

УДК 004.4'2

REVIEW OF CROSS-PLATFORMS FOR MOBILE APPLICATION DEVELOPMENT

Chepovetskyi O.D.

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, Kyiv

Researched most popular modern cross-platform tools for developing mobile applications, described advantages and disadvantages of their use, considered most popular mobile operating systems, described kinds of mobile applications. Analysis shows current trends in the development of cross-platform mobile applications.

Keywords: mobile application, cross-platform tools, Android, iOS, BlackBerry OS, Windows Phone, Firefox OS, Sencha, PhoneGap, Appcelerator Titanium.

Чеповецький О.Д. Огляд крос-платформних засобів для розробки мобільних додатків / Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Україна, Київ

Досліджено найпопулярніші сучасні крос-платформні засоби для розробки мобільних додатків, наведено переваги та недоліки їх використання, розглянуто найпоширеніші мобільні операційні системи, описано види мобільних додатків. Аналіз показує сучасні тенденції в розробці крос-платформних мобільних додатків.

Ключові слова: мобільний додаток, крос-платформні засоби, Android, iOS, BlackBerry OS, Windows Phone, Firefox OS, Sencha, PhoneGap, Appcelerator Titanium.

Чеповецький А.Д. Обзор кросс-платформенных средств для разработки мобильных приложений / Национальный технический

университет Украины «Киевский политехнический институт имени Игоря Сикорского», Украина, Киев

Исследованы самые популярные современные кросс-платформенные средства для разработки мобильных приложений, приведены преимущества и недостатки их использования, рассмотрены наиболее распространенные мобильные операционные системы, описаны виды мобильных приложений. Анализ показывает современные тенденции в разработке кросс-платформенных мобильных приложений.

Ключевые слова: мобильное приложение, кросс-платформенные средства, Android, iOS, BlackBerry OS, Windows Phone, Firefox OS, Sencha, PhoneGap, Appcelerator Titanium.

Introduction. Mobile technology has become an important part of people's life. The number of smartphone and tablet users is increasing, and so is the need for efficient mobile application technologies. This tendency has forced companies to develop mobile applications in order to follow current market standards and satisfy customers. Cross-platform mobile development is a new area of software engineering that allows the companies to reduce development time and cost. One of the obstacles companies face is development for an increasing number of mobile platforms. Building a different app for each platform is very expensive. This has been the main reason why cross-platform tools were created.

Analysis of recent research and publications.

According to Gartner [1] the total number of mobile app store downloads worldwide will increase to 268 billion in 2017. 42% of organizations expect to increase spending on mobile app development by an average of 31% in 2016, according to a recent survey by Gartner [2].

Highlights of the aspects of the problem, to which this article is dedicated to. Specifically, the paper surveys some cross-platform mobile development tools available on the market, provides the benefits and drawbacks of their use, considers some widely used mobile operating systems, describes the types of mobile applications.

The main presentation of the studies' materials. A mobile OS is an operating system (platform) which operates digital mobile devices e.g. smartphones and tablets. The most common mobile operating systems are the following:

- Android. It is released by Apache license and built on Linux. Android app can be built using Windows, Mac, or Linux and, for example, Java as a primary language of Android. It is a software package or software stack for mobile devices, containing an operating system, middleware, and main applications. Native Android applications are developed in Java, C++, and C programming languages with the help of Android Software Development Kit (SDK), which provides all the necessary tools and Application Programming Interfaces (APIs) [3]. The Google play store is the official site and portal for Android app.

- iOS is the operating system originally created by Apple Inc. mainly for the following products: iPhone, iPad and iPod. iPhone SDK, provided by Apple, includes XCode IDE, iPhone simulator and a suite of additional tools for app development. Swift and Objective-C and programming languages and XCode IDE and Xamarin Studio are used for developing native mobile applications for the iOS platform. App Store provides user facility to search and download app developed by iOS SDK.

- Windows Phone is a mobile OS created and released by Microsoft. The native mobile applications for this platform are mainly developed with

C# and VB.Net programming languages, with the use of Windows Phone SDK, which contains all the necessary tools needed to develop a native app for Windows Phone OS.

- Blackberry OS is an operating system which was created by Blackberry Inc. for its' Blackberry smartphone devices. The native mobile applications are created in Java using QNX Momentics IDE.

- Firefox OS represents new generation of mobile operating systems, namely a web based OS. It is designed based upon open standards and approaches from HTML5 applications, JavaScript and web APIs. This approach brings open web APIs communicating directly with cellphone hardware and it also features a direct link to the web-based application marketplace [4].

A few years ago cross-platform tools were not trusted enough to be used in the industry, but nowadays have improved and matured. Companies are more often using these tools to decrease the programming time and publish the app for all platforms simultaneously. Each one of the aforementioned platforms requires a particular programming language, different development environments and programming models based on platform-specific APIs. It is apparent that, if a company decides to support both Android and iOS platform, there is a constraint to maintain two versions of a single product: one version implemented with e.g. Java for Android and a second version implemented using Swift or Objective-C for iOS.

As the concept of write once, run anywhere can't be applied when building native applications, the best alternative option for companies is the cross-platform mobile development. Cross-platform development simplifies the maintenance and deployment processes, and saves development time and effort.

Using cross-platform software development environments has several benefits:

- Development time reduction. The fact that the code is written once, but compiles to several mobile operating systems reduces development time.
- Ease of development. Cross-platform tools, most often, require one programming language in the development for all the platforms. The tools usually require easy-to-learn web development programming languages: HTML5, CSS3 and JavaScript.

- Ease of maintenance. It is cheaper and easier to maintain one code base as opposed to maintaining different codes for different mobile platforms [5]. Even though cross-platform tools have many advantages, there are some known drawbacks of using them as well. For example:

- Performance. Poor performance in comparison to native applications.
- Access to native API. Not all the tools have access to all the mobile devices' functionality, e.g. access to camera, accelerometer, and storage.

- Reliability. Your project will be dependent on the tool and there is no guarantee that the tool will continue to exist.

- Up to date. Mobile platforms are changing at a fast pace and it takes cross-platform tools time to support the new features in the new mobile platform [5].

According to Korf and Oksman [6] there are mainly 3 deployment formats of applications that can be created with cross-platform mobile development tools: web, hybrid and native apps. Native apps live on the device and are accessed through icons on the device home screen. Native apps are installed through an application store (such as Google Play or Apple's App Store). They are developed specifically for one platform, and can take full advantage of all the device features – they can use the camera, the GPS, the accelerometer, the compass, the list of

contacts, and so on. And native apps can use the device's notification system and can work offline [7]. Developers use the development tools and language that the respective platform supports (e.g., XCode and Objective-C with iOS, Eclipse and Java with Android). Examples of native apps are Angry Birds and Shazam. Native apps have a few disadvantages. Maintenance of the native app is complicated task both for the users and developers. Developers have to program it according to different platforms and users have to update it regularly. The development cost of this app is more if you are making application for different platforms. Sometimes it becomes very difficult for the developer to give maintain and offer support as users of different mobile may be using different versions of the apps.

Web apps are not real applications; they are run by a browser and typically written in HTML5. Web apps became really popular when people realized that they can obtain native-like functionality in the browser. With HTML5 the distinction between web apps and regular web pages has become blurry. Users need not to go to mobile app store to update or download the application. There will be the single universal version which can be used by any mobile platform. Hence, the maintenance cost of the web app is low [8].

On the other hand, web app has some shortcomings also. Internet connection is a must in web apps. Web apps are not compatible with smartphone features like camera, GPs, phone dialing, etc. Performance of web apps is slower as compared to native apps. They are also more difficult to build a regular user-base, unless they save it as a bookmark. Users won't have the app's icon on their devices.

Hybrid apps combine technologies from native and mobile Web apps to gain the benefits of each. They behave like a native app because they are installed from a web store and have access to device specific features

as in native app but developed using web app tools. Hybrid mobile apps can be released on multiple platforms when using certain web technologies like HTML5, CSS3 and JavaScript. It will save the overhead time and cost used to prepare software for each platform. Facebook, Netflix, LinkedIn are some of the examples of hybrid apps [8].

According to survey's of businessofapps.com result [9], the top three cross-platform tools currently on the market are: Sencha, PhoneGap, Appcelerator Titanium. This subsection gives a detailed description for each of the mentioned tools.

1. Sencha: Sencha API is pure JavaScript. It uses XML and HTML to create interface design and procedural code for creating a UI object. Once a code is made or converted to HTML5 format it is very simple and quick to use them to both iOS and Android.

Strengths: It enables developers to build powerful applications that work on iOS, Android, BlackBerry, Windows Phone, and more. Sencha is not dependent on jQuery, so is compatible with both the iPhone and Android. Has a very solid documentation and a large community. Native packaging is possible via Apache Cordova/PhoneGap or Sencha's SDK.

Weaknesses: It relies on a heavyweight framework, which is quite inconvenient. Another drawback is complex learning: sometimes it is not worth it to go deep into JavaScript. In addition, the UI development is quite hard, as the elements are added with JavaScript and not HTML [10].

2. PhoneGap: PhoneGap helps first time app developers to translate their codes from HTML5, JavaScript and CSS. The PhoneGap JavaScript API provides the access to the native functionality (e.g. camera and contacts). The best suited projects for this tool are existing web applications, needed to be converted into a mobile application. Strengths:

One of the biggest benefits provided by PhoneGap is a variety of plugins to choose from. It enables developers to create Android and iOS apps supporting them in the cloud without the need of local SDKs. Thus, a non-Mac user can develop iOS-based applications using Apache PhoneGap[9]. Finally, PhoneGap has a strong developers community.

Weaknesses: The user interface quality is the biggest issue of PhoneGap due to the quality of the platform's rendering engine. Web views on different platforms have some limitations, and this is why the quality of PhoneGap UI is significantly lower than applications with a native UI [10].

3. Appcelerator Titanium: Using JavaScript, Titanium's SDK creates native iOS and Android apps while reusing anywhere from 60% to 90% of the same code for all the apps you make. Appcelerator Studio is an add-on Eclipse-based IDE, which contains an editing tool with Git integration, inline debugging and cloud connection.

Strengths: One of the goals of Titanium is to reduce the gap between the native devices and cross-platform mobile development tools. It is possible to use native UI extensions in Titanium. Appcelerator provides value-adds such as a BaaS, app analytics and a marketplace for 3rd party components. Weaknesses: Titanium is only available for a few mobile operating systems, due to the complications with development of Titanium API for all platforms. The development complexities (and costs) rise more than proportionally to application complexity increases. Documentation isn't always great.

Conclusions. This research evaluated major cross-platforms for mobile development. The ultimate goal of cross-platform mobile app development is to achieve native app performance and run on as many platforms as possible. The current state of the cross-platform mobile

development tools market is dynamic, which means that the researches have to be conducted several times a year.

References:

1. *Gartner Says Mobile App Stores Will See Annual Downloads Reach 102 Billion in 2013 (2013)* <http://www.gartner.com/newsroom/id/2592315>
2. *Gartner Survey Reveals Enterprise Spending on Mobile App Development Remains Low (2016)* <http://www.gartner.com/newsroom/id/3353317>
3. X. Shu, Z. Du, and R. Chen, "Research on mobile location service design based on Android", *Wireless Communications, Networking and Mobile Computing*, 2009, pp.1–4.
4. Tor-Morten Grønli, Jarle Hansen, Gheorghita Ghinea, Muhammad Younas. *Mobile application platform heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS.*
5. J. Parker. "Pros and cons of cross-platform mobile app development tools", March 2013. <http://www.techrepublic.com/blog/app-builder/pros-and-cons-of-cross-platform-mobile-app-development-tools/2379>
6. M. Korf and E. Oksman. "Native, HTML5, or Hybrid: Understanding your mobile application development options." [http://wiki.developerforce.com/page/Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options](http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:Understanding_Your_Mobile_Application_Development_Options)
7. <https://www.nngroup.com/articles/mobile-native-apps/>
8. Nabil Litayem, Bhawna Dhupia, Sadia Rubab. *Review of Cross-Platforms for Mobile Learning Application Development*
9. <http://www.businessofapps.com/top-5-cross-platform-mobile-application-development-tools/>

10. *Alireza Pazirandeh, Evelina Vorobyeva. "Evaluation of cross-platform tools for mobile development"*