ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (260-268), Month: July - September 2014, Available at: www.researchpublish.com

ANDROID IN OPPOSITION TO IPHONE

[1]Ms. A. Sivasankari, [2] Mrs. G. Sangeethalakshmi, [3]L. Sandhiya

[1,2,3]Department of Computer Science, D.K.M College for Women, Vellore, Tamil Nadu, India

Abstract: **Mobile devices are not in any way as they used to be in the past. Not only have the screens grown in size and quality, but also the internal hardware has grown to reach performance levels seen only in laptop computers some years ago. In addition to traditional mobile phones, the market has seen the rise of devices with screen up to over 10 inches, so called tablets. All of this opens doors for new, bigger, faster, better looking and possibly yet never seen applications to be developed. Google Android and Apple iOS being among the biggest players in the mobile operating system market, there is also a need for usable environments in which more or less experienced developers can create applications of their own for these specific environments. As the mobile application development is luring more and more developers into the market it has also become an attractive topic in educational environment. The growing popularity of Android and iOS has made them the two most interesting plat- forms for now. We compare Android and iOS users according to their demographic deference, security and privacy awareness, and reported behavior when installing apps. We present an exploratory study based on an online survey with more than students and describe directions for further research.**

***Keywords*: Smartphone; iOS; iPhone; Android; Personal Data; Security Awareness; Privacy Awareness.**

I. INTRODUCTION

Mobile devices are not in any way as they used to be in the past. Not only have the screens grown in size and quality, but also the internal hardware has grown to reach performance levels seen only in laptop computers some years ago. In addition to traditional mobile phones, the market has seen the rise of devices with screen up to over 10 inches, so called tablets. All of this opens doors for new, bigger, faster, better looking and possibly yet never seen applications to be developed.

Google Android and Apple iOS being among the biggest players in the mobile operat-ing system market, there is also a need for usable environments in which more or less experienced developers can create applications of their own for these specific environments. As the mobile application development is luring more and more developers into the market it has also become an attractive topic in educational environment. The growing popularity of Android and iOS has made them the two most interesting platforms for now.

The computer has been in constant evolution since the middle of the 20th century. Computers are continued to get smaller in size, using less power and performing more advanced calculations. In 2007 Apple released their iPhone to achieve the next goal in computing. This new type of communication tool, called Smartphone, is generally referred to as a phone, which is a poor labeling. A Smartphone is a handheld computer, which can place phone calls. Although the term Smartphone was first used in 1992, Apple was the first company to release a Smartphone to a wider audience. This evolution is led by computer manufacturers and software companies and not handset manufacturers, which have controlled the market thus far.

A. Objective

Google's Android and Apple's Iphone provides not only the mobile operating system but also provide a mobile development platform because of this they both are facing a tough competition against each other. The basic approach of the two is different but both have the power to win the user's heart. The objective of this paper is to compare the Android Software Development Kit and Apple's iOS Software Development Kit th each other. We will install both development

kits on workstations, create applications with them, rwiesearch the publishing process of the application and finally find out how much each environment would cost in educational use. All of these steps are documented and the documentation is presented in this paper.

B. Methodology

This Paper is carried out as a research project and is partly based on theoretical and partly on empirical research; the systems are first taken into use, and the results will then be documented. It is a research project comparing two development environments intended for similar use. Other source material will be gathered from related literature and from the companies responsible for the systems.

C. The Development Environment

Android SDK makes use of Java programming language, similar to Java Standard Edition (J2SE), called Java Android Library. This is an advantage to developers familiar with programming languages originating from the programming language family C. The syntax is the same as Java in terms of operands, selections, and iterations file handling and more. The more specific Android classes and packages use other names that are not similar to Java editions, such as the Activity Class and the View Class.

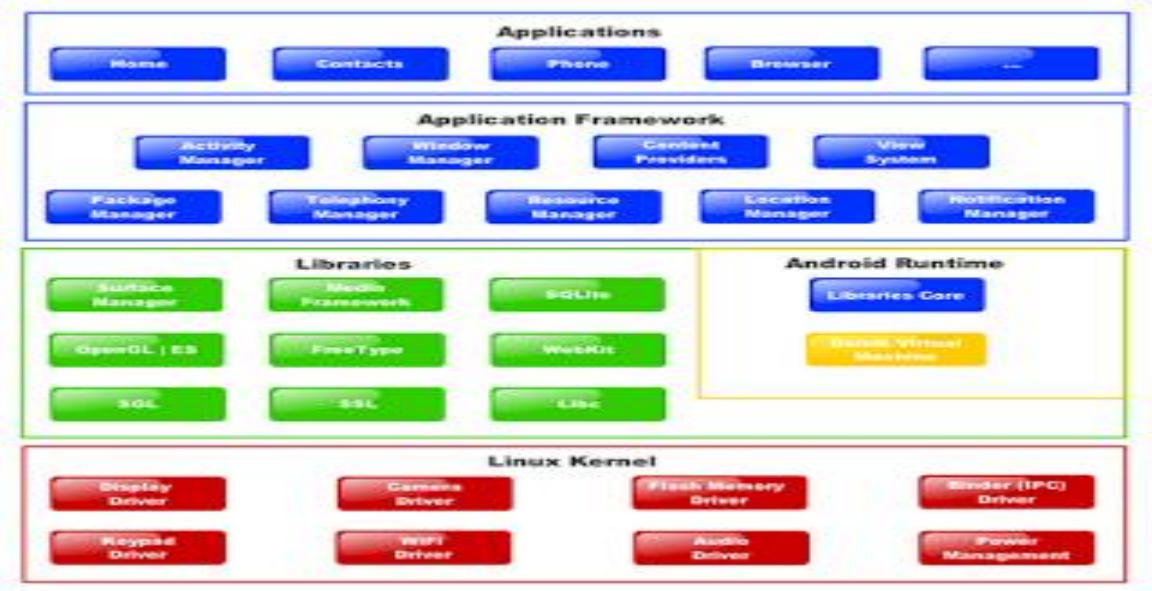

Fig 1: Android Architecture

Linux Kernel: Core services (including hardware drivers, process and memory management, security, network, and power management) are handled by a Linux 2.6 kernel. The kernel also provides an abstraction layer between the hardware and the remainder of the android architecture stack.

Libraries: Running on top of the kernel, Android includes various C/C++ core libraries such as lib and SSL, as well as the following:

- A media library for playback of audio and video media
- A surface manager to provide display management
- Graphics libraries that include SGL and OpenGL for 2D and 3D graphics
- SQLite for native database support
- SSL and Web Kit for integrated web browser and Internet security.

Android Run Time: The run time is what makes an Android phone an Android phone rather than a mobile Linux implementation. Including the core libraries and the Dalvik VM, the Android run time is the engine that powers your applications and, along with the libraries, forms the basis for the application framework.

Core Libraries: Although most Android application development is written using the Java language, Dalvik is not a Java VM. The core Android libraries provide most of the functionality available in the core Java libraries, as well as the Android-specific libraries.

Dalvik VM: Dalvik is a register-based Virtual Machine that's been optimized to ensure that a device can run multiple instances efficiently. It relies on the Linux kernel for threading and low-level memory management.
Application Framework: The application framework provides the classes used to create Android applications. It also provides a generic abstraction for hardware access and manages the user interface and application resources.

Application Layer: All applications, both native and third-party, are built on the application layer by means of the same API libraries. The application layer runs within the Android run time, using the classes and services made available from the application framework..

D. IOS

iPhone operating system is a mobile operating system developed and marketed by Apple Inc. It is the default operating system for the iPhone, the iTouch and the iPad. The iPhone OS was derived from Mac OX X and the version history of iPhone OS began at June 29, 2007 with the release of the iPhone. iPhone OS had no official name until the first beta version of the iPhone SDK released in March 6, 2008. Before then, Apple marketing literature simply stated the iPhone uses OS X. The initial version of iPhone was released on June 29, 2007. The version 1.0.2 was initially released on iPod Touch on September 14, 2007. Version 1.1.1 updated the interface for the calculator application, supported the TV out and provided adjustable volume and louder speakerphone. Later the version 1.1.2 was released on November 12, 2007. It provides the battery charge level indicator and international language & keyboard support.

Version 1.1.3 provides more feature then previous versions. It added the mail, maps, stocks, weather and notes applications for iPod touch, enable lyrics support for music and chapter support for movies. Besides, it increased the SMS storage capacity from 1,000 to 75,000 and gains the new feature on Google Maps, the 'Locate Me' can determine the phone's approximate location. Version 1.1.4 and 1.1.5 was released on February 26, 2008 and July 15, 2008 respectively. They fixed bugs and improve the speed of interface. The version 2.0 was available with the release of the iPhone 3G on July 11, 2008. It enhanced the operating system, making it able to turn Wi-Fi back on in Airplane mode and support SVG and Cisco IPsec VPN. 2.0 also enhance the language by providing extra keyboard languages and support the traditional and simplified Chinese handwriting recognition.

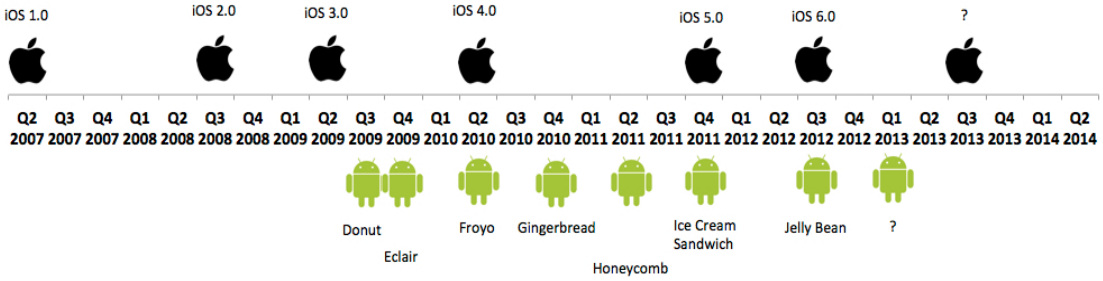

Fig 2: Android Vs IOS

II. MOBILE DEVICE PROGRAMMING

Mobile devices have evolved dramatically over the past decade. They have grown from cellphones that could just make calls to full-fledged computers capable of doing nearly everything a desktop computer can. With them, users can find information on a variety of levels, ranging from social networks to corporate data and e-mail. Coupled with the evolution of high-speed data networks, mobile devices are essential in an increasingly connected world. In addition, mobile devices have become multifunctional. They represent the device convergence that began at the start of the 21st century. No longer do users need to carry a phone, music player, and digital camera. Today's mobile devices can perform all these tasks and more with ease. Most mobile devices are also "location-aware," meaning they use the global positioning system (GPS) to determine where they are in the world. As a developer, you have access to all these features when you create applications. As of this writing, three major consumer operating systems (OSs) are available on mobile devices: Apple iOS on the iPhone, iPad, and iPod Touch; Google Android on Android devices ranging from smartphones to tablets; and Microsoft Windows Phone 7 on a variety of Win Phone devices. Arguments can be made for the primacy of each platform. In terms of model units sold, iPhone probably reigns supreme. Apple sold more than 19 million units in the second quarter of 2013. In terms of OS units sold, Android is probably at the top.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (260-268), Month: July - September 2014, Available at: www.researchpublish.com

Fig 3: Application Developing

A. Risky App Behaviors: Ios Vs. Android

Of the 100 free apps – 50 Android apps and 50 iOS apps in five equivalent categories – iOS apps exhibited more risky behaviors. In fact, all 50 iOS apps (100%) and 46 of the Android apps (92%) send and receive data without encryption. This potentially includes user data collected by the app and delivered back to the developer. The results show that iOS apps have more access to user data. The majority of iOS apps track for location (60%), share data with advertising or analytics networks (60%) and have access to the user's contact list (54%). A small percentage of iOS apps also had access to the user's calendar (14%). Android apps were not too far behind. Half of the Android apps shared data with ad networks and/or analytics companies, and 42% tracked for location. However, substantially fewer Android apps had access to contacts (20%) and none of them accessed the user's calendar.

An interesting emerging trend is the popularity of single sign-on (SSO) support on both iOS and Android. SSO can be great for users from a functionality perspective, allowing them to leverage Facebook, Twitter, or other popular social networking authentication methods (username and password). However, common security vulnerabilities in SSO methods can also be detrimental to any app that incorporates the faulty SSO feature.

B. Developing For IOS

Like in the Android part before, the first application we created for iOS was Hello- World. As everyone with any knowledge of programming knows, HelloWorld is usual- ly the simplest possible kind of software. It is basically used for demonstrating the syn- tax of a given programming language. Normally its sole purpose is to print out the text "Hello World" and do nothing more.

The behavior of the example demonstrated here is quite similar to that, except that the user first needs to click a button after which the application prints out "Hello, User!" The basic idea is the same, but we just added the button to demonstrate at least some sort of dialog between the user and the application.

As the previous chapter explained, the creation of a new project in Xcode 3.2.6 in- cludes four steps. After creating a new project we began the implementation. Allan's (2010, 34–38) HelloWorld is the source for our example of the same application. After implementing our example application according to the instructions we ran it. The application was built and run without errors. We used iPhone Simulator with the version 4.3 to see that the application is working the way it should. The source code for the application is found in Appendix B.1. - HelloWorld for iOS.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (260-268), Month: July - September 2014, Available at: www.researchpublish.com

Fig 4: Apple Phone

III. SYSTEM ANALYSIS

A. Evaluating the Methods

Our methods for this thesis included theoretical and empirical research. For theoretical research we read related articles, documentation and other literature. The material was mostly provided by the systems' developers, Google and Apple, but also by 3rd parties such as authors, and company - and developer bloggers. For empirical research we in- stalled and deployed the SDK environments, developed applications, and ran them on the built in emulator or simulator and finally on the actual mobile devices.

These research methods suited our needs well since our scope consisted of getting fa- miliar with the installation of the SDK's, and development, deployment and releasing process of an application. Being completely new to the mobile development in general, a lot of background work was needed to get familiar with both environments. On the other hand, the vast amount of background work was nicely balanced by the learn-by- doing –type of working method.

Since both of us have a technical background, the idea of first developing and then actually running the applications on real devices was a good motivator for this re- search. Our background also made it relatively easy to understand the concepts of the processes regarding application development for mobile devices.

B. Evaluating the Research Results and Their Validity

In our opinion, the research was a success. Throughout the research, we managed to keep to our original objective in everything we did. After conducting the needed back-ground research, we were able to install both of the SDKs and create and deploy appli- cations with them. We were also successful in finding out the costs and related license issues in using the SDKs in educational environments. In the end, we were able to find a group of similarities and differences between the development environments and we documented all steps taken as was mentioned in the objective.

C. Evaluating the Learning Process

In general, we learned a lot while conducting this thesis. Neither of us knew the programming languages used with either Android SDK or iOS SDK before this research. In addition to that, we were not at all familiar with developing mobile applications, and we were both completely unfamiliar with the Apple environment. Neither of us had ever even used an Apple product before which created a challenge to begin with.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (260-268), Month: July - September 2014, Available at: www.researchpublish.com

As both development environments are well documented, it was not difficult to find information. Quite the opposite, sometimes the amount of information was over- whelming, since there was a lot of overlapping within the documentation. We were also forced to be critical and learn how to filter out unwanted information, so we would stay within the scope. In the end, we got very familiar with the developer sites of both systems. As both systems are very popular, it was also easy to find answers to technical questions which rose mainly during the different development phases. Solv- ing these problems was also a good way of learning about the systems.

IV. OVERALL COMPARISON

we gathered our findings together, performed the comparison between the two operating systems and development environments related, and finally summarized the results. The pros and cons of both operating systems are discussed freely under their own subheading by the person responsible for the given system.

A. Pros and Cons of Using Android as A Platform

We were both generally very pleased with Android as a mobile operating system. At least from a technically oriented person's point of view, it is very nice that the system is almost fully customizable yet very usable. Despite the technical strengths, Android is not any harder to use in everyday life than any other operating system. The menus are built logically and the application icons clearly demonstrate which application they represent.

Android is an open source system, it has been developed to be like that from the very beginning. This makes it usable with basically every mobile device possible, unless the use of other systems has been prevented in the specific device. The fact that Android is an open source system means that developers and manufacturers can develop the sys¬tem onwards to suit their own needs. In fact, many manufacturers port the base platform of Android into their user interfaces, an example of this is the Sense GUI made by HTC.

	iOS (iPhone)	Android
Developer	Apple	Google
Copy/Paste	✔	✔
Multitasking	✔	✔
Flash Support	✘	✔
Silverlight Support	✘	✘
HTML5 Support	✔	✔
Unified Inbox	✔	✔
Exchange Support	✔	✔
Threaded Email	✔	✔
Visual Voicemail	✔	✔
Video Calling	✔	✔ Third Party App
Universal Search	✔	✔
Internet Tethering	✔	✔
Removable Storage	✘	✔
Facebook Integration	✘ (Third Party App)	✔ (Third Party Integration)
Twitter Integration	✘ (Third Party App)	✔ (Third Party Integration)
Folders	✔	✔
Apps Organization	Customizable	Customizable
App Store	300,000+ Apps	90,000+ Apps
Microsoft Office Support	Third Party App	Third Party App
Widgets	✘	✔
Media Sync	iTunes Mac & PC	Direct File Transfer + Third Party Software
X-Box Live Integration	Via Third Party App	Via Third Party App

Fig 5: Comparison

V. SIMULATION RESULT

The number of Apple-Inspired by these results, we have examined the geographic distributions of these two products. Our method here was to first aggregate Apple and Android adoption totals over Norwegian postal codes, and then we take the ratio of the two. Figure 5 shows the results superimposed on a map of Norway. What we find, very simply, is that Apple is dominating in Norway's cities. Since these results are also from Q3/2013, there are roughly equal numbers of Apple and Android phones—so that Apple cannot win everywhere. Thus we see a rather stark urban/rural dichotomy, with Apple dominating the cities and Android turning up as scattered blue spots in the countryside.

We conjecture (but have not yet tested) that the high-centrality users (as measured by eigenvector centrality) are concentrated geographically in the cities (just as they are concentrated, by definition, in the dense core of the social

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (260-268), Month: July - September 2014, Available at: www.researchpublish.com

network). In any case, all of the above results give a picture of Apple users as being more attracted to other Apple users than are Android users to other Android users—but also, more social in general. To test this idea, we show in Figure 6 the average degree Android links.

From these data we can get the average number of Apple and Android friends an Apple user has—and the same for an Android user. The result was clear: the average Apple user had over two times as many Apple friends as statistically expected from no preference—while all other results (number of Apple friends of Android users, and number of Android friends of Apple and Android users) were statistically consistent with no Apple/Android preference. In short: restricted to smartphone users, we again find that Apple users have more friends, and a stronger preference for their 'own kind'.

Using our geographic information on subscribers, we have displayed the results in Figure 6 in terms of three broad geographic categories—"urban", "small town", and "rural". Here we see a clear result that is counterintuitive: for all three groups of nodes (Apple, Android, all), we find that the average degree centrality increases steadily as one moves from urban to small town to rural. This result is not we looked at the growth of the iPhone adoption network over time, showing clearly the development of a 'social monster'—a giant connected component of the adoption network which shows the fastest growth. We equated the strength of this monster with the presence of iPhone adopters in the 'dense core' of highly central subscribers—a sign of success of the product in taking off. Presence in the dense core is also inevitably associated with a high density of adopter-adopter links—a sign that the product adoptions is 'social'. Here, in using the term 'social adoption', we do not attempt to distinguish homophily effects from true inter-customer influence: we simply seek to measure the tendency for those who talk together to adopt together.

we compare the growth of the Apple adoption network with that of the Android adoption network, on a quarterly basis. In each case, we start with the quarter in which the 'product' was first launched. While we see no dramatic difference in the first-quarter picture (Fig 1(a)), it is clear that already, two quarters later (Fig 1(c)), the Apple 'monster' (Largest Connected Component - LCC) is growing much more rapidly than the Android monster. This holds not only for total adopters. The black dotted curve in Figure 2 gives the number of adopter pairs expected, for the given total number of adopter pairs on the fixed call network, if adoption was purely random.

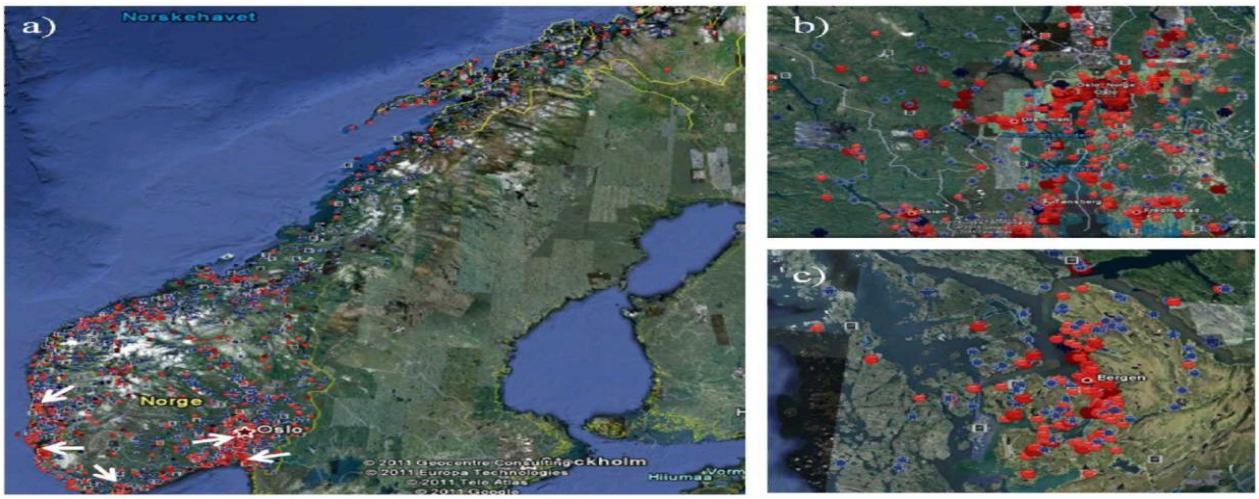

Fig 6: Android Vs IOS

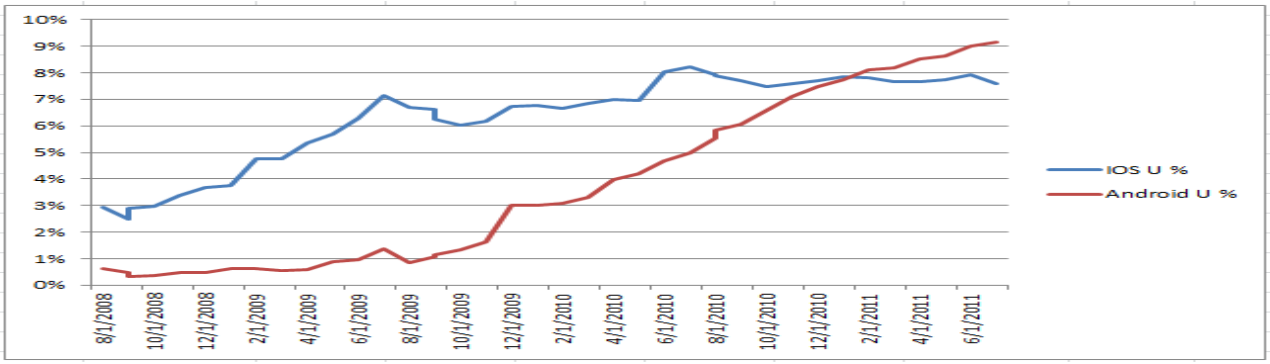

Fig 7: Android Vs IOS

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (260-268), Month: July - September 2014, Available at: www.researchpublish.com

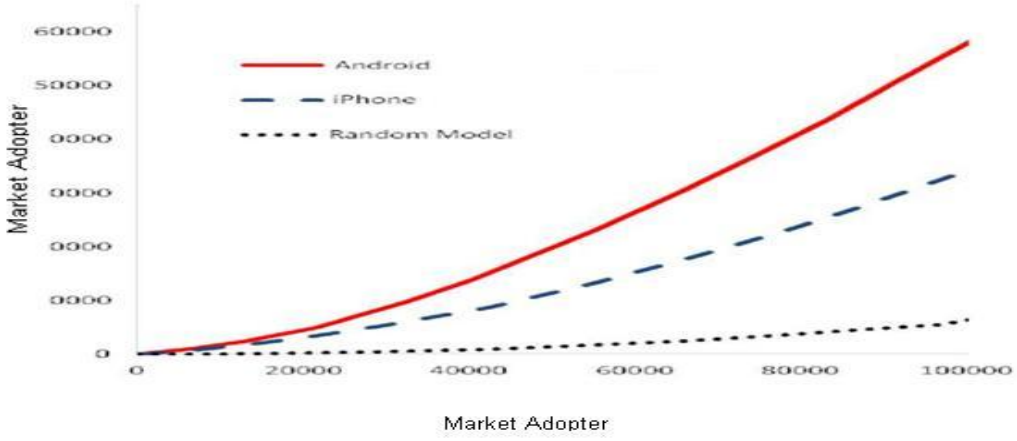

Table : SmartPhone

Top Smartphone Platforms 3 Month Avg. Ending Sep. 2012 vs. 3 Month Avg. Ending Jun. 2012 Total U.S. Smartphone Subscribers Ages 13+ Source: comScore MobiLens			
	Share (%) of Smartphone Subscribers		
	Jun-12	Sep-12	Point Change
Total Smartphone Subscribers	*100.0%*	*100.0%*	*N/A*
Google	51.6%	52.5%	0.9
Apple	32.4%	34.3%	1.9
RIM	10.7%	8.4%	-2.3
Microsoft	3.8%	3.6%	-0.2
Symbian	0.9%	0.6%	-0.3

VI. CONCLUSION

In conclusion, both of the software development kits researched here have their strengths and weaknesses. Both SDKs are equally usable and capable of completing the same tasks, but they are still somewhat out of each other's league. The user of the iOS SDK needs to be an Apple-person at least at some level; it's not possible to develop and test the application without an Apple computer and mobile device. The Android SDK user can enjoy the positive sides of the system being open source; the SDK works on any operating system and the Android system is available for many different devices.On the other hand, while installing the SDK at first, the Android SDK user needs to do much more work as many different systems need to be combined together, as the iOS SDK basically installs with the click of a button. Though, if taking the application development to the level of actual publishing, Android SDK is easier to use as iOS SDK requires the use of many different digital assets.

REFERENCES

[1] C. Van den Bulte and S. Wuyts,"Social Networks and Marketing", Marketing Science Institue 2007.

[2] S. Hill, F.Provost and C. Volinsky, “Network-Based Marketing: Identifying Likely Adopters via Consumer Networks”, Statistical Science. 2006, Vol. 21, No. 2, 256-276.

[3] Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., Nanavati, A. A., and Joshi, A. 2008. “Social ties and their relevance to churn in mobile telecom networks”. In Proceedings of the 11th international Conference on Extending Database Technology: Advances in Database Technology (Nantes, France, March 25 - 29, 2008). EDBT '08, vol. 261. ACM, New York, NY, 668-677.

[4] J.P. Onnela,J. Saramäki,J. Hyvönen, G. Szabó,D. Lazer,K. Kaski,J. Kertész and A.-L. Barabási,"Structure and tie strengths in mobile communication networks." Proc Natl Acad Sci U S A. 2007 May 1; 104(18): 7332–7336.

[5] Sinan Aral, Lev Muchnik, and Arun Sundararajan. "Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks." Proceedings of the National Academy of Sciences, 106(51):21544–21549, December 2009.

[6] P Sundsøy, J Bjelland, G Canright, K Engø-Monsen, R Ling, "Product adoption networks and their growth in a large mobile phone network", IEEE Advanced in Social Network Analysis and Mining (ASONAM 2010).

[7] N Eagle, A. Pentland, D Lazer, P Alex "Inferring friendshop network structure by using mobile phone data", National Academy of Sciences 106.36 (2009) 15274-15278, 2009.

[8] Rushi Bhatt, Vineet Chaoji, and Rajesh Parekh. 2010. Predicting product adoption in large-scale social networks. In Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM '10). ACM, New York, NY, USA, 1039-1048.

[9] P. Sundsøy, J. Bjelland, K. Engø-Monsen, G. Canright, R. Ling, "Comparing and visualizing the social spreading of products on a large-scale social network" ,

[10] To appear in "The influence on Technology on Social Network Analysis and Mining, Tanzel Ozyer et.al (Springer 2012).