EasyChair Preprint

№ 3905

A Comparative Study of Hybrid Mobile Application Development

Ashikur Rahman

July 18, 2020

A comparative study of hybrid mobile application development

Ashikur Rahman

COMP8701 Fundamentals of Computing GE

Flinders University

rahm0137@flinders.edu.au

Abstract

With the fast advancement of smartphones, the use of mobile application is growing drastically. However, mobile applications are platform dependent which makes the development more challenging and costly. Hybrid or cross platform mobile application development is a relatively new concept where developers use a single code base to build applications for different platforms (eg. Android, iOS etc). This paper discusses the pros/cons of hybrid application development, introduces and compares different popular tools/frameworks and also compares performance and other attributes with native application development.

1 Introduction

Can we remember a day when we did not use our mobile phone? Our new generation won't believe that there was a time when we did not have cell phone in our pocket. The advancement of mobile phone and its evolution to become smarter day by day has been so fast that we forget this fact. Today 45.4% of total population of the world uses smart phone ("Smartphone users worldwide 2020 | Statista", 2020).

Though the first smartphone was built by IBM in 1992, the main evolution begins with the creation of Android operating system in 2008. Its closest competitor iOS was also launched officially for iPhone in 2007. Currently, Android has 73.3% market share worldwide while iOS has 25.89%. ("Mobile Operating System Market Share Worldwide | StatCounter Global Stats", 2020). Besides making calls one of the main features of smartphones is using applications for different purposes. Both Android and iOS has their own marketplace where they store these apps and any user can download them free or with payment. Most of the apps are developed by other third party companies or individuals. However Android, iOS and all other platforms have developed their own set of standards, tools, languages to develop their own apps. That means Android app will not run in iOS device and vice versa (Dhillon and Mahmoud, 2014).

The app development industry is booming these days and the demand for relevant app developers is also increasing. However, one of the main challenges of this industry is – they need to make separate apps for Android and iOS devices.

To achieve this, one obvious solution is to have separate development team for Android and iOS (and any other platforms they care) which increases time and cost significantly (Delia, L., et al., 2019). There are many small to medium business who can't afford it. That's why the search of a solution to develop cross platform application using single code base was a hot topic in the last few years. Some companies, including Google and Facebook, already came forward with different solutions and tools/frameworks. However, there are obviously both pros and cons if we compare hybrid/cross platform application development with native application development (El-Kassas, Abdullah, Yousef & Wahba, 2017).

2 Development Approaches

Native platforms provide different tools and frameworks for app development. For example, Android provides a tool called Android Studio which is a bundle of IDE, compiler and other tools that are needed for app development. However, generally Android developers should have sound knowledge in the Java programming language. For iOS, apple also provides a similar tool called Xcode. iOS developers need to be efficient in Swift Language or Objective-C (P. Que, X. Guo and M. Zhu, 2016). So the whole development and build process is different. Also, developers need to have a mac device usually to build native iOS apps (El-Kassas, Abdullah, Yousef & Wahba, 2017).

For hybrid app development, usually it is written in web languages (HTML, CSS, JavaScript). Web languages are easy to learn and code. However, there are different tools and frameworks that provide API to control device's native features (eg. Camera, GPS etc) and then finally build the app to desired formats. So developers write code once and then use such tools (eg. Phonegap, Ionic) to build apps for different platforms (Bosnic, Papp and Novak, 2016).

For both native and hybrid approach, once the development is finished, developers need to upload the app to platform specific app store separately.

3 Popular frameworks and tools

One of the most popular hybrid development frameworks is Apache Cordova. It enables developers to use web languages. It provides different plugins to play with devices native functions (S. Bosnic, I. Papp and S. Novak, 2016). Adobe acquired Apache Cordova and later released it as an open source project ("Apache Cordova", 2020). Adobe also built another commercial framework called Phonegap, which was actually built on top of Cordova. There are many other popular frameworks that are built on top of Cordova framework, such as - Ionic, Intel XDK, Framework 7 etc**.** Apache Cordova is the original concept of hybrid app as it uses web view to render the user interface. It is neither a web app nor a native app (Achilleos and Kapitsaki, 2013).

Recently Google also introduced a new framework called 'Flutter'. However unlike Cordova, it uses Dart Programming Language. Flutter uses Google's Skia library as a rendering engine (Technical overview, n.d.).

One another popular framework is React Native which is developed by Facebook. It lets developers to use reactJS which is a javascript framework released by Facebook itself (Eisenman, 2015).

These different tools also provide many paid services. For example Ionic provides their own development tool, layout design tool, cloud service to build apps online instead of setting up local environments etc (Dunka, Emmanuel and Oyerinde, 2017). By using these services productivity can be increased by a number of times.

4 Performance & Usability Comparison

There are some obvious advantages and disadvantages of hybrid app compared to native apps. We already discussed few advantages of hybrid application development compared to native development. In summary advantages are:

1. Cost effective solution: it requires less development efforts and less developers.
2. It uses popular languages like HTML, JavaScript. So finding developers are easy and less costly.
3. Many platforms provide tools to design app easily. Some also provide cloud service to build the app online which reduces hassles for small to medium companies/teams and increases productivity.
4. Updating app becomes easier as developers do not need to modify codes separately for each platform whenever an update is required.
5. As Apache Cordova is an open source project, it has a strong community. There are lots plugins available for different purposes developed by community which makes it easier to develop apps.
6. Learning curve is very good as hybrid app development is easier to learn.

However some disadvantages can be:

1. App performance is the key factor where native app definitely is the winner. P. Que, X. Guo and M. Zhu done a research in 2016 where they done an analysis of performance comparing one natively developed app with hybrid app. They found that their hybrid app's installation time was 22.6% longer, CPU usage was 106% higher and memory space requirement was 73% higher (P. Que, X. Guo and M. Zhu, 2016).
2. Hybrid app's user interface doesn't give native feel usually (Xanthopoulos and Xinogalos, 2013).
3. It's normally not possible to develop apps that need intensive graphics or games with hybrid framework.

5 Real Examples of Hybrid Apps

It's not possible to understand if an app is developed natively or in hybrid approach by a normal user. There are lots of popular apps that we are using regularly are actually built using hybrid platforms. Facebook, Instagram, Pinterest, Skype, Tesla, Uber etc apps are developed using React Native. Though react native is not truly hybrid app as it doesn't use web view for user interface. But it uses same principles that hybrid app focuses, that is – cross platform development.

Ionic is used by MarketWatch, Pacifica, Sworkit, Nationwide etc apps.

Flutter gained popularity very quickly. Alibaba, Hamilton Musical, Greentea, Google Ads etc are developed by flutter (React Native vs. Ionic vs. Flutter: Comparison of Top Cross-Platform App Development Tools, 2019).

6 Future Prospects

A survey conducted by Appcelerator and IDC in 2012 showed that despite various difficulties and cost involved, companies are always interested to build apps that are supported by atleast major platforms like Android and iOS (Q3 2012 Mobile Developer Report, 2012).
There is no doubt that hybrid app development is becoming more popular day by day. Giants like Facebook or Google made their own framework which made this more interesting and competitive. Also the performance of hybrid app is increasing as these companies want to give the closest native feel. It can be assumed that the use of such framework will be increased significantly in future and also these frameworks will work hard to get rid of some of the negative issues that they have now.

7 Conclusion

This paper discussed about various hybrid development tools, compared with native development, advantages and disadvantages etc. It is obvious that native apps are still the winner from performance perspective. However, hybrid approach is also not far behind in the competition because it is cost effective and easier to develop and maintain. Also the performance is increasing day by day. There are various frameworks which gives native look and feel to hybrid apps.

However as still hybrid apps platforms pose many limitations, my finding is that it is wise to use hybrid platform to develop apps that are less graphics intensive. For simple app the performance difference is negligible. However for graphical intensive apps, games etc, native approach is better. It is faster, uses less resources of device.

References

Dhillon, S. and Mahmoud, Q., 2014. An evaluation framework for cross-platform mobile application development tools. Software: Practice and Experience, 45(10), pp.1331-1357.

Delia, L., et al. (2019). Development Approaches for Mobile Applications: Comparative Analysis of Features, Cham, Springer International Publishing.

El-Kassas, W., Abdullah, B., Yousef, A., & Wahba, A. (2017). Taxonomy of Cross-Platform Mobile Applications Development Approaches. Ain Shams Engineering Journal, 8(2), 163-190. doi: 10.1016/j.asej.2015.08.004

Achilleos, A. P. and G. M. Kapitsaki (2014). Enabling Cross-Platform Mobile Application Development: A Context-Aware Middleware. Web Information Systems Engineering – WISE 2014, Cham, Springer International Publishing.

Xanthopoulos, S. and S. Xinogalos (2013). A comparative analysis of cross-platform development approaches for mobile applications. Proceedings of the 6th Balkan Conference in Informatics. Thessaloniki, Greece, Association for Computing Machinery: 213–220.

S. Bosnic, I. Papp and S. Novak, "The development of hybrid mobile applications with Apache Cordova," 2016 24th Telecommunications Forum (TELFOR), Belgrade, 2016, pp. 1-4.

P. Que, X. Guo and M. Zhu, "A Comprehensive Comparison between Hybrid and Native App Paradigms," 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), Tehri, 2016, pp. 611-614.

Apache Cordova. (2020). Retrieved 25 March 2020, from https://en.wikipedia.org/wiki/Apache_Cordova

Mobile Operating System Market Share Worldwide | StatCounter Global Stats. (2020). Retrieved 25 March 2020, from https://gs.statcounter.com/os-market-share/mobile/worldwide

Dunka, B., Emmanuel, E. and Oyerinde, Y., 2017. Hybrid mobile application based on ionic framework technologies. International Journal of Recent Advances in Multidisciplinary Research, 4, pp.3121-3130.

2012. Q3 2012 Mobile Developer Report. [ebook] Available at: <http://www.appcelerator.com.s3.amazonaws.com/pdf/Appcelerator-Report-Q3-2012-final.pdf> [Accessed 15 April 2020].

2019. React Native Vs. Ionic Vs. Flutter: Comparison Of Top Cross-Platform App Development Tools. [online] Available at: <https://codeburst.io/react-native-vs-ionic-vs-flutter-comparison-of-top-cross-platform-app-development-tools-71c8011309ac> [Accessed 15 April 2020].

Flutter.dev. n.d. Technical Overview. [online] Available at: <https://flutter.dev/docs/resources/technical-overview> [Accessed 15 April 2020].

Eisenman, B., 2015. Learning react native: Building native mobile apps with JavaScript. " O'Reilly Media, Inc.".