt.

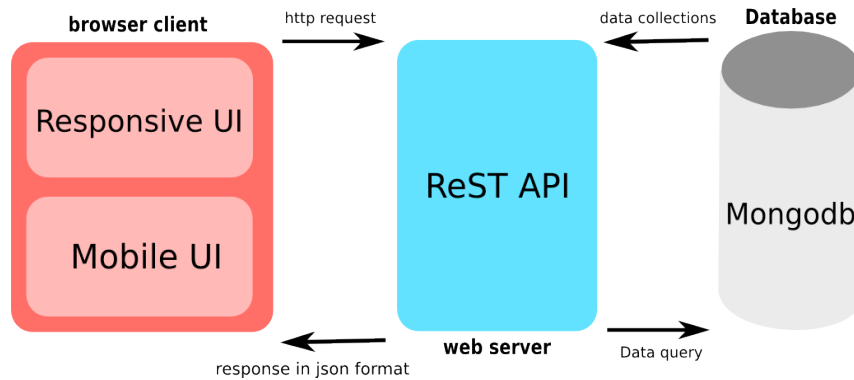


Abbildung 1: Architektur der Anwendung

3.2 Umsetzung mittels eines responsive Web Frameworks

Zur Entwicklung der responsive Variante der mobilen App wurden mehrere Javascript Frameworks verwendet. Als Hauptframework für die Benutzeroberfläche wird Bootstrap, jQuery für die Interaktion zwischen dem Benutzer und der Anwendung und NVD3.js zum Erzeugen interaktiver Grafiken eingesetzt. Des Weiteren werden Renderer für die Seitennavigation, Infinite-Scrolling und Pagination verwendet. Durch die Verwendung der genannten Frameworks wurde die Bedienbarkeit der responsive Variante der mobilen App optimiert. Die Abbildung 2 zeigt links die Applikation auf einem Desktop und rechts die Darstellung auf einem mobilen Gerät.

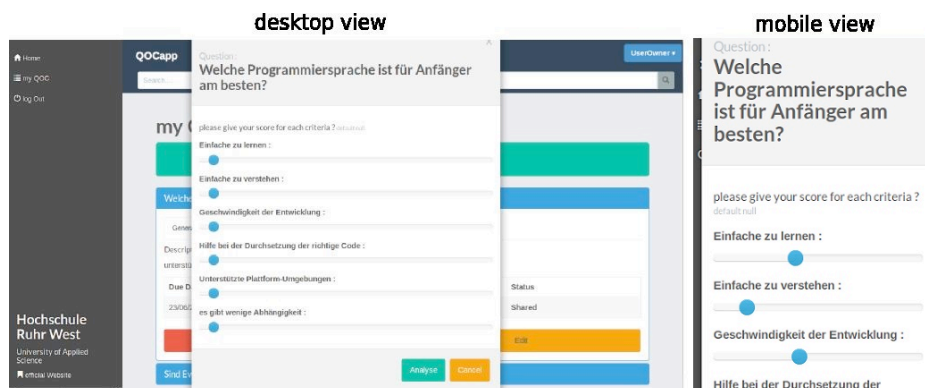


Abbildung 2: Responsive App - Bewertungsfenster

3.3 Umsetzung mittels eines mobilen UI Frameworks

Für die Umsetzung der mobilen App auf Basis eines mobilen UI Frameworks wurde ein modernes, MVVM-Modell basiertes Open-Source Framework (AngularJS) mit dazu passenden UI Komponenten (ionic) verwendet. Das Besondere am ionic-Framework ist die Eigenschaft, dass es ein Bedienungskonzept, wie man es vom Smartphone gewöhnt ist, umsetzt und auch die einzelnen Bedienelemente entsprechend Abbildung 3 gestaltet. Es ist außerdem durch die Einbindung von Cordova als native App für Android und iOS kompilierbar und so für die Verbreitung in den jeweiligen App Stores geeignet.

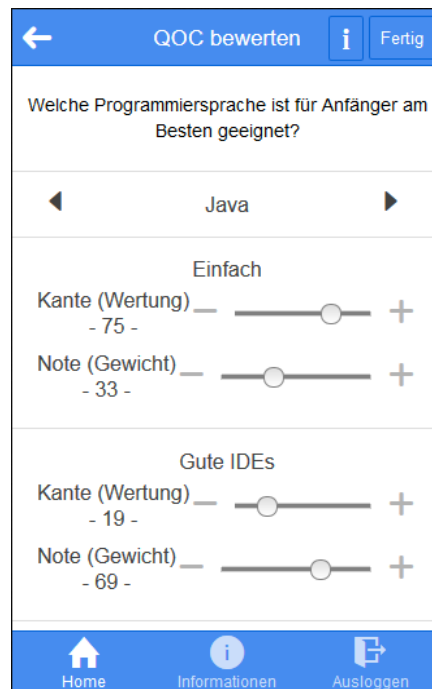


Abbildung 3: Mobile App - Bewertungsfenster

Mit Hilfe der mobilen Applikation können die Benutzer sowohl einzelne Optionen als auch zugehörige Kriterien sammeln und bewerten. Abschließend wird den Benutzern eine Auswertung der jeweiligen Optionen, inkl. der präferierten Option, präsentiert.

4 Evaluation

Die beiden Implementierungen der QOC-Apps wurden im Rahmen einer Vorlesung mit Studenten aus dem vierten Semester auf Basis von TAM (Perceived Ease of Use, Perceived Usefulness) [Da89] und SUS (System Usability Scale) [BR96] evaluiert. Die Studenten wurden hierzu jeweils in zwei Gruppen mit jeweils sechs Studenten aufgeteilt.

Jede Gruppe bekam in zwei Durchläufen jeweils eine der beiden Implementationen zum Testen. Nach jedem Durchgang wurden die Studenten aufgefordert, die entsprechenden Fragebögen mit den TAM- und SUS-Fragen zu beantworten. Im Folgenden werden zuerst die Ergebnisse aus der jeweiligen Evaluation dargestellt und anschließend kurz diskutiert.

4.1 Ergebnisse aus der Evaluation nach TAM

Die an der Evaluation teilnehmenden Studenten haben Fragebögen mit den in Tabelle 1 stehenden Fragen zum Bereich Perceived Usefulness (PU) und Perceived Ease of Use (PEU) beantwortet. Für jede dieser Fragen mussten die Studenten auf einer 5-Level-Likert-Skala [Li32] ihre Einschätzung angeben.

Tabelle 1: Fragebogen für PEU und PU

PU – Perceived Usefulness	PEU – Perceived Ease of Use
F1.1: Die QOC-App hilft mir bei der Entscheidungsfindung.	F2.1: Die QOC-App ist leicht zu erlernen.
F1.2: Wenn ich auf die QOCs jederzeit und überall zugreifen kann, nutze ich das System häufiger, was das Umsetzen der Arbeit beschleunigt.	F2.2: Umgekehrt gefragt: Die QOC-App ist starr und unflexibel in der Handhabung.
F1.3: Der Einsatz der QOC-App innerhalb von Projekten hilft, die Arbeit effizienter zu realisieren.	F2.3: Die QOC-App ist einfach zu bedienen und verhält sich wie erwartet.
F1.4: Umgekehrt gefragt: Die Nutzung der QOC-App senkt meine Produktivität innerhalb einer Projekt- bzw. Arbeitsgruppe.	F2.4: Die Handhabung der QOC-App ist für mich leicht zu verstehen.
F1.5: Die QOC-App erleichtert das Erreichen der Projekt- bzw. Arbeitsziele.	F2.5: Umgekehrt gefragt: Es ist schwer für mich zu behalten, wie man QOCs in der App erstellt und bewertet.
F1.6: Umgekehrt gefragt: Die QOC-App erschwert das Erreichen der Projekt- bzw. Arbeitsziele.	F2.6: Umgekehrt gefragt: Das Arbeiten mit der QOC-App ist anstrengend.
F1.7: Die QOC-App hilft mir, die Ziele eines Projekts oder meiner Arbeit besser zu identifizieren.	F2.7: Insgesamt finde ich die QOC-App benutzerfreundlich.
F1.8: Insgesamt finde ich die Anwendung der QOC-App innerhalb von Projekt- und Arbeitsgruppen sinnvoll.	

Die jeweiligen Ergebnisse wurden mit Hilfe des Cronbachs Alpha [Cr51] auf ihre Reliabilität überprüft. Für die mobile App ergibt sich der entsprechende Wert zu $\alpha=0.88$ sowie für die responsive App zu $\alpha=0.85$. Beide Werte zeigen eine ausreichende Reliabilität für die weiteren Analysen. Ein signifikanter Unterschied konnte im Bereich PEU zwischen der responsive Implementierung ($M=3,71$, $SD=0,72$) und der Implementierung auf Basis des mobilen UI Frameworks ($M=4,065$, $SD=0,72$) festgestellt werden, mit $t(13)=4,51$ und $p=0,001$. Der Bereich PU hingegen weist keinen signifikanten Unterschied zwischen der responsive Implementierung ($M=3,76$, $SD=0,52$) und dem mobilen UI Framework ($M=3,78$, $SD=0,59$) auf. Die nachfolgende Abbildung 4 zeigt die jeweiligen Mittelwerte (M) sowie die zugehörige Standardabweichung (SD) der einzelnen Fragen aus dem PU und dem PEU Fragebogen.

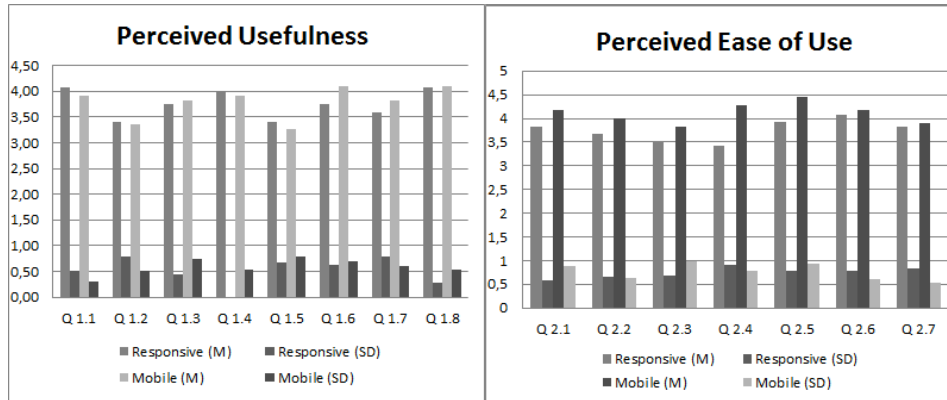


Abbildung 4: Mittelwerte der Fragen von PU und PEU

Die PEU Ergebnisse zeigen somit deutlich, dass die Benutzerfreundlichkeit der QOC-App mit einer mobilen UI Implementation höher ist als mit einer responsive Implementation.

4.2 Ergebnisse aus der Evaluation nach SUS

Ebenso wie im Bereich der TAM-Analyse wurden die Angaben der Fragebögen mit Cronbachs Alpha auf die Reliabilität getestet. Der Cronbachs Alpha Wert für die mobile App besitzt den Wert $\alpha=0.91$, während der entsprechende Wert des responsive Ansatzes einen Wert von $\alpha=0.84$, ergibt. Beide Werte zeigen, dass die Daten verlässlich sind. Der ermittelte SUS-Mittelwert der mobilen Applikation beträgt 74,79 und der der responsive Applikation 73,13. Die Tabelle 2 zeigt die Klassifizierung (Gradescale) der SUS-Mittelwerte nach [BKM09] und [SL12, S. 204].

Tabelle 2: SUS score nach Bangor und Sauro

Web-App type	SUS score	Bangor		Sauro	
		SUS range	Grade	SUS range	Grade
Responsive Web Application	73.13	70 - 79	C	72.6 - 74	B-
Mobile Web Application	74.79	70 - 79	C	74.1 - 77.1	B

Basierend auf den Werten aus [BKM09, S. 121] erhalten beide Anwendungen den Grad C, was bedeutet, dass beide Anwendungen vom Benutzer akzeptiert und als eine einfach zu benutzende Anwendung klassifiziert werden. Dabei betragen die Mittelwerte

(Learnability²=86,46, Usability=71,88) bei der mobilen Applikation und bei der responsive Applikation (Learnability=83,33, Usability=70,57) (vgl. Abbildung 5).

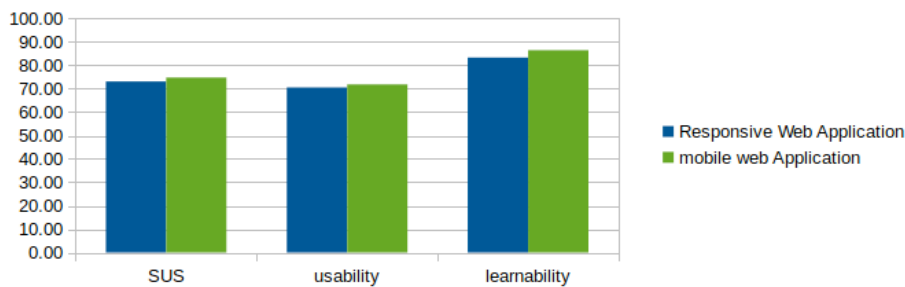


Abbildung 5: SUS Vergleich der responsive und mobilen App

Die obere Grafik zeigt, dass die Akzeptanzrate sich zwischen der responsive Web Application und der mobile App kaum unterscheidet. Darüber hinaus haben beide Anwendungen die Eigenschaft, dass Learnability der einflussreichste Faktor ist und dass Learnability und Usability eine hohe Pearson- Korrelation von 0,69 (responsive) und 0.9 (mobile) aufweist.

4.3 Diskussion

Insgesamt zeigen diese Ergebnisse, dass sowohl auf Basis von responsive Ansätzen die Implementation plattformunabhängiger mobiler Lernszenarien möglich scheint sowie auch auf Basis von rein mobilen Entwicklungsmethoden. Obwohl sich im Bereich der Einfachheit der Anwendung (PEU) ein signifikanter Unterschied zu Gunsten der rein mobilen Entwicklung gezeigt hat, wirkt sich dies auf Basis unserer SUS-Analyse nicht negativ auf die Akzeptanzrate der Benutzer aus, bei der wir keinen signifikanten Unterschied feststellen konnten und die sich für beide Implementierungen auf hohem Niveau bewegt. Um zu klären, ob es sich hier um einen Einzelfall handelt, müssen weitere Apps mit beiden Implementierungsmethoden entwickelt und durch eine größere heterogenere Testgruppe evaluiert werden.

5 Ausblick

Auf Basis der in diesem Papier dargestellten Ergebnisse, lassen sich Apps für mobile Lernszenarien sowohl auf Basis rein mobiler plattformunabhängiger Technologien als auch auf Basis von responsive Ansätzen entwickeln. Für beide Ansätze ergaben sich vergleichbare Akzeptanzwerte bei den späteren Benutzern, mit Vorteilen für mobile Ansätze in Bezug auf die einfache Handhabbarkeit. Auf der anderen Seite, ergaben sich

² Der Learnability-Wert in SUS basiert auf den Kenntnissen der Benutzer, wohingegen PEU und PU noch die persönliche Meinung miteinfließen lassen.

im Bereich der responsive Ansätze deutliche Vorteile in Bezug auf die einzusetzenden Entwicklerressourcen, wenn parallel zu der mobilen App auch eine Implementierung für stationäre Systeme geplant ist. Letztendlich kann die Entscheidung für den einen oder anderen Ansatz von anderen Faktoren, wie z.B. Kenntnissen in dem jeweiligen Entwicklerteam und zur Verfügung stehenden Entwicklerressourcen, abhängig gemacht werden.

Literaturverzeichnis

- [ALY11] Akinkuolie, Babatunde B.; Lin, Chia F.; Yuan, Shyan-Ming: A Cross-Platform Mobile Learning System Using QT SDK Framework. In: ICSEC, 2011. S. 163–167
- [Ba11] Baloian, N. et. al.: The Future Role of HTML5 in Mobile Learning Scenarios. In: Proceedings of Mobile Learning Conference, 2011.
- [BI13] BITKOM, Presseinformation – 63 Millionen Handy-Besitzer in Deutschland, August 2013 – URL: http://www.bitkom.org/files/documents/PI_BITKOM_Handy-Verbreitung_in_Deutschland_26_08_2013.pdf.
- [BKM09] Bangor, A.; Kortum, P.; Miller, J.: Determining what individual SUS scores mean: Adding an adjective rating scale. In: Journal of usability studies, 2009. 4: Nr. 3: S. 114–123.
- [Br96] Brooke, J.: SUS-A quick and dirty usability scale. In: Usability evaluation in industry, 1996. 189: S. 194 ff.
- [Cr51] Cronbach, L.J.: Coefficient Alpha and the Internal Structure of Tests. In: Psychometrika, 1951. 16: S. 297-334.
- [Da89] Davis, Fred D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. In: MIS quarterly, 1989. S. 319–340
- [GJ11] Geisler, S.; Jansen, M.: Getting Serious About a Platform Independent Application for the Usage of Mobile Moodle Quizzes: A Case Study. In: Workshop Proceedings of the Workshop Mobile Learning at Delfi, 2011.
- [ID14] IDC Worldwide Quarterly Mobile Phone Tracker (Q1 2014) – URL: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>.
- [Ja06] Jansen, M.: Integrating Smart Devices in Java Applications, Verlag Dr. Kovac, Hamburg, 2006.
- [Ku14] Kuhn, E. et. al.: Preliminary evaluation of PTSD Coach, a smartphone app for post-traumatic stress symptoms. In: Military medicine 179, 2014. 1: S. 12–18.
- [Li32] Likert, R.: A Technique for the Measurement of Attitudes. In: Archives of Psychology, 1932. 140: S. 1-55.
- [ML91] MacLean, A.; Young, R.M.; Bellotti, V.M.E.; Moran, T.P.: Questions, Options, and Criteria: Elements of Design Space Analysis. Human-Computer Interaction, 1991. 6(3&4): S. 201-250.
- [NB13] Nielsen, J.; Budiu, R.: Mobile Usability: Für iPhone, iPad, Android, Kindle. mitp/bhv, 2013. – ISBN 978-3-8266-9503-2
- [SL12] Sauro, J.; Lewis, J. R.: Quantifying the user experience: Practical statistics for user research. Elsevier, 2012.