

HTML5 oder Native? Die Antwort für Mobile!

20 Dezember 2013 | **User Experience** | [Nikolaos Kaintantzis](#)

Lesezeit: 2 Minutes

Zur Frage „HTML5 oder Rich Client“ auf dem Desktop wurde viel geschrieben und empfohlen. Auch vom mir gibt es [Vorträge](#) und [Fachartikel](#). Als ich mich aufmachte, das Thema für die „mobile Welt“ zu beleuchten, erlebte ich eine Überraschung. Meine Empfehlungen für Desktop-Anwendungen, passten nicht in die mobile Welt. Aus den Erkenntnissen ist ein [Vortrag](#) für die [GOTO Zurich](#) entstanden.

Was ist anders in Mobile als in Desktop?

- **User Experience:** Die Use Cases sind verschieden. Schnelle Datenerfassung und tägliches stundenlanges Arbeiten mit der Applikation ist ein häufiger Use Case im Desktop-Bereich. Shortcuts und unmittelbare Antwortzeit ist hier gefordert. Dieser Use Case ist für die mobile Welt inexistent. Zwar gilt auch für mobile Geräte: „Native = bessere User Experience“, doch es gibt weniger Use Cases, die Native zwingend nach sich ziehen.
- **Unterschiedliche Plattformen:** Während im Firmenumfeld hauptsächlich für Windows-PCs entwickelt wird, gibt es in der mobilen Welt noch keinen Sieger und auch keine gemeinsame Plattform wie Java.
Hinzu kommen unterschiedliche Styles, Platzierung von Navigationselementen und Hardware Buttons anhängig von der Plattform.
- **Hohe Gerätevielfalt** und Segmentierung betreffend Auflösungen und Bildschirmgrößen: Auch die Apples Welt passt sich an. Vorbei ist die Zeit mit nur einer Bildschirmgröße. Im [Open Signal Report](#) wird das auf den Punkt gebracht.
- **Application Life Cycle management (ALM):** Auf dem Desktop bedient man eine oder maximal die letzten zwei Windows-Versionen. In der mobilen Welt hingegen müssen mindestens iOS and Android bedient werden. Je nach Markt auch Windows Phone oder BlackBerry. Dies ist aber noch nicht alles. Während beim iPhone neue Versionen sehr schnell beim Anwender installiert werden und demnach der Support der letzten zwei Versionen genügt (iOS7 hat heute über 77% [Verbreitung](#), und iOS6 knapp 20%), braucht es bei Android schon drei Versionen um genügend Anwender abzudecken.
Bei allen Plattformen muss die zukünftige Version ebenfalls unterstützt werden. Brauchen Sie zu lange bis Ihre App auch mit der neuen Version des Betriebssystems funktioniert, bekommen Sie schnell schlechte Bewertungen in den Stores. Zusammengezählt müssen Sie also pro Plattform drei bis vier Versionen unterstützen.
- **App-Stores:** Die Präsenz in den App-Stores war zu Beginn des Smart-Phone-Zeitalters extrem wichtig. Selbst Applikationen, die besser als responsive oder adaptive Seiten

implementiert worden wären (wie z.B. Zeitungs-Portale), kamen als Apps in die Stores. Die Präsenz im Store kann immer noch als sehr wichtig erachtet werden.

- **Monetarisierung:** Die Stores haben hier vieles erleichtert. Aber es gibt auch andere Wege der Monetarisierung.

HTML5 oder Native? Was ist nun die Antwort?

Bei User Experience — und diese wird erwartet — liegt Native ganz klar vorne. Monetarisierung ist bei Native einfacher. Bei allen anderen Punkte ist es eher oder klar HTML5.

Und was wenn z.B. gute User Experience wichtig ist und sehr viele Plattformen unterstützt werden müssen? Dann ist die Frage „HTML5 oder Native?“ mit „HTML5 und Native!“ zu beantworten. Wichtige Use Cases werden Native und weniger wichtige in HTML5 implementiert.

Links aus dem Artikel:

- Vortrag “Wie steht es um RIA? – RIA aus der Sicht der Software-Entwicklung”:
http://www.jug.ch/events/slides/100120_Nikolaos_Kaintantzis.pdf
- Ix-Artikel „Reicher werden: Wie HTML5 Rich Internet Applications verändert“:
<http://de.slideshare.net/Zuehlke/ix-12-2011reicherwerdenkaintantzis>
- GOTO Zurich Vortrag „HTML5 versus native“:
<http://de.slideshare.net/xnka/2013-0411-nikolaoskaintantzishtml5ornative>
- Android Fragmentation: <http://opensignal.com/reports/fragmentation.php>
- iOS 7 adoption: https://mixpanel.com/trends/#report/ios_7