MOBILE APP DEVELOPMENT: NATIVE OR WEB?

Adrian Holzer
University of Lausanne
adrian.holzer@gmail.com

Jan Ondrus
ESSEC Business School
ondrus@essec.edu

Abstract

The mobile application market has generated significant interest in the business community. One major issue when developing mobile applications is the fragmentation of the mobile device market. This fragmentation and the fierce competition between the different mobile platforms (e.g., iOS, Android, Windows, Blackberry) preclude mobile application portability between operating systems. Therefore, companies have to evaluate what development strategies are suitable for them: i) mobile native apps, ii) web apps, or iii) hybrid apps. In order to address this issue, this short research-in-progress paper presents the first steps towards the design of a decision support system (DSS) to evaluate these different mobile application development strategies. Using a design science approach, this research effort establishes a list of relevant criteria to differentiate the current strategies. As each strategy has benefits and drawbacks, we provide a preliminary step towards the creation of a prescriptive model to enable better decision-making while making trade-offs possible.

Keywords: mobile application, decision support system, development strategies

1. Introduction

With an estimated 420 million units shipped in 2011 (IMS Research 2011), the smartphone has incontestably become one popular consumer electronic device. Smartphones already represent 28% of all handsets. With their full-fledged web browser and their access to broadband Internet through the 3G/4G networks, they significantly increased the use of the mobile Internet. Simultaneously, smartphones have boosted the mobile application market. Analysts estimate its overall value in 2010 at USD 6.8 billion and it is expected to reach as much as USD 25 billion by 2015 (MarketsandMarkets 2012). As for the biggest application retailer to date, Apple hosts over 650’000 applications (Apple Press Release, 26 June 2012). As the application market has grown significantly in less than 5 years, the current fragmentation of the device market (i.e., large variety of devices) is a central issue for companies when choosing their mobile application development strategies. Traditionally, providers would start by offering an iPhone application and then, if they have enough financial and human resources, they would develop for other platforms (Maxwell 2011). With the growing customer base on Android and eventually other platforms, it is legitimate for providers to explore alternative development strategies. One accepted solution is to develop mobile web applications. Farrell and Nezlek (2007) already predicted that rich web applications with Ajax would rival native apps on the desktop. However, in September 2012, Mark Zuckerberg stated that one of Facebook’s major mistakes was to focus too much on HTML5, rather than on mobile native apps (Olanoff 2012).
Mobile applications can be native, web-based, or hybrid (Charland and Leroux 2011). Native apps are created and compiled using specific programming languages to run uniquely on one mobile platform. Web apps are built with web languages (e.g., HTML, CSS, Javascript, AJAX) and are interpreted using any web browser. Hybrid apps combine web apps with a native app container. Each alternative offers pros and cons. The challenge for companies is to select the alternative, which corresponds best to their needs, priorities, and constraints. This leads to the following, largely unresolved research question: How to design a decision support system (DSS) that can guide companies in their mobile development strategy decisions?

Using a design science approach (Hevner et al. 2004), this research aims at creating a DSS, which comprises a set of relevant criteria to support developers when assessing the three current strategic development alternatives (i.e., native, web, and hybrid). We derive the criteria from the scarce academic literature on the topic and archival data found on the Internet (specialized websites, blogs, and development forums). We also use data collected during exploratory interviews with French experts working for different industries. To structure our analysis, we categorize the criteria based on the life cycle of an application (i.e. development, distribution, and usage). The implication of this research is to provide the first step towards the construction of a decision model to assist multiple criteria decision-making. This model aims at supporting the strategic analysis phase when building mobile applications. The paper is organized as follows. Section 2 describes the research approach and the data we use in order to design the decision model. Then, Section 3 presents the criteria used in our analysis. Following Section 4, which provides an overview of the differences between strategies, Section 5 concludes the paper and exposes our further research plans.

2. Building the decision model: a design science approach

The construction of a decision model to support mobile application development may look similar to theory building. However, the aim of our model is substantially different as the goal is not to explain or understand a phenomenon, but about solving a problem. It is about finding the relevant dimensions to enable better decision-making when developing mobile applications. This research effort is a preliminary step to ease the development of a DSS using a multiple criteria approach. The first step is to provide the set of conflicting criteria for the decision model to support the informed judgments of possible alternatives while making trade-offs possible (Keeney and Raiffa 1976). March and Smith (1995) claim that two kind of scientific approaches are legitimate in information system research. The first one seeks to understand and explain reality (i.e., social sciences approach). The second one aims at creating artifacts that serve humans purposes (i.e., design science approach). Research in information systems often integrates both social and design science approaches. March and Smith (1995) prescribe four research activities: i) build, ii) evaluate, iii) theorize, and iv) justify. The first two belong to the design science approach whereas the other two are related to a social sciences approach.

In this paper, we only focus on the design science components. We first build the assessment framework using multiple sources of evidence such as archival data and industry reports found in the literature (Anik and Baykoç 2011, Buyukozkan and Ruan 2008, Lai et al. 2002, Moraga et al. 2006) and on the Internet (Closs 2011, Lewis 2011, Siegler 2011, Mahemoff 2011, Claburn 2011). As the topic is emerging, the academic literature is still scarce. Therefore, we searched specialized mobile development websites, blogs, and forums to extract the information needed to
establish a first set of criteria. We then use these criteria to the different development strategies building on the qualitative data collected during 8 interviews of French experts with different industry backgrounds (telco, entertainment, software, mobile hardware). These interviews were carried out from April to June 2011 and lasted in average 1 hour each. For reasons of anonymity, several experts asked not to display the name of their respective company.

3. A multi-criteria decision model for mobile development strategies
In order to evaluate the strength and weaknesses of our three types of strategies, we establish a list of criteria based on previous related work as mentioned in the previous section. In contrast to previous research, we mapped the criteria specifically to the mobile application life cycle: development, distribution, and usage.

3.1 Development criteria
Salient differences exist between development processes and the types of application developed. We divide them into the following aspects: technical specifications, development support, portability, and maintenance.

Technical specifications. Nowadays, a number of mobile applications include location, accelerometer, or device motion features. However, development strategies are not on always on an equal footstep with respect to access to technical specifications (Lai et al. 2002, Anik and Bayok 2011). Web apps are more restrictive since their execution is mainly confined to an Internet browser. Therefore they lack several technical specifications compared to native and hybrid apps, such as access to camera (Mahemoff 2011), push notification (Tim Closs 2011), extended offline execution capability and multitasking. These issues are seen as major drawbacks of web apps by several of our interviewees. The situation is continuously evolving as features can be added to web browsers by recompiling it with the added functionality if the OS provider authorizes access via APIs (Charland and Leroux, 2011). A number of experts expect web apps to soon close the gap in terms of technical specifications (Mahemoff 2011) others contend that web apps will be left behind for a long time (Siegler 2011). Through their native container, hybrid apps can also take advantage of all the technology available on the phone. Hybrid apps are seen by many as a possible alternative to bridging the technical features gap between web apps and native apps (Claburn 2011, Mahemoff 2011, Siegler 2011).

Development support. As the development process can be greatly enhanced by suitable tools, it is important to evaluate what kind of support a given platform provides (Lai et al. 2002, Anik and Baykoç 2011, Goadrich and Rogers 2011). Native apps have the most advanced development tools at their disposal. These include SDKs, emulators, debugging tools and performance analysis tools. Support for web apps is more limited. Charland and Leroux (2011) report that web app development has become easier than it used to be. When they started developing for these mobile browsers, no framework worked properly. Nowadays there are more frameworks that allow easier support for multiple target web-apps. Hybrid apps enjoy advanced programming support for their native component, but most rely on standard web support for their web component.

Portability. As smartphones run different operating systems, applications portability is an essential factor to consider when evaluating mobile development strategies (Anik and Baykoç
Native apps are platform-specific and must be rewritten almost from scratch for every platform, therefore increasing development costs. This concern is seen as the major drawback of native apps by one of our interviewees. Conversely, web apps follow web standards that aim at becoming entirely compatible across platforms. An interviewee argues that this factor can contribute to the success of web apps, since web browsers can play the role of common denominators among platforms. However, some interviewees were also concerned about the insufficient compatibility across browsers. The portability of hybrid apps is in-between native and web apps, since their native component suffers from the same limitations as native applications and their web component can benefit from the same portability as web apps. Note that some technologies allow to translate application code into different native languages in order to address the issue of code portability for native apps (e.g., Appcelerator: www.appcelerator.com/products/). If these become successful, they might partly solve the portability issue for native applications.

**Maintenance.** Application maintenance depends on both the reusability of code and on the distribution platform. Some native distribution platforms require validation before a new version can be rolled out. Web apps, on the other hand, can be updated anytime with no delay. This makes web apps especially attractive when code must be updated on several platforms (Charland and Leroux 2011). In our interviews, the editor in chief of the digital version of a newspaper finds the delay between submission and acceptance on the Apple’s AppStore to be a major drawback for native apps. He would like to be able to update his native application for major current events for example. This is not possible due to the delay. In this regard web apps could solve his concern. Hybrid apps are midway between both worlds, the web component can be easily updated and maintained, but the native part suffers the same drawback as native apps.

### 3.2 Distribution criteria

Distribution factors can be grouped in three main issues: *access to consumers, ease of distribution* and *monetization*.

**Access to consumers.** Reaching consumers is a central issue. Consumers should be able to find and download applications easily. Centralized portals that work as one-stop shopping centers with the one-click purchase mechanisms proved to be successful mechanisms to increase the mobile application market (Holzer and Ondrus 2011). Native apps and hybrid apps have an advantage over web apps, since they have established application portals and consumers are now used to download and use native apps and hybrid apps. Web apps do not really enjoy such portals yet, however this could change in the future (Mahemoff 2011). For one of our interviewees, centralized portals such as Apple’s AppStore through which it is possible to reach a large pool of users, is a major advantage for native applications, especial native iPhone apps. For him “if you are not on the AppStore, you are nowhere”. He suggests that web apps could target niche consumers who would not mind searching web apps. The difference in consumer pools is reflected by another interviewee, who considers that consumers are used to native apps and it will be hard to educate them to use web apps because there are few “usage gateways”.

**Monetization support.** There are different revenue models to generate an income from application development that can be used individually or in combination, such as pay per download (e.g., CutTheRope), pay per subscription (e.g., New York Times), advertising (e.g.,
Blendr), pay for features (e.g., Hipstamatic), and freemium (e.g., Angry Birds). Application portals and platform providers have now provided support for most of these business models for native apps and hybrid apps. More precisely they have made these models easy through one-click operations and access to portals from the device. Web apps on the other hand seem to still lack a proven monetization business models and still do not provide on click purchase support (Claburn 2011). Mahemoff (2011) and several experts interviewed acknowledge this issue, but argue that this can change in the future, as monetizing the web has always been possible.

**Distribution constraints.** A number of portals impose restrictions on the types of apps they support. As web apps can easily be deployed on an independent portal they do not suffer from such restrictions. Conversely, native apps and hybrid apps, and especially iPhone apps are penalized by such restrictions. This issue was reflected in our interviews. Furthermore, this support generally includes a 30% fee for the portal provider. This is seen as a major drawback for many developers. Web app portals have the advantage of not yet being restricted by the 30% fee. For example Google’s Chrome Web Store takes a fee between 0%-5% (Claburn 2011). One of our interviewees says the 30% distribution fee is one of the native app drawbacks that could lead people to move towards web apps. He contends that his company “is not ready to give Apple 30% of its revenue as it would take away all its profits”.

### 3.3 Usage criteria

When it comes specifying the requirements of an application there are two main differences that are mentioned in the literature and arose in our interviews: performance and look-and-feel.

**Performance.** With the advancement of hardware and bandwidth it is possible for mobile phone to reach unprecedented levels of performance. In general, performance of the compiled native code is faster than the interpreted web language used in web apps. Even though Javascript is becoming faster and faster it still lags behind native code (Mahemoff 2011, Charland and Leroux 2011, Claburn 2011). Hybrid apps have an in-between performance, with a possibly fast running native component and a slower running web component. One of our interviewees notes that the low performance in image rendering is an important concern for web apps. More confident, another interviewee expects that web apps will soon yield performances that can almost match those of native apps.

**Look-and-Feel.** An important aspect of the user experience on a mobile platform is the coherence of the user interface, i.e., the look-and-feel (Lai et al. 2002). Platforms provide guidelines in order to harmonize the look-and-feel across native applications. However, with web apps, there is no requirement to follow platform standards. As a consequence, web apps often lack the user-friendly look and feel native apps provide (Oehlman and Blanc 2011). Furthermore, in terms of interface, native apps have a richer set of controls and experiences (Charland an Leroux 2011). Mahemoff (2011) argues that web apps can still be customized to suit platform look-and-feel, and that look-and-feel for some applications, such as games, do not need to follow platform standards. Kautz et al. (2007) argue that web apps have the advantage of providing a unified and standard look and feel across platforms. One of our interviewees argues that the ease of use that qualifies native apps is a barrier for other types of apps. Hybrid apps are in between native and web apps in this respect.
4. A multi-criteria evaluation of mobile app development strategies

In Table 1, we observe that whenever native apps score well, web apps score badly and whenever web apps score well, native apps score badly. For the nine criteria presented, native apps have a positive score on six and a negative score on three. Web apps have the opposite results and score well for portability, maintenance, and distribution constraints. Hybrid apps score three positive results (technical specifications, distribution support and monetization support), five neutral results (development support, portability, maintenance, performance and look-and-feel) and one negative result (distribution constraints). Interestingly, hybrid apps seem to be able to mitigate negative scores while they are still capable to obtain several positive results. However, they are not able to follow a true “best of both worlds” performance. As none of the alternatives clearly dominates, the choice would mainly depend on the type of mobile application to be developed. At this stage, our analysis only describes the overall performance of the mobile application development strategies in light of the criteria. We do not take into account the nature of specific applications, which require a particular attention on certain criteria.

<table>
<thead>
<tr>
<th>Features</th>
<th>Web apps</th>
<th>Hybrid apps</th>
<th>Native apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development tools</td>
<td>- regular support</td>
<td>+ advanced support (native)</td>
<td>+ advanced support</td>
</tr>
<tr>
<td>Portability</td>
<td>+ portable</td>
<td>+ portable (web)</td>
<td>- not portable</td>
</tr>
<tr>
<td>Maintenance</td>
<td>+ unrestricted</td>
<td>+ unrestricted web update</td>
<td>- possibly restricted</td>
</tr>
<tr>
<td>Access to consumers</td>
<td>- uneducated consumers</td>
<td>+ discoverability</td>
<td>+ discoverability</td>
</tr>
<tr>
<td>Monetization support</td>
<td>- no integrated billing</td>
<td>+ billing through portal</td>
<td>+ billing through portal</td>
</tr>
<tr>
<td>Distribution constraints</td>
<td>+ free (self-hosting)</td>
<td>+ revenue sharing with portal</td>
<td>+ revenue sharing with portal</td>
</tr>
<tr>
<td>Performance</td>
<td>+ slow</td>
<td>- slow web component</td>
<td>+ fast</td>
</tr>
<tr>
<td>Look&amp;Feel</td>
<td>- inconsistent</td>
<td>-/+ partly standardized</td>
<td>+ consistent</td>
</tr>
</tbody>
</table>

Table 1. Summary of the evaluation of each development strategies

5. Conclusion and further research

This short research-in-progress paper is the first step of a larger research effort to better comprehend the development of mobile applications. As the phenomenon is emerging and evolving fast due to the rapid pace of innovation, the evaluations can change according to the technology improvements. However, the established criteria constitute a rather stable framework that can be reused iteratively. The decision model (i.e., criteria and alternatives) proposed is an essential step in the design process of the DSS. The next step is to select a group of relevant experts working in different industries in order to complete our model with their preferences. The objective is to complete the model by adding weights (i.e., relative importance) for each criterion. These weights could depend on the type of application developed (e.g., games, entertainment, social, news, weather, travel, ...). After establishing the weights, we could run a simple multi-criteria method (e.g., weighted sum model) or a more sophisticated technique (e.g., AHP) to compute a ranking of the alternatives. The model obtained will be implemented into the DSS to help companies and developers to take better decisions concerning their mobile development strategies in the future.
Reference list