

Evaluating the Mobile Operating System Effectively

Edward Bramanti

November 5, 2013

Abstract

1 Introduction

In today's world, mobile operating systems are now as prevalent as personal computers. More and more consumers are turning to smartphones and tablets to daily drive their time spent in productivity as well as entertainment. As this field has continued to develop, the consumer is given a myriad of choices between many different operating systems. Hardware and software manufacturers have arisen to provide a competitive environment. This has led to successes for user choice, as hardware and software companies square off to provide the most innovative interface.

While this has been a great success for the consumer, it has also caused problems for firms. A series of patent wars between companies has arisen in the past few years, which has demonstrated the desire for monopoly over the market by utilizing the legal system. Paik and Zhu note that "According to this view, technology firms race to assemble patent portfolios - initially for defensive purposes in the context of a dynamic and competitive field - but then, as the industry matures, convert their shields into weapons to eliminate their competitors in pursuit of market dominance with its platform." [PZ13] This desire to achieve market independence has led many reviewers to create preconceived notions that the market is no longer innovating, but instead creating a "future-proofing" system that fights pointless legal battles to control software innovation.

Because of this new growth in the smartphone market, reviewers are pitting operating system against operating system to help the user discover if it is the best fit for them. The question remains though: Are the journalistic and firm-based reviewers doing an accurate job of evaluating a mobile operating system? In particular, we will be taking a look at Pfeiffer Consulting's "Mobile OS User Experience Shootout" as a reference for how companies conduct evaluations of mobile operating systems today, and how focusing on certain things that connect consumers and developers together through mental process will increase reasoning and understanding behind the results collected from users.

2 Background

To provide background on how evaluations of a mobile operating system should be performed, we will look at the data from the two most important points of view: first, the developer; and then, the consumer, or end-user. Seeing how the developer analyzes the operating system versus the end-user will help to provide a bridge between the mental thought process of the two, and will provide those observing a foundation for how to approach reviewing a mobile operating system effectively.

2.1 The Developer Perspective

Why is the developer's perspective important to evaluating a mobile operating system? Developers are essential to any platform building its ecosystem, and a mobile operating

system is no exception. In addition, understanding the developer's thought process will be invaluable to hearing the other side of the story, since some evaluations of mobile OS' occur only through the eyes of the end user (Pfeiffer's report is an example).

So how do developers evaluate mobile operating systems, and choose which is the best to design for? Palme et al. state that there are six dimensions to how a developer decides on the best mobile operating system for them: "corporate buyer choice, consumer buyer choice, OS vendor's market growth potential, ease of implementation, security, and revenue." [PTSP10] Specifically, ease of implementation is one factor to focus on in determining if a mobile OS is successful. Evaluators generally overlook this, because they focus on the ease of use for consumer instead of the ease of implementation for the developer. Besides, the end-user never sees the developer's thought process, right? This is a common misconception amongst reviewers of mobile OS functionality. The developer's job is to mesh his mental model of how something should work with the user's model. Interaction design itself is defined this way: "A system's designer/developer must effectively communicate his mental model of the system to the system's users through the "image" presented by that system." Therefore, if the "terms and conditions [of the mobile Software Development Kit(SDK)]... limit[s] the freedom of development by restricting the domain of application or variety of functionality," the ease of use for the user suffers. [PTSP10]



Figure 1: This picture portrays the mental model of the developer choosing a mobile OS to design an app for. [PTSP10]

It is important to realize the designer's role in the evaluation of the success of an operating

system. Since the ease of implementation is heavily tied to the ease of use for the user, it would be an incomplete analysis of operating system success without considering the developer’s perspective on the functionality of the OS at the software development level.

2.2 The End-User Perspective

When many go to evaluate and review mobile operating systems, they run unknowingly into the issue of context. During a controlled evaluation, many users may respond in a certain way to an interface. Some may respond in certain ways when placed in certain situations. But at the end of the day, what is the goal of the end-user? The end-user’s experience needs to be evaluated contextually, according to Coutaz et al. The end-user value a contextual experience for their experience of an OS on the go. [CCDG05]

So, why is context a part of the end-user perspective that so many seem to gloss over? It’s very difficult to gauge an experience contextually. Apps like Aviate for Android are bringing validity to the importance of contextual computing, and are showing a valid area that is being overlooked by reviewers: what users do at certain moments in certain locations. For example, Aviate considers if you are at the coffee shop by pulling your location from GPS services, and displays a new context on the launch screen with things specifically tailored to your current experience. [Bry] These kind of things matter to the end-user on a mobile operating system. Since the location of a mobile OS is not statically located like desktop operating systems, users will consider context to be essential to the experience, as can be seen in Figure 2. Evaluations of mobile OS’ search for this key term,

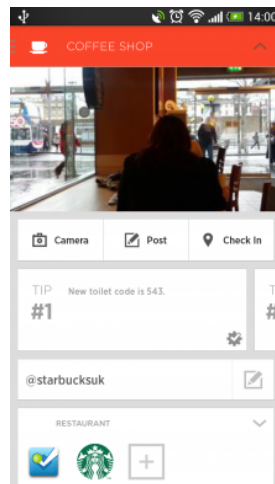


Figure 2: Aviate is a contextual launcher application for Android. Pictured here is the mobile OS detecting the current context of the phone, and modifying its launch screen to show what the user needs in this certain situation. [Bry]

3 Methods

Now that we have filled in the background of developer and end-user perspective with past research, we can now effectively evaluate Pfeiffer Consulting’s Mobile OS evaluation. Pfeiffer evaluates four ”usability metrics” in a way that they describe as a ”look at aspects that have a direct impact on the day-to-day user experience of an average, non-technical user.” [Con13] The four metrics they are analyzing will be evaluated as compared to the background found in Section 2.

3.1 Cognitive Load Evaluation

Pfeiffer takes a survey over a term they have coined ”cognitive load,” which ”counted the number of apps/widgets as well as other icons and user interface elements a default installation of the operating system contains.” [Con13] Pfeiffer uses this in their argument that the lower the cognitive load, the easier it will be for a user to navigate the operating system, at the cost of features. They define this as ”the number of apps/widgets as well as other icons and user interface elements a default installation of the operating system contains.” [Con13] The first problem that this survey runs into, when compared to background, is how contextual computing can nullify a cognitive load test. If the end-user has an interface that is contextual (in the case of this survey, say they are using Aviate on their Samsung Android device) then the extra apps are not a hindrance, but represent a low cognitive load. The apps you need only appear based on context, which makes this survey a little difficult to quantify.

In addition to this, a lower cognitive load proves to not necessarily be better or worse for the OS. Even though iOS 7 had a higher cognitive load than iOS 6, it still scored higher overall. Pfeiffer Consulting’s study shows that cognitive load has no correlation with ease of use for the user, which invalidates this part of the study (which is supposed to be evaluative).

3.2 Efficiency and Integration Evaluation

Pfeiffer’s second part of their Mobile OS evaluation is a study of ”efficiency” and ”integration,” mapping out the user’s access to key points of the operating system on a rating scale from 0 to 10. From a usability metrics measurement of efficiency, it inaccurately follows interaction design principles. According to usability consultant Jakob Nielsen, ”efficiency is based on speed of performance once one has learned the tasks.” [Nie93] Since this is normally measured in time, the evaluation shows a lack of detail that should not be omitted. Since efficiency is measured in time, representing the time to access certain features would be more revealing. For example, some mobile operating systems are criticized for lacking features others have, but it does not necessarily mean that the implementations that are made have a given benefit of efficiency.

IOS 7	7/10	IOS 6	6/10	Android (Samsung) 7/10	Windows Phone 8 4/10	Blackberry 10	5/10
Efficiency and Integration Options							
• Notification area (customizable)		• Notification area (customizable)		• Notification area			• Notification area (customizable)
• Notifications on Lock screen		• Notifications on Lock screen		• Notifications on Lock screen	• Notifications on Lock screen		• Notifications on Lock screen
• Multitasking		• Multitasking		• Multitasking	• Multitasking		• Multitasking
				• Multi-window mode			
• Control area				• Control Area			• Control area
• Multitouch controls		• Multitouch controls					
• Direct Camera Access from Lock screen		• Direct Camera Access from Lock screen					• Direct Camera Access from Lock screen

Figure 3: Pictured is the evaluation of efficiency and integration from the OS Experience Shootout. Notice the lack of concrete numbers for efficiency, but instead just a list of features that is assigned an arbitrary score. Efficiency is the time it takes to do tasks, not necessarily how feature rich a platform is. [Con13]

3.3 Customization Evaluated

The third part of the study dives deep into evaluating the customization merits of the chosen mobile operating systems. Out of all the sections of the evaluative study, according to background information this seems to be the most reliable of the studies. Customization depends on a direct comparison of the given operating system features, and even though measured on a 10 point scale once again, the scores are clearly outlined with a "why." Customization falls under satisfaction, which is why a 10 point scale is a strongly supported metric to use when evaluating customization. Users who are more satisfied with customization should consider an Android phone. Pfeiffer does not explain what low customization means for the user in their evaluation though. If customization is low, the ease of use for apps could become difficult for the user. As ease of use becomes difficult for the user, it can also be difficult to implement apps for developers, as accessing features may be restricted by a lack of customization in the mobile OS. Pfeiffer needs to more clearly present this in their evaluation, as the Background section connects ease of use for consumers with ease of implementation for developers.

3.4 User Experience Friction Evaluation

This final part of the study attempts to evaluate frustrations users have with mobile operating systems. From an interaction design standpoint, it does not study what Nielsen

refers to as "errors" in his usability metrics with a concise approach [Nie93]. Instead, this part of the study evaluates learnability; if something is hard to figure out, it gives it a high rating, and if easier, it receives a lower score. Therefore, this is not a study in errors as Pfeiffer outlines it to be. Instead, this list of user experience friction is a study in learnability, and which OS is easiest to learn overall. It also irrelevantly points out lacking features when it presents no standard for what features a smartphone should have. For example, the importance of usefulness in a situation (contextual computing) is not even mentioned in this part of the study. As users become more dependent on mobile operating systems to be contextual,

4 Discussion

Overall, I find Pfeiffer Consulting's evaluation of the current state of mobile operating systems to be confusing, incomplete, and lacking in detail. To begin, there is a lack of raw data presented in the study. Arbitrary values are assigned to metrics everywhere, and the unsuspecting reader will take these studies for granted. Pfeiffer does say that they want this to be a simple presentation of usability for average, non-technical users, but anyone interested in true evaluation will have the statistics to back up the data. Since Pfeiffer has not presented them here, it leaves me underwhelmed.

In addition, I do not believe the study to accurately reflect the true user experiences of the mobile operating systems. For example, utilizing efficiency metrics from Nielsen on iOS 7 shows slower efficiency as the animation transition time between apps has increased [Nie93]. If that is an important element of a study of usability, then iOS 7's score should reflect that in Pfeiffer's evaluation.

Finally, as mentioned in the background about a rise of importance in contextual computing, Pfeiffer's study ignores this as a metric altogether. As users require a device and operating system that will meet their needs in certain situations, the research proves that these types of things can't be ignored. One good example that was missed completely in this study is the contextually aware Google Now on Android 4.1 and above. Since Samsung phones support this software, it's a shame that Pfeiffer did not take this study further and see its importance to the user. Since context is key in a movable device according to Coutaz et al., Google Now nails this by providing traffic reports on your commute, nearby restaurants around lunch time, and birthday notifications for your closest friends. [Hod] I wholeheartedly agree with what the literature says on contextual computing: having apps you need, when you need them, is a core part of an usable interface in mobile computing. Therefore, a mobile OS that makes use of this should be evaluated more positively, simply because of the increase in efficiency to perform tasks and a reduction in learnability as your device presents what you need in the moments you need them [Nie93].

5 Conclusion

The Pfeiffer Consulting study is useful to this study to show a lack of thought and polish put into analyzing the benefits and drawbacks of mobile operating systems today. So many reviewers barely scratch the surface of true evaluation due to ambiguous statistics and opinions that seem like plausible facts. According to past studies, a proven model for a successful mobile operating system is based on an easy implementation for developers and a contextual experience for end-users. By contextual, we mean an experience that correlates with what Nielsen calls strong "usability": highly efficient in context and low learnability and memorability time due to contextual awareness. Reviews such as Pfeiffer's mobile OS evaluation cover some of the core usability elements, but not well. A clear understanding of interaction design and metrics is needed to conduct a study; the details may be simplified to be easier to read for consumers, but the data needs to back the figures up. From what Pfeiffer has provided, it demonstrates a lack of understanding of how to review a mobile OS; but, learning the importance of contextual computing and simplicity in development will determine the future of mobile OS usability.

References

- [Bry] Martin Bryant. Aviate: A gorgeous and smart new android homescreen built for the age of context. <http://thenextweb.com/apps/2013/10/15/aviate-a-smart-new-android-homescreen-built-for-the-age-of-context/>.
- [CCDG05] Joëlle Coutaz, James L. Crowley, Simon Dobson, and David Garlan. Context is key. *Commun. ACM*, 48(3):49–53, March 2005.
- [Con13] Pfeiffer Consulting. How ios 7 stacks up: Smartphone os user experience shootout. 2013.
- [Hod] Kelly Hodgkins. Google now brings contextual search and on-the-fly reminders to android 4.1. <http://www.intomobile.com/2012/06/27/google-now-brings-contextual-search-and-onthefly-reminders-android-41/>.
- [Nie93] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [PTSP10] Elia Palme, Chuan-Hoo Tan, Juliana Sutanto, and Chee Wei Phang. Choosing the smart phone operating system for developing mobile applications. In *Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business*, ICEC '10, pages 146–152, New York, NY, USA, 2010. ACM.

- [PZ13] Yongwook Paik and Feng Zhu. The impact of patent wars on firm strategy: Evidence from the global smartphone market. 2013.