

Application development on mobile platforms

Do we need “Smartphone 2.0”?

Introduction

A recent census of 100 software developers strongly suggests that the software development fraternity feel “mobile” is a trending area. This is not a big surprise given the exponential growth of the smartphone market. According to Gartner (2011) sales of mobile devices in general grew by 5.6% in Q3 2011, while sales of smartphones specifically, grew by 42% during the same period. Total smartphone sales of 1 billion devices are predicted for 2013 (Deloitte, 2012). Here I present a research report incorporating an analysis of relevant literature and leading to a discussion of the barriers, challenges and opportunities related to developing software for mobile platforms.

What do you think is the most important current trend in software development?

The census concerned is small-scale, and general in nature. Developers were asked approximately 200 questions about many aspects of their work, workplaces and working practices. In this report I focus *only* on the responses to the question “What do you think is the most important current trend in software development?” I conducted a thematic coding analysis of the data. The data are clearly indentifying “mobile” as the most significant trending area. All responses to the question are included verbatim as an appendix.

Notes on the analysis, coding, and methodology

All null responses to the question were disregarded. The coding scheme was developed in two stage process. Firstly all the data was scanned subjectively to ‘get to know the data’, and I identified a small number of recurrent themes. I defined themes, and assigned survey responses to them according to keyword usage within each response. If a keyword appeared in a survey response then that response was assigned to the relevant theme(s). Individual responses may be included in multiple themes (for instance “Mobile and the web” would be counted in both relevant themes. All non-null responses, that did not match any theme are included in the table below as ‘Not coded’.

Results

Frequency	Theme	Keywords
17 (38%)	Mobile	<i>Mobile, Mobility, Mobile Device, BlackBerry, Mobile Apps, Smartphones</i>
7 (16%)	Web development	<i>Web design, Web development, Web, HTML, CSS</i>
9 (20%)	Distributed systems	<i>Cloud, Cloud computing, Software as a Service, Distributed Systems,</i>
4 (9%)	Data processing	<i>Concurrency, Multi-core, Big Data</i>
2 (4%)	Open source	<i>Open source</i>

2 (4%)	Object orientated programming	<i>OOP, OO</i>
12 (27%)	Not coded	N/A

Note that the percentages represent frequencies of coding, not total number of responses.

What does 'mobile' mean?

One of the key attractions of a smartphone when compared to a feature phone is its ability to run rich applications. I assume that the growth in smartphone adoption has led to a surge in smartphone software development, facilitated by the online software marketplaces associated with mobile operating systems. The two leading marketplaces are the Apple App Store and Google Play Store. Apple's App Store, launched in 2008, passed the 25 billion download mark in March 2012, and hit 30 billion only 3 months later (Business Insider, 2012). This report makes the assumption that the software developers that cited 'mobile' as a growth area were generally speaking, referring to development on smartphone platforms.

Literature Review

Cross-platform tools, web, or native?

For the most part software developers will always be looking to have the largest possible target audience, as such will need to develop software for multiple (usually incompatible) platforms. There are different approaches to tackling this problem: cross-platform development toolkits (XDTs); browser/web based software; creating multiple version of software using native SDKs.

There are pros and cons to each approach, it seems that one of the most reliable approaches to ensure a good user experience is to develop using the official SDK, however this requires a multi-skilled development team, and will dramatically increase development costs (not to mention annoying users, if one platform has an apparently better application). There are a plethora of tools available for XDT development including LiveCode, Marmalade and Phone Gap. Each different approach is subtly different, however a common theme is that none of the XDTs allow as much functionality as using the official SDK, or introduce other problems, Julian Ohrt and Volker Turau (2012) have produced a comprehensive review of the functionality of XDTs.

The inability for any XDT to truly allow a developer to maintain a single code base for deployment across many platforms, let alone produce an app that utilises all of the native SDKs UI finesse, is closely tied to the different restrictions that hardware manufacturers place upon developers. For instance Apple only allow access to their Bluetooth API from other Apple devices, whereas Windows Phone 7 doesn't have any Bluetooth API at all! XDT development has to take all of this into account, and usually arrives at either a relatively unsatisfying XDT application, often causing developers to return to expensive and hard to maintain native solutions, or only develop for one platform at a time. Neither are ideal for developers. A natural solution to this cross-platform quandary is a move toward web-based applications, accessible to *any* smartphone via its browser, and easily updated at the backend. Increasingly, advanced features of phones are being exposed to web browsers (for example exposing GPS or accelerometer data) allowing for richer application development. HTML5 and related advances in web technologies, are making web-apps for smartphone an increasingly viable alternative

to native apps. Despite these steps forward, the inconsistencies between platforms, lack of access to local file systems, inability to run truly multi-threaded code, and most importantly the fact that native apps have unrivaled access to UI features, any developer who wants to deliver the best user experience they can will still have to focus on native apps (Charland, A. & Leroux, B., 2011).

Interestingly researchers at Microsoft Research and the University of Washington identified a similarly fragmented ecosystems for home automation systems. Although accepting that heterogeneity of systems and lack of standardisation in home automation are stifling innovation in the area Dixon, C et al (2010) propose *another* – open and adaptable – development platform as the solution.

HTML5 based applications are becoming increasingly rich and indistinguishable from native applications, even given that HTML5 is only at the early stages of adoption (Griffiths; Ogden; Aspin, 2012). Facebook – once champions of HTML5 - recently abandoned their HTML5-centred application in favor of a fully native iOS system, explaining this the iOS product manager for facebook, Mick Johnson, said “embedding HTML5 inside an app isn’t what people expect” (The Verge, 2012). The developers of the Sencha framework reacted to this by showcasing a faster, better featured version of Facebook’s Android and iOS offerings that was made entirely in HTML5 (Vimeo.com, 2012). This suggests that browser-based applications can, at least *theoretically*, keep up with native alternatives.

Proponents of HTML5 being the future of mobile applications include Oracle’s Cameron Purdy (eWeek, 2012) and – as far back as 2009 - Google’s Vic Gundotra (Financial Times, 2009). But as one technology blogger put it “Native technology allows for excellent apps, whereas HTML5 apps will be at best just OK” (Mobtest.com, 2012).

In summary: the best UI features are only reliably available if using the native SDK. XDTs allow developers to move towards a single codebase, but no XDT can provide all of the features that developers (or users) want. HTML5 is able to deliver native-like performance and features, but hardware manufacturers stifle this by restricting access to APIs.

What about “bad” software?

Although early, pre-iPhone, smartphones were affected by malware to an extent (Schmidt, A.-D. et al., 2009) so far the new generation of smartphones have been mostly unaffected by malicious developers. However as smartphone adoption grows, one would expect so will the volume and range of malware written for smartphones. Felt, A.P. et al (2011) clearly show that there is a growing trend towards malware written for smartphones, and a portfolio of different approaches.

Browsers, but beyond JavaScript?

On desktop machines Google’s Native Client, a sandbox for running x86 native code inside a web browser (Yee, B. et al., 2010) could be the sort of technology that starts to allow developers to take advantage of native-like functionality (and local processing power) directly from a web browser. Other APIs such as WebGL expose the power harnessed in GPUs to web developers (Golubovic, D. et al., 2011). WebGL is already accessible to mobile developers, although no system similar to Native Client current exists for smartphones.

Everybody works the same?

Smartphones, and their ability to run rich applications, are used in incredibly diverse ways and contexts (Soikkeli, T., Karikoski, J. & Hammainen, H., 2011). The extreme levels of diversity in how

smartphones are used has an impact on the concerns of mobile developers, in particular for strategies to allow customisation of how applications impact upon power drain (Falaki, H, 2010). One would assume that as adoption grows, particularly in developing nations, that this diversity will only continue to grow.

Android and Apple, monopolising the smartphone OS market?

There is limited research into the marketplace for mobile operating systems (Lin, F; Weiguo Y, 2009) but contrary to an iOS monopoly - oft-predicted as recently as 2010 (Networkworld, 2010) - Apple's dominance has waned in recent years (eWeek, 2011). This is true globally at least; Apple still have a majority market share in the United States. Apple's relative loss of market share, when considered alongside ever growing assortment of Android forks (such as the Kindle Fire and the NOOK) and most significantly the announcement of entirely new smartphone platforms such as Tizen and FirefoxOS, suggest that the platforms on which smartphone development takes place is in fact diversifying, rather than moving towards a monopoly, or perhaps duopoly.

Standardisation, Innovation, or both?

There was a considerable increase in research around standardisation and innovation between 1995 and 2008 (Choi, D.G., Lee, H. & Sung, T., 2011), although there is no significant research around how it impacts software development for smartphones specifically. From what literature that exists, two general points of view emerge, firstly that standardisation may bring about 'competitive crowding' (Boudreau, K.J., 2011) and thus stifle innovation. Alternatively it can be argued that when influential players adopt a standard, it may have a profound affect on *other* players. Although written in a time before smartphones and the web, Farrell & Saloner's work on standardisation (1985) concludes that "a large firm's customers experience relatively little change in their payoff when other firms decide whether to be compatible with the large firm. By contrast, the large firm's decision substantially affects the payoffs to buyers of other firms' products".

Discussion

Smartphone ownership is becoming more and more ubiquitous, smartphone application development grows with it. Smartphone manufacturers are engaged in a race to get the most market share. This collective scramble for market share has meant that hardware and software producers innovate at an astounding rate. The pace of development means that it hasn't been possible for any consensus to emerge organically addressing what is the 'best' way to develop software for smartphone platforms.

New operating systems and approaches to development seem inevitable, hardware manufacturers are keen to protect their businesses by ensuring that they're not bound to a sole operating system provider (the exception is Apple, who produce hardware and software themselves). As a result of this diversification, it is likely that software developers will be asked to produce software for an increasing number of smartphone platforms.

Several questions are raised about how developers, manufacturers, and even consumers, might want to synthesise some of the issues raised in the report over the coming years to inform a cohesive path forwards:

1. How will a growing, but increasingly divided smartphone OS market, effect developers?

2. Should hardware manufacturers be willing to move towards universal browser-based applications?
 - i. How can smartphone OS providers protect revenue streams coming from software marketplaces, assuming browser based applications become prevalent?
 - ii. Can allowing developers the ability to run native code inside browsers make HTML5 apps more viable?
 - iii. Will security and anti-malware measures become as important for mobile as they are for desktop systems, and can a move toward browser based applications have an impact on this?
3. In terms of efficient software development for smartphone platforms, does *how* users use devices become as important as *what* (application) they're using?
4. Can standardisation of development platforms help stimulate innovation in smartphone development?

In addition, wider developments on the internet, web, and with physical computing may well add a whole new set of questions, all of which deserve attention from the smartphone developers' perspective. For instance how will the semantic web and linked data applications (Bizer, C., Heath, T. & Berners-Lee, T., 2009) have an impact on standardisation in smartphone development? Can XDTs or browser-based apps for mobile platforms be used by manufacturers of equipment for the 'internet of things'? (Atzori, L., Iera, A. & Morabito, G., 2010)

Conclusion

The questions above characterise the barriers, challenges and opportunities for developers of smartphone software covering technological issues (including the infancy of HTML5 and XDTs) and systemic problems stemming from the interests of an extremely powerful few (Apple, Google, and other hardware manufacturers). Although the exponential growth in the area, increasing ubiquity of smartphones, and the tangible changes in societal behavior are evidence that the smartphone ecosystem is 'working', this report exposes clear opportunities to develop more cohesive, more powerful, safer, and more efficient frameworks for smartphone software developers.

This report began with a tentative census of developers and reviews literature related to the area that the majority developers think is 'the most important trend' in their industry. By researching the nature of what 'Smartphone 2.0' might look like, and different paths by which to get there, we can assure that the software development community have the tools they need to deliver on the demand that consumers provide, in the most efficient way possible.

References

- Atzori, L., Iera, A. & Morabito, G., 2010. The Internet of Things: A survey. *Computer Networks*, 54(15), pp.2787–2805. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S1389128610001568> [Accessed November 1, 2012].
- Bizer, C., Heath, T. & Berners-Lee, T., 2009. Linked data-the story so far. *International Journal on Semantic ...*. Available at: <http://www.igi-global.com/article/linked-data-story-far/37496> [Accessed January 23, 2013].

Boudreau, K.J., 2011. Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation. *Organization Science*, 23(5), pp.1409–1427. Available at: <http://orgsci.journal.informs.org/cgi/doi/10.1287/orsc.1110.0678>.

Businessinsider.com, 2012. More Than 30 Billion Apps Have Been Downloaded In The App Store. *Business Insider*. Available at: <http://www.businessinsider.com/more-than-30-billion-apps-have-been-downloaded-in-the-app-store-2012-8> [Accessed January 19, 2013].

Charland, A. & Leroux, B., 2011. Mobile application development: web vs. native. *Communications of the ACM*, 54, pp.49–53. Available at: <http://dl.acm.org/citation.cfm?id=1941504> [Accessed January 20, 2013].

Choi, D.G., Lee, H. & Sung, T., 2011. Research profiling for “standardization and innovation”. *Scientometrics*, 88(1), pp.259–278. Available at: <http://www.springerlink.com/index/10.1007/s11192-011-0344-7> [Accessed November 27, 2012].

Deloitte, 2012. What’s ahead in technology, media & telecommunications. Available at: http://www.deloitte.com/view/en_GX/global/industries/technology-media-telecommunications/tmt-predictions-2013/index.htm [Accessed January 19, 2013].

Dixon, C. et al., 2010. The home needs an operating system (and an app store). *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks - Hotnets ’10*, pp.1–6. Available at: <http://portal.acm.org/citation.cfm?doid=1868447.1868465>.

EWeek.com, 2012. HTML5, Cloud and Mobile Create “Perfect Storm” for Major App Dev Shift - Application Development - News & Reviews - eWeek.com. *eWeek.com*. Available at: <http://www.eweek.com/c/a/Application-Development/HTML5-Cloud-and-Mobile-Creat-Perfect-Storm-for-Major-App-Dev-Shift-235285/> [Accessed January 20, 2013].

Falaki, H. et al., 2010. Diversity in smartphone usage. *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys ’10*, p.179. Available at: <http://portal.acm.org/citation.cfm?doid=1814433.1814453>.

Farrell, J. & Saloner, G., 1985. Standardization, compatibility, and innovation. *The RAND Journal of Economics*, 16(1), pp.70–83. Available at: <http://www.jstor.org/stable/10.2307/2555589> [Accessed January 23, 2013].

Felt, A.P. et al., 2011. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices - SPSM ’11*. New York, New York, USA: ACM Press, p. 3. Available at: <http://dl.acm.org/citation.cfm?doid=2046614.2046618> [Accessed October 31, 2012].

Gartner.com, 2011. Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent. *gartner.com*. Available at: <http://www.gartner.com/newsroom/id/1848514> [Accessed January 20, 2013].

Golubovic, D. et al., 2011. WebGL implementation in WebKit based web browser on Android platform. In *2011 19th Telecommunications Forum (TELFOR) Proceedings of Papers*. IEEE, pp. 1139–1142. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6143751> [Accessed January 21, 2013].

Griffiths, L; Ogden, R; Aspin, R., 2012. A profile of the future: what could HTML 5 do for HE by 2015? *Research in Learning Technology*, 20, p.1. Available at: <http://www.researchinlearningtechnology.net/index.php/rlt/article/view/19199/html>.

Lin, F. & Ye, W., 2009. Operating System Battle in the Ecosystem of Smartphone Industry. In 2009 International Symposium on Information Engineering and Electronic Commerce. IEEE, pp. 617–621. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5175193> [Accessed November 16, 2012].

Mobtest.com, 2012. Web Developers, Don't be Lazy and Learn a Native Language. mobtest.com. Available at: <http://blog.mobtest.com/2012/12/web-developers-dont-be-lazy-and-learn-a-native-mobile-language/> [Accessed January 23, 2013].

Networkworld.com, 2010. iPhone vs. Droid vs. BlackBerry, The iPhone gets all the attention, but don't overlook its competitors. Available at: <http://www.networkworld.com/news/2010/060710-tech-argument-iphone-droid-blackberry.html> [Accessed January 19, 2013].

Ohr, J. & Turau, V., 2012. Cross-Platform Development Tools for Smartphone Applications. Computer, 45(9), pp.72–79. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6178193> [Accessed October 29, 2012].

Schmidt, A.-D. et al., 2009. Smartphone malware evolution revisited: Android next target? In 2009 4th International Conference on Malicious and Unwanted Software (MALWARE). IEEE, pp. 1–7. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5403026 [Accessed November 8, 2012].

Theverge.com, 2012. Facebook for iOS goes native, waves goodbye to HTML 5. theverge.com. Available at: <http://www.theverge.com/2012/8/23/3262782/facebook-for-ios-native-app> [Accessed January 15, 2013].

Times, F., 2009. App stores are not the future, says Google | Tech blog. Financial Times. Available at: <http://blogs.ft.com/tech-blog/2009/07/app-stores-are-not-the-future-says-google/> [Accessed January 22, 2013].

Vimeo.com, 2012. Sencha Fastbook, Available at: <http://vimeo.com/55486684>.

Yee, B. et al., 2010. Native Client. Communications of the ACM, 53(1), p.91. Available at: <http://portal.acm.org/citation.cfm?doid=1629175.1629203> [Accessed January 6, 2013].

Bibliography

Robson, C., 2011. Real World Research (3rd Edition), Wiley.

Appendix

Below are all of the responses to the question “What do you think is the most important current trend in software development?” from the developer census.

automatised and standardised testing
Big Data.
Cloud computing
cloud computing ...

cloud-computing. why? because SOA is gone and the next trend is not there yet.
Competition in the industry
Copy others.
Crowd Sourcing and Agile developments. Because They both resolve problems we see with un-interested developers. You cannot be un-interested when using either of these methods as it shows up really well on daily reports.
Development of [highly] concurrent systems. Why? Advent of multiple-cores and compute intensive data analytics.
functional/map reduce languages eg hadoop as opposed to OO.. because it will be the only way to scale across many cores, multi machines/big data
I would say it's the shift from desktop solutions to mobile solutions. Any corporation that does not invest considerable resources in Mobility is going to face a lot of problems in the near future.
I am also wary of people and organizations jumping on he cloud-computing bandwagon, because I see inherent risks when all or most software resources reside outside the host system. Unreliable Internet connectivity, especially in developing economies, just erodes the benefits offered by Cloud.
in app purchasing, free to download
In my team, even though our main job is development, we also do a major amount of testing. This is important so that we as developers can fix bugs as and when we develop instead of waiting for another testing team to find bugs and then make changes at a lter stage.
It seems to me that stakeholders require new functionality constantly. It is no longer good enough to work on a project for six months and return after six months with a deliverable. Instead, new functionality is required every few weeks.
To me, this mrrors our increasing need for news updates and our appetite for gobbling up bite sized chunks of news on the internet. We want updates fast and we want them all the time.
Sometimes it is more rewarding to be patient.
Migration from Legacy Systems to new technologies and Maintenance of existing projects.
Mobile / Big Data. Empowering end-users whilst improving understanding
Mobile and cloud computing, as many people are likely to have their own smart phone mobile.
Mobile application. Due to the rise of smart phones and thier complexity in structure, hardware and features.
Mobile applications.
The hardware has become very popular the last years.

mobile apps
Mobile development, and the way large platform companies are starting to invest in developers for their platform. BlackBerry as an example, offering incentives for app store launches, developer co-working spaces, conferences, free devices etc.
Tablet sales are high, Smartphone sales are high, market penetration is high.. Not only is it where the money is at in terms of app sales, companies without apps are losing out. In 5 years maybe we will see lots of JavaScript and Mobile developers where web developers once sat
Mobile device, because everyone use it
Mobile.
Mobile. Distributed. Semantic.
Gut feeling.
Mobility and Cloud.
Move to more focused apps as opposed to general systems.
Move towards concurrency, multicore and distributed systems (again).
multilingual education. times where you can learn one language and get by are over.
Need for catching up with new technology
Not sure I can identify a single important trend. Some of the trends that I'm very interested in pursuing (and many of these are linked) are: test-driven development, continuous integration, patterns-based programming, separation of concerns (e.g. MVC, MVM models).
OOP... It is the future
Open source development
Open standards and commoditisation in user platforms
Opensource projects - Mobile applications
providing software as a service for cloud computing
Scalability is becoming more important to entry-level developers. Mobile apps are everywhere, web apps now have to deal with potentially hundreds of thousands of concurrent users due to the growth of app stores.
The most important trend is the expert in one way of code such as the software needs C language to develop the major features, the trend will be used to find a developer to develop the quality software.
The move to cloud deployment.
UI Simplicity. Things appear to be going back to the start and its all about getting the intended job done rather than all fanciness which became a trend over the last 5/6 years.
Using the precise technology so as to enhance business processes e.g. SAP

Web based applications freeing software from the constrain of narrow platforms and adding mobililty

Web based, cross platform compatibility

Web technologies on mobile devices

Web.

Most (if not all) systems are currently being migrated to the Web (or to the cloud).

With regards to web design/development, the most important trend we have been practising and implementing is responsive web design using HTML5/CSS3.. I find this the most important trend right now due to it's usability across all platforms mobile to desktp computer, with the number of people browsing the web on mobile always increasing it's essential to have a website that functions really well across all devices.