

# Cross platform development vs native development

Nikita A. Shevtsiv, Andrii M. Striuk

Kryvyi Rih National University, 11 Vitalii Matusevych Str., Kryvyi Rih, 50027, Ukraine

## Abstract

The paper analyzes the advantages and disadvantages of cross-platform and native mobile application development. The conditions are highlighted in which native and cross-platform development reveal their advantages. These conditions include the project size, work comfort, popularity, relevance. It was concluded that a beginner developer should start learning from native development, and then try cross-platform.

## Keywords

native development, cross-platform development, React Native, Java, Kotlin, Flutter

## 1. Introduction

In 2010, in the world of mobile development, the main questions were: which to choose Android or iOS? What to learn: Java or Objective-C? Over the course of 10 years, these topics have gradually lost their relevance in the circles of the IT community, because alternatives to these languages have appeared. As Kotlin for Android and Swift for iOS. Parallel to this, a new branch was developing – cross-platform technologies.

In 2020, you can often hear controversy and speculation about which side will take the main part of the mobile app development market: native or cross-platform. Native development involves technologies that are focused on a specific platform. For example, the Java programming language is aimed at developing applications for Android, the brainchild of Apple – Swift for iOS.

For cross-platform development, technologies aimed at both platforms are used. Among them, two of the most promising can be identified: React Native (based on js) from Facebook and Flutter (based on Dart) from Google.

You can ask a mobile developer who has at least a year of experience working with cross-platform technologies and native development (it doesn't matter if it is an Android or iOS developer), he would answer: neither side will ever supplant the other.

But under certain conditions, these parties have clear advantages over each other.

---

CS&SE@SW 2020: 3rd Workshop for Young Scientists in Computer Science & Software Engineering, November 27, 2020, Kryvyi Rih, Ukraine

✉ [nikita0503ua@gmail.com](mailto:nikita0503ua@gmail.com) (N.A. Shevtsiv); [andrey.n.stryuk@gmail.com](mailto:andrey.n.stryuk@gmail.com) (A.M. Striuk)

🌐 <http://mpz.knu.edu.ua/pro-kafedru/vikladachi/224-andrii-striuk> (A.M. Striuk)

🆔 0000-0001-9240-1976 (A.M. Striuk)

© 2020 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

## 2. Comparative analysis of cross-platform and native development

### 2.1. Project sizes

Quite often, IT companies with up to 10 employees choose cross-platform. Typically, development teams take on small and medium-sized projects, where development time can take on average from 3 months to a year. As a rule, such projects do not require high technological needs from programmers. Therefore, in these cases cross-platform technologies can be used.

It will be beneficial for both the customer and the company.

The number of application developers can be reduced; most often it will be only one person. On the part of the customer, it will be more comfortable to communicate with one person, rather than explaining to two developers at once how this or that screen should work. In fact, not all people think the same way. And they can understand both the design part and the technical part in different ways, for example, the rules for loading and displaying elements on the screen. Accordingly, the appearance will be not similar on different platforms, and the speed of work will also differ. It will take more time to come to a similar result.

For a small company, being cross-platform has some advantages. Since fewer developers were required, it means that the costs for the application will be less.

It makes no sense to consider the option when one developer first makes an application for one platform, and then for a second one. Since it is obvious that this will take a lot of time and money, so it is not very profitable for both parties.

Will this affect the quality? Applications that do not require a lot of resources from smartphones, for example, simple social network or an online shop, is unlikely to require intervention in low-level processes (Although, in React Native, it is possible to write native modules). So the answer is no. This will not affect the quality and performance of simple applications in any way.

For clients who start their own business and already have a working website, as a project development, cross-platform application development will be cheaper and faster than native.

But this applies to small projects that will not need support, and even more so in further development. If we talk about projects with long development times, constant improvement and expansion of functionality, cross-platform is seriously inferior to native development.

In large projects, a lot of attention is paid to design, and quite often to animation of objects. Despite the fact that, according to the development ideology, this item is noted as a plus, practice shows that the implementation of a non-trivial design for two operating systems gives many bugs. This increases the time to eliminate errors. The UI is displayed differently and the time to adapt is also increased. The most noticeable problems occur with animations, clicks and scrolling – the application may freeze. The user interface is developed in HTML, but you will have to spend months to achieve native platform performance. The native development languages for Android and iOS would be better tools in this case than the cross-platform technology APIs.

You need to develop an interface that is intuitive for both iOS and Android users. Otherwise, an application built according to the IOS Human Interface OS Guidelines will be inconvenient for Android users. Ultimately, you will spend more time improving the user experience.

Long-term projects are not designed to save resources. In this case, customers are looking for performance, quality and reliability. Native development in large-scale projects will be the best solution. For a high-quality application, the project will need developers with serious experience and deep knowledge in the areas of development for Android (Java or Kotlin) and iOS (Objective-C and Swift). Compared to cross-platform technologies, there are 2 times more developers, plus the work of good specialists is more expensive, in other words, the project costs increase. But, it is worth recalling again, large projects involve serious costs, as a result of which, as a rule, this is justified.

Application speed – the compiled project code is optimal for the native platform. The application gets full hardware support for the device and uses multithreading for complex tasks.

With the release of updates to the programming language or hardware functionality, new features will be available immediately. That is, there will be no need to wait for the implementation and adaptation of these updates in cross-platform development technology.

And the most important advantage of native development is the absence of restrictions, and this, in turn, is the main condition for the development of huge projects. An indisputable advantage is that it uses all the functions of the system software as flexibly as possible, such as a camera, microphone, geolocator, accelerometer, calendar, media, at the same time, the application sparingly consumes memory and device battery.

## **2.2. Work comfort**

In addition to the fact that the technology must meet the requirements of the project, it is worth considering a number of factors that are associated with the developer.

The first thing to look out for is the ease of learning. The more popular the technology, the more courses, lessons and articles related to this tool can be found on the Internet. Native development has an advantage in this regard. For example, it is worth considering the Kotlin programming language, it is the official language for developing applications for the Android platform.

Why Google loved Kotlin? On the official website, Android developers write that they have watched the rise of Kotlin in recent years. Google is not shy about describing the language as impressive and concise, more powerful and enjoyable to work with. It has increased performance: the program code on it is, on average, 40% shorter than in other languages, and Kotlin also allows you to avoid some errors in the code. One of the determining factors in Kotlin's popularity with Google is that it is compatible with Java, which is already used in developing applications for Android.

In the future, more libraries and tools will be developed for the language, more teaching materials, it will be easier to find solutions to possible problems.

In other words, the company actively promotes and popularizes the language. Considering that Google is the giant of the IT industry, many companies follow the trend and release a number of courses and other training materials on Kotlin on their own. More and more articles related to the language appear on Habr, on Stack Overflow they solve a huge number of problems using this technology. Thus, the popularity of Kotlin is growing rapidly. It can be concluded that for beginners the entry threshold is greatly reduced and it becomes easier to learn, since there are a lot of materials.

On the other hand, there is the brainchild of the same corporation – Flutter. An incredibly promising technology that Google is promoting in the world of cross-platform development. Flutter, takes a different approach than React Native and Xamarin. It does not turn the source code into native code, which is executed by the platform, but in fact draws a window on the smartphone screen and renders all the elements itself. The language used is Dart, which Google created as an improved version of JavaScript. But as of 2020, the technology is still a bit dry. Due to the fact that Flutter is quite young, the amount of training materials is not as much as we would like, in addition, there are still few additional libraries.

The second point is the threshold of knowledge of entering technology. In other words: the required amount of effort and time that needs to be spent on learning technology so that the knowledge gained is enough to implement a simple project.

From this side, cross-platform comes out a little ahead. Take a Facebook product as an example.

React Native is a JavaScript framework for building natively rendered Android and iOS apps. It is based on Facebook’s React JS library for creating user interfaces. If a programmer has experience in web development, especially in ReactJS, then he can become a developer of applications for mobile devices in no time. Since ReactJS and React Native have 80% similarity in the code base, you can write a business card application for phones almost immediately.

In addition, there is a platform that makes it much easier for beginners to work with React Native.

Expo is a set of tools with which you can write a React Native app in minutes. It includes out-of-the-box tools like Android Studio and XCode configurations, Apple and Google certificate management, push notifications. The Expo development team has developed a wonderful tool called Snack that allows you to write and test code right in the browser, without any pre-installation or setup of development environments. Accordingly, the “threshold of entry” into React Native becomes very low. And thanks to good documentation from Facebook, going from web development to mobile development becomes a breeze. Especially, it is worth noting that this documentation provides materials for those who have experience with native development and recommends working with the React Native CLI. And for those who are new to mobile development, it suggests using the aforementioned Expo CLI [1].

It is worth considering the issue of the popularity and relevance of both sides in more detail.

### 2.3. Popularity

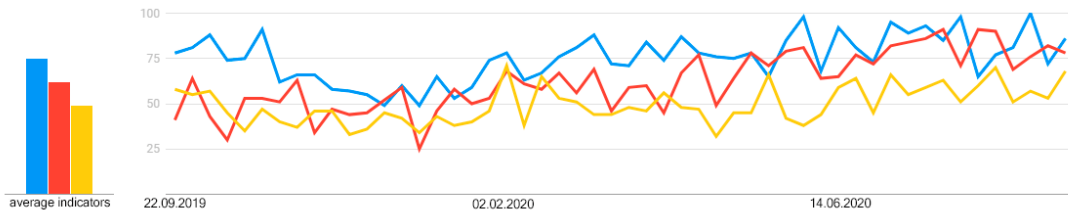
Google Trends [2] service will be used to research popularity.

Consider the popularity of cross-platform development technologies – React Native and Flutter – with the programming language for the Android platform – Kotlin.

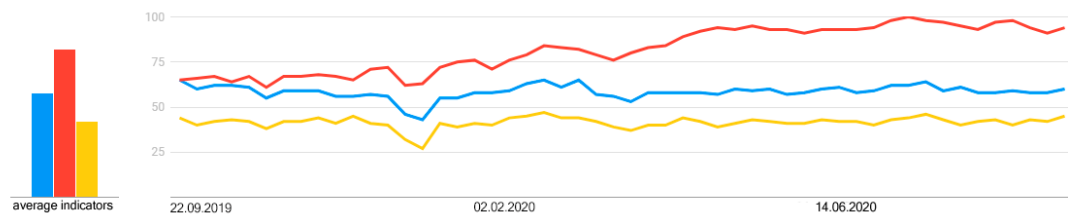
React Native is blue, Flutter is red, Kotlin is yellow (figure 1).

Judging by the graph, React Native has been leading in popularity in Ukraine over the past two years, but Flutter is gaining popularity and even bypasses RN in search statistics at some intervals.

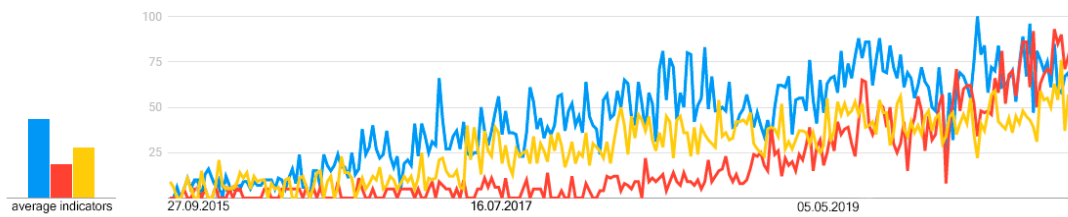
Figure 2 clearly shows how Flutter is gaining popularity every month, more and more. React Native and Kotlin have consistently maintained their popularity in 2019 and 2020.



**Figure 1:** Search statistics for technology names in Ukraine for 2019-2020



**Figure 2:** Search statistics for technology names worldwide for 2019-2020



**Figure 3:** Search statistics for technology names in Ukraine, over the last 5 years

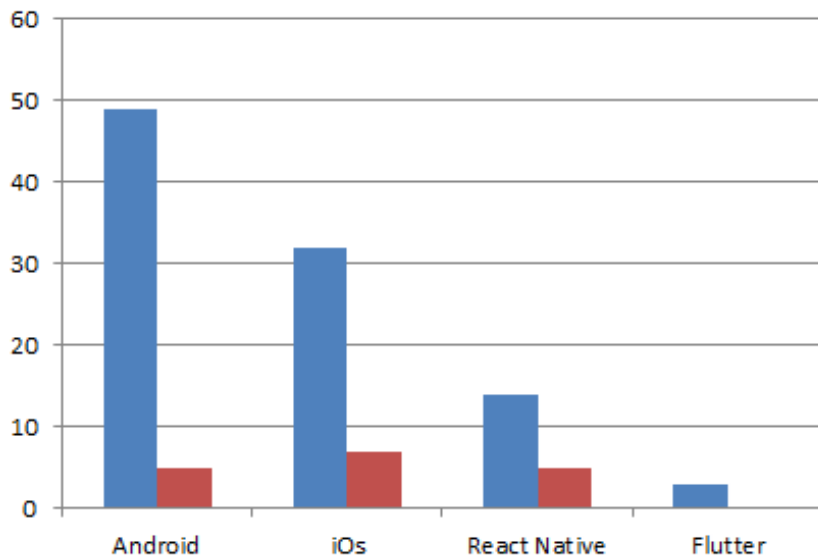
Figure 3 shows the popularity of Google searches for the current list of technologies over the past 5 years in Ukraine. React Native and Kotlin were significantly more popular than Flutter until 2019. But starting in June 2020, Flutter came out on top in the search volume ranking.

The fourth figure shows that over the course of 5 years, React Native has been more popular than the rest of the technologies on the list (despite 2020).

But making a choice, judging only by the number of searches for a programming language on Google, would be the wrong decision. First, the list does not include the Swift programming language for iOS. The fact is that Google Trends under the string “Swift” means not only technology for development, but also as an international interbank system for transferring information and making payments. Secondly, the very popular Java language was not included in the comparison, because it is used not only for developing Android applications, but also for other platforms (web, desktop, etc.). Accordingly, in this comparison, he would have left



**Figure 4:** Search statistics for technology names worldwide, over the past 5 years



**Figure 5:** Job statistics at work.ua

no chance for his opponents. In other words, relying on Google Trends is not a good idea.

But after analyzing many reviews and articles [3, 4] among developers, native development is very popular. Basically, cross-platform development is treated as a tool that only experienced developers use on small projects.

## 2.4. Relevance

The second factor, which is one of the most important in choosing a technology, is its relevance.

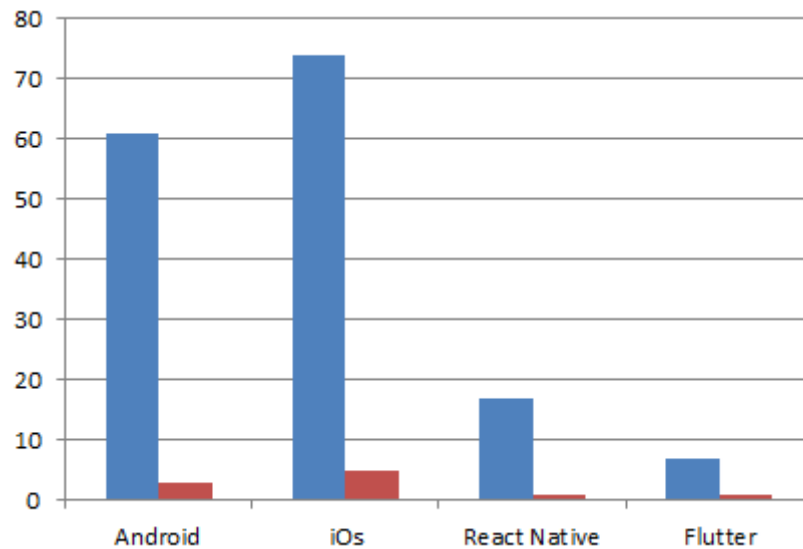
Consider the number of vacancies on the following sites: work.ua [5], rabota.ua [6], DOU [7], grc.ua [8].

On the graphs, the number of vacancies that are provided for working in the office is marked in blue, and on a remote basis, in red.

Statistics from these sites show that native development in offices is in demand more than cross-platform. But on a remote basis, all technologies have roughly the same number of va-

**Table 1**  
Statistics of vacancies at work.ua

OS / technology	office	remote
Android	49	5
iOS	32	7
React Native	14	5
Flutter	3	0



**Figure 6:** Job statistics on rabota.ua

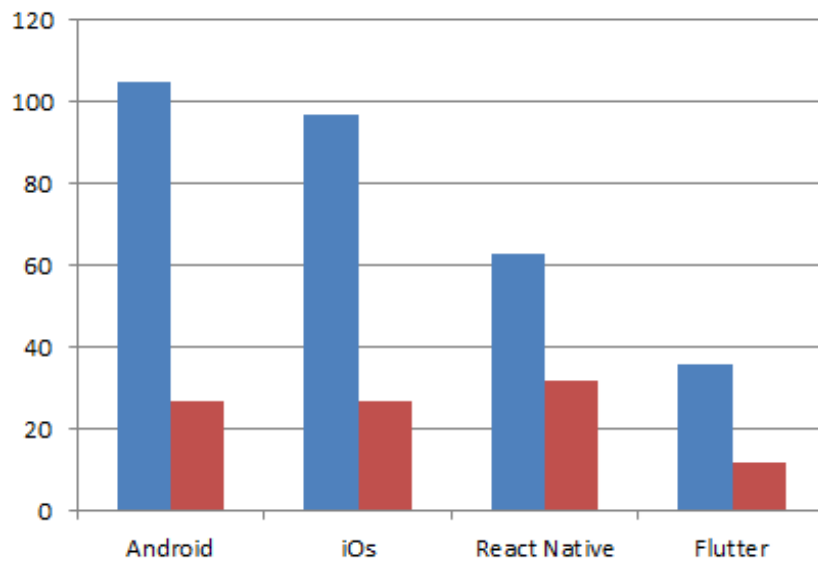
**Table 2**  
Statistics of vacancies at rabota.ua

OS / technology	office	remote
Android	61	3
iOS	74	5
React Native	17	1
Flutter	7	1

cancies.

### 3. Conclusion

In conclusion, it is worth concluding: there is an application of any technology in the labor market, since there are a huge number of ideas from startups to huge and complex projects. It is impossible to give a clear answer which is better: native or cross-platform development.



**Figure 7:** Statistics of vacancies on DOU

**Table 3**

Statistics of vacancies at DOU

OS / technology	office	remote
Android	105	27
iOS	97	27
React Native	63	32
Flutter	36	12

**Table 4**

Statistics of vacancies at grc.ua

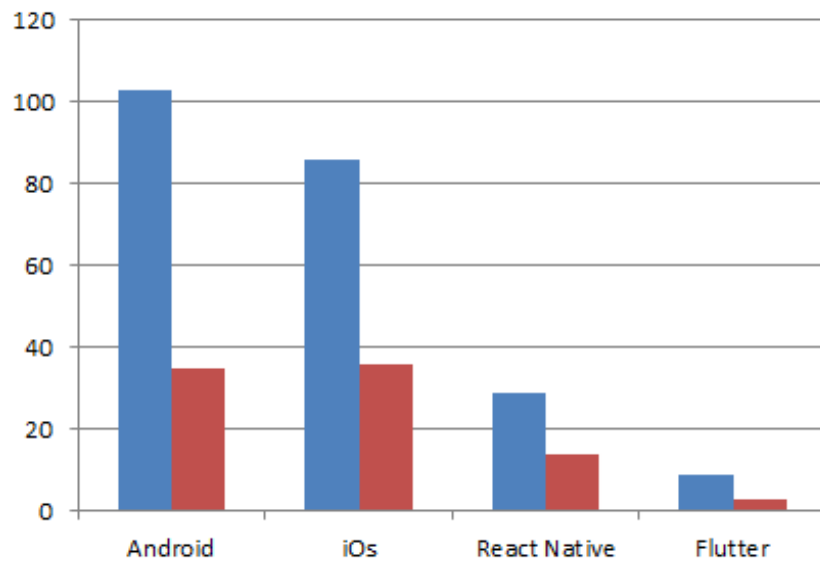
OS / technology	office	remote
Android	103	35
iOS	86	36
React Native	29	14
Flutter	9	3

Each project has its own technology requirements and is unique. Also, each developer has their own skills and preferences. But, if you are a beginner developer, then you should start learning from native development, then try cross-platform.

## References

[1] Introduction to Expo, 2020. URL: <https://docs.expo.io>.





**Figure 8:** Job statistics on grc.ua

- [2] Google Trends, 2020. URL: <https://trends.google.com/trends/?geo=US>.
- [3] A. Polomodov, Mobile development: native, cross-platform, hybrid, web ..., 2020. URL: <https://apolomodov.medium.com/%D0%BC%D0%BE%D0%B1%D0%B8%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F-%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0-native-cross-platform-hybrid-web-178c355c174>.
- [4] N. Priyatselyuk, Native or cross-platform - what should a novice mobile developer choose? experts answer, 2019. URL: <https://tproger.ru/experts/native-or-crossplatform/>.
- [5] Work.ua, Work.ua is Ukraine's #1 job search site, 2021. URL: <https://www.work.ua/en/?setlp=en>.
- [6] robota.ua, Work in Ukraine. Employment and job search on robota.ua, 2020. URL: <https://rabota.ua>.
- [7] DOU.ua, Community of programmers, 2021. URL: <https://dou.ua>.
- [8] HeadHunter Group of Companies, Work in Ukraine, personnel search and job posting - grc.ua, 2021. URL: <https://grc.ua>.