

Technology Cross-Platform Approaches for Native and Hybrid to Mobile Application Development

Devesh Goswami¹

Vikash kumar²

Yash panday³

Dr. Ritesh Rastogi⁴

Noida Institute of Engineering and Technology, Gr. Noida^{1,2,3,4}

Abstract:- One of the most important marketing software today is mobile marketing. In this article, we examine the differences between two approaches to mobile development: (i) native (custom development for each platform) and (ii) cross-platform (when standard code is shared across platforms). Experienced teams, students, and all new to mobile development will have a hard time choosing between methods and techniques. Our aim was to compare these methods from a student-to-professional perspective in 166 participants from 3 different studies and explore factors that might influence their selection. We focus not only on dimensions, but also on quality needs and ideas (e.g. project needs, final product line and support platform) that may influence this decision. The results can help students and inexperienced groups see clearly the best way to develop mobile in a given context and guide them in making choices. Through our research, we have found that even if the amount indicates a combination, it may not be the best decision due to other advantages and ideas. We identified two non-technological factors that could influence the selection of the best method: previous team experience with the technology chosen; and technology development and integration. With this in mind, we recommend creating a map of the project's needs and the advantages/disadvantages of each.

I. INTRODUCTION

In 2017, Stack Overflow, a popular and respected website for developers, published the results of its global survey of 64,000 developers worldwide. According to the survey, 23% of respondents are mobile developers. Of this group, 64% use Android, 57% use iOS, 4.3% use Windows Phone and 0.7% use Blackberry [22].

These data show the number of mobile phones sold to end users worldwide as 325394.4 units using Android, 71525.9 units using iOS, 4395.0 units using Windows and 906.9 units using the BlackBerry platform [24].

Therefore, developers and companies that want to get more users for their apps will focus on the Android and iOS environment.

An application that needs to run on multiple platforms has different source code on each platform [4]. Platform-specific applications that can directly access the hardware

are called native applications [11]. Developing native applications for multiple platforms is difficult and potentially costly because source code has to be copied across multiple platforms, which can lead to significant costs [19]. In response, tools have emerged that allow developers to use an application and port it to multiple platforms.

Such applications are called cross-platform, and their main benefit is to simplify the life of the application through centralized development in a single codebase [4]. Despite this advantage, the cross-platform approach also has disadvantages such as hardware access and high memory usage [15].

The game development industry is an example where software development requires multiple platforms.

The most used and mature systems such as Unity and Unreal allow cross-development not only on mobile platforms, but even on consoles, desktop operating systems, smart TVs and web browsers. In 2016, about 59% of mobile games used cross-platform as a game engine [23].

Different races and systems have different characteristics and should consider these before choosing the most appropriate method. It can be difficult to decide which is best for a project due to different environments, technologies and needs. Evidence of these factors is essential to determine the process and process of all ideas and technologies.

Our goal is to provide such evidence through research, not to process technology when choosing it, but to write good content. previous studies compared quality and quantity using one of these models [15] [14]. We compare race and cross-development from a student's perspective. We also confirm these points from the point of view of professional developers. We researched and divided our target into three groups: beginners (college students with different levels of application), intermediate (college students in graduate programs), and advanced (experts from different mobile software development communities).

After reviewing our findings, we hypothesized that multiple features might not be the most important thing to consider when choosing a race or cross-platform.

The remainder of this article is organized as follows. Part 2 discusses the history and process of mobile development. Step 3 shows us some important activities to earn money. Chapter 4 presents the research design of this study and discusses our research questions and methodology.

Next, Chapter 5 describes our results from research, particularly by students and mobile software developers.

Finally, the following section: Results summarizes the article, highlights its main conclusions, and provides guidelines for future work.

II. BACKGROUND

Due to its hardware features, software distribution and release cycles, mobile apps are unique in the development process. Developers should address issues related to storage, phone features, mobility, user experience, security and privacy [11]. Mobile apps are generally robust, released in small cycles and require little storage and are cost-effective for end users [9].

➤ *Native Applications*

It is designed for a specific operating system (OS) such as iOS or Android, meaning there is no mobility between platforms. Typically, these apps are available in digital stores such as; App Store for iOS and Play Store for Android), responsible for downloading applications to devices and making them available to end users [18]. Native apps tend to follow the interface (look and feel) and technology standards supported by a particular platform. These applications can directly access assets such as sensors, cameras, GPS (Global Positioning System) [11]. This approach provides users with high performance and professional UI/UX [18]. However, it requires a specific framework and SDK [15].

➤ *Hybrid Applications*

Another way to develop applications for multiple platforms is to use cross-platform technology. This recommendation results in a codebase that allows apps to be deployed across multiple platforms. Two different ways of using the technology are covered: web applications and hybrid applications. Web applications are applications designed to use web technology as a mobile device. The device must be accessed via a browser (eg.

For example, Firefox, Safari, Chrome). They provide integration across multiple mobile platforms [18]. The technology follows the client-server model, in which the source code runs remotely. However, it uses a modified front-end tool that sees limited mobile devices and fixes the UI.

Integrating ethnic and web-based approaches. This approach allows developers to use web technologies to target multiple mobile devices and build similar applications to traditional applications from a single source of code. Hybrid apps are built into internal devices using the WebKit

rendering engine, which is based on the operating system browser and can be found in Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux[2]. It improves the user experience in the app while using traditional products [1]. In this article, we define a hybrid application as a web application built primarily using HTML5 and Javascript.

Native containers wrap web applications and provide access to platform functionality [20]. according to this

Ciman [8], when using hybrid technologies, battery consumption increases by more than 250% compared to equivalent paper. Additionally, according to Rakesh [20], hybrid apps generally consume more RAM than traditional apps, render animations less fluidly, and often appear slower when page and state change.

Deciding on native, web application and hybrid approaches requires careful analysis of the technical and non-technical aspects of the project. However, they are not of equal weight as they depend on many external factors such as the development and implementation of the target operation, the need for software and the various platforms required.

III. RELATED RESEARCH WORK

We review projects related to mobile development. Over the past 13 years, we have evaluated significant discussions, research, and the development of new methods to improve mobile development [13]. We are always looking for alternatives that reduce cost and development time with the least possible impact on the functionality, UI/UX, and other aspects of the application's security. The aims and methods of some of these studies are similar to our study. We compare some of the methods adopted by other researchers with ours.

Charan et al. [7] Provide a theoretical comparison of cross-platform and native app development in terms of UI coding, performance on each platform, and recommended usage. While dealing with cross-development, they came to the conclusion that hybrid solutions have many advantages over traditional solutions. We share the same passion, especially business (like the requirements of the project).

Prezotto [21] makes a distinction between native, web application and hybrid approaches. Authors provide a proof of concept by creating a hybrid app. After solving the problem, Prezotto found different behavior on the platforms the app was running on. We see the problem of recognizing the potential of hybrid applications before traditional methods. In our survey, we aim to illustrate this difference by analyzing the perspectives of beginner and advanced developers.

Dalmasso [10] introduced various commercially available hybrid application development tools (e.g. PhoneGap, Titanium, Rhomobile and JQuery Mobile) and compared them through analytics testing. Dalmasso's work

discusses qualitative research on these trends to provide business developers a quick summary. Our research aims not only to cover cross-platform methods, but also to compare them more empirically with native methods.

Majchrzak and Grønli [16] discuss how to choose between different skills. The author uses Ionic, React, and Fuse as research objects and mostly thinks only about functionality in terms of quantity. Recently, Biørn-Hansen et al. [5], defined the definition of the main content and content record and created a taxonomy for the development of the application interface. The authors argue that developers are faced with many cross-development methods to choose from, and there is no one-size-fits-all tool.

Other work by Biørn-Hansen and Majchrzak [6,17] shows that Progressive Web Applications are the best way for cross-platform development. In our research, we discuss non-strategic issues when determining the best strategy for a project. We also discussed how to choose a mobile development method, but we did not consider the individual technologies used for this decision.

Latif [15] presents a case study of how various combinations work (eg., language, architecture, and performance) and highlight their strengths and challenges. A similar study was conducted by Gaouar [14], in which the authors divided the cross-platform rules into six groups: supporting platforms, data, security, access to traditional resources, liking, and resource usage. Unlike these authors, we do not focus on teaching the specifics of any method or method, but instead consider the content that the student expresses and performs. In our research, we compared different and traditional builds not only for various SDK platforms, but for all operating systems.

Carlstrom [3] did a study that came close to our goal in an attempt to separate the results for each of the indicated uses.

In general, the development time for cross-platform applications is shorter than for normal development. Shan Jiang [12] discussed the technical, empirical and practical solutions of different problems. The authors conclude that there is no universal solution that can be called the end of mobile app development when there is no big difference between the development method and the product.

IV. DISCUSSION

In this section, some insights and conclusions from the collected data will be discussed. Our results show that "how good and technology integrated" is the most important thing to consider when choosing a mobile development plan. This means that developers often have to check the quality and usability of information technology (eg code files, wikis, forums) as well as their support and community in the project environment.

This is directly related to the security and integration of mobile technology in the developer community. The development team's past practices and the learning curve with competing technology can influence this decision. Research shows that students intuitively value the same things as professionals with slightly different values.

Contrary to what the literature and recent studies suggest, we hope that the combination is not the best option in most cases. This statement seems to be true, although several hybrid technologies are already similar.

There is always a preference, as the app needs UI/UX functionality, not video game design. Since resources are always needed (which is rare for other devices), it is better to choose the same option.

From the collected data, we observed that there is no best option for every situation and that every project and every software has its own characteristics. Based on the results of the data analysis, we provide general recommendations shown in Table 1 and described below.

➤ *Native Approach*

Publish the best UI/UX graphics and make complex animations easy to use. It is best to follow the design process and standards of the chosen platform (Android or iOS). Most traditional operating systems are best served by providing extensive knowledge, technical support, and a large community of developers with projects already in progress. For social apps, vendors have a lot of good builds. With proper use of these features, they can be optimized for low battery usage, RAM usage, video memory and latency.

We recommend traditional development methods for:

- *High performance projects;*
- *Tasks requiring support from technology vendors; - projects require faster learning for inexperienced developers.*

➤ *Hybrid Approach*

Render accepts UI and UX compatibility for all content platforms in graphics. However, it shows more difficulty using new (non-standard) or more animations and graphics. This approach works well in terms of cost effectiveness in terms of budget and time ahead of the platforms that can be used. This is because its main advantage is code reuse across different platforms. Even features that use certain resources still require platform-specific code.

Not all hybrid technologies provide a complete development environment (eg IDE, SDK, CLI). The currently most widely used hybrid approach also has good and extensive documentation, a wiki, and support from its community. The final authoritative code or binary implementation is not suitable for a single platform, causing more stress for the product to equal its native equivalent.

We recommend a combination for:

- Targeted campaigns on two or more platforms;
- Projects that require access to basic equipment;
- The job does not need a top view or follow up;
- With the financial restrictions project.

Table 1 Recommended Criteria to Choose between Different Approaches in Mobile Application Development.

	Native	Hybrid	Web App
UI/UX performance	High	Low	Low
UI/UX customization	High	Medium to low	Relative to web technology adopted
UI/UX platform guideline compliance	Maximum	Medium to high	none
Documentation, support, community	High	Low to Medium (relative to the selected hybrid technology)	Relative to web technology adopted But low in the context of web app
Hardware stress (battery consumption, RAM, GPU, response time)	Low	Medium to High(very perceptive in some cases)	High to Maximum
Optimized product output (binary or generated source)	Can be very optimized (relative to developer)	Not optimized	Not optimized
Relative cost per number of platforms.	Maximum cost per platform.	medium or low (relative to the selected hybrid technology)	Minimum cost per platform
Other comments	Easy to set up the working environment, a high number of developer tools; maximum optimization for game software.	Usually longer to set up the working environment (with exceptions);	Limited access to the device resources
Recommended for	For projects with a lower priority in the budget, projects that do not need to be deployed on multiple platforms, apps that need high or maximum performance are required as well as a need for proper support from the technology providers and needs a faster learning curve in the development team.	For projects targets, two or more platforms, require access to primary device resources, projects that do not require excellence in graphical performance or compliance and have a high priority of a lower budget.	For apps that have simple UI/UX requirements do not need to be stored in the device, apps that need to run in an undetermined number of devices and applications that do not require device-specific features (camera, accelerometer, touch-id, gyroscope, compass, proximity, microphone)

➤ *Web App Approach*

It creates at least UI/UX according to its framework that does not follow certain criteria. This approach is ideal for using data that already exists in a production environment with simple user interfaces (for example, CRUD-based applications). This approach has the best level of code reuse. Corresponding to its higher relationships is limited access to material. We recommend this model for:

- Projects with simple UI / UX requirements;
- Projects that do not need to be completely stored on the device;
- Projects that need to be run on an indefinite number of devices;

- *Not required Special items for device capabilities (e.g. camera, accelerometer-ether, touch ID, gyroscope, compass, proximity, microphone)*

V. CONCLUSION

In this article, we consider choosing better products; better; higher; strong; Develop a better way to build a mobile app, whether native or cross-platform">. Our discoveries can bolster understudies and teachers to systematize the choice of the foremost suitable system in a versatile extent amid an undergrad or graduate course related to the program building field as well. To reply to the central investigation address, we conducted overviews with 166 members from three bunches: experienced, halfway, and master versatile engineers. From the information investigation, we concluded that professionals, no matter their experience, regard both specialized and nontechnical prerequisites such as the bolster advertised by the innovation (documentation, wiki, and community), the execution of the ultimate item, venture prerequisites, and learning bend.

When it comes to specialized viewpoints, we examined the preferences and challenges of the local, crossover, and web app portable approach. We performed an experiment with middle of the road designers where they actualized highlights in both local Java Android and Ionic and afterward, compared their involvement. Understudies came to comparative conclusions of specialists, where we concluded that professionals are emphatically slanted to select a local approach when it comes to accomplishing tall execution (i.e., moo battery utilization, moo idleness, tall graphical execution, liveliness, moo memory utilization). Be that as it may, in the event that execution is not

With very little hardware/device required and working, a hybrid solution is considered fine; better; higher; strong; To create portable applications for different platforms "> More. Based on these facts, we suggested a set of specialized and nontechnical perspectives to assist newcomers of versatile advancement to systematize the choice of the portable computer program advancement procedure for their ventures. We ought to stretch that we found no proof that there's an ideal versatile approach choice, and it is unequivocally impacted by both specialized and nontechnical components displayed in these papers, Also by installing the application.

➤ *Subsequent Work*

As a future study, we need to complete our survey and collect data from manufacturers in other countries to confirm our findings from different sources. Another ponder ought to perform a quantitative examination comparing both local and cross-platform versatile approaches to approve the come about. We too recommend utilizing the profiling apparatuses to have a clear understanding of the execution distinction between executions of the same include in local and cross-platform applications.

ACKNOWLEDGEMENTS

The findings of this research can help students and teachers choose the most appropriate way to build mobile apps, regardless of race or platform. This article provides professional techniques and practical tips to help novices choose a mobile app development strategy for their project.

The authors stated that mobile phone use is not an optimal choice and was influenced by the methods and negative methods presented in this paper, including the content of the application. Therefore, this article serves as an important guide to help developers make informed decisions when choosing mobile development.

REFERENCES

- [1]. Bernardes, T.F., Miyake, M.Y.: Cross-platform Mobile Development Approaches: A Systematic Review. *IEEE Latin America Transactions* **14**(4), 1892–1898 (Apr 2016)
- [2]. Charland, A., Leroux, B.: Mobile Application Development: Web vs. Native. *Commun. ACM* **54**(5), 49–53 (2011)
- [3]. Ciman, M., Gaggi, O.: An empirical analysis of energy consumption of cross-platform frameworks for mobile development. *Pervasive and Mobile Computing* **39**, 214–230 (2017)
- [4]. Corral, L., Janes, A., Remencius, T.: Potential Advantages and Disadvantages of Multiplatform Development Frameworks - A Vision on Mobile Environments. *Procedia Computer Science* **10**, 1202–1207 (Jan 2012)
- [5]. Dalmasso, I., Datta, S.K., Bonnet, C., Nikaein, N.: Survey, comparison and evaluation of cross platform mobile application development tools. In: 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC). pp. 323–328 (July 2013)
- [6]. El-Kassas, W.S., Abdullah, B.A., Yousef, A.H., Wahba, A.M.: Taxonomy of Cross-Platform Mobile Applications Development Approaches. *Ain Shams Engineering Journal* (2015), <http://www.sciencedirect.com/science/article/pii/S2090447915001276>
- [7]. JIANG, S.: Comparison of native, cross-platform and hyper mobile development tools approaches for ios and android mobile applications. University of Gothenburg (2016)
- [8]. Johnson, H., Blum, S., Tyburski, J., Lummus, A., Martin, D., Mendez, M., Pati-saul, C., Hurewitz, K.: Methods and systems for providing platform-independent shared software components for mobile devices (Jan 10 2006), <https://www.google.com/patents/US6986148>, uS Patent 6,986,148
- [9]. Lamia Gaouar, Abdelkrim Benamar, F.T.B.: Desirable requirements of cross platform mobile development tools. *Electronic Devices* (2016)

- [10]. Latif, M., Lakhrissi, Y., Nfaoui, E.H., Es-Sbai, N.: Cross platform approach for mobile application development: A survey. In: 2016 International Conference on Information Technology for Organizations Development (IT4OD). pp. 1–5 (March 2016)
- [11]. Majchrzak, T., Grønli, T.M.: Comprehensive analysis of innovative cross-platform app development frameworks. In: Proceedings of the 50th Hawaii International Conference on System Sciences (2017)
- [12]. Majchrzak, T.A., Bjørn-Hansen, A., Grønli, T.M.: Progressive web apps: the definite approach to cross-platform development? In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018)
- [13]. Malavolta, I.: Beyond native apps: web technologies to the rescue! (keynote). In: Proceedings of the 1st International Workshop on Mobile Development, Mobile!SPLASH 2016, Amsterdam, Netherlands, October 31, 2016. pp. 1–2 (2016)
- [14]. Martinez, M., Lecomte, S.: Towards the Quality Improvement of Cross-Platform Mobile Applications. In: 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft). pp. 184–188 (May 2017)
- [15]. P.K, R., M.Kannan: Online mobile application development using ionic framework for educational institutions. In: International Journal of Advanced Research Methodology in Engineering & Technology. vol. 1 (May 2017)
- [16]. Prezotto, E.D., Boniati, B.B.: Estudo de frameworks multiplataforma para desenvolvimento de aplicações mobile híbridas. Universidade Federal de Santa Maria, Trabalho de Conclusão de Curso (2014)
- [17]. StackOverflow: Developer survey results 2017 (2017), <https://insights.stackoverflow.com/survey/2017>
- [18]. Unity: Cross-platform game engines (2018), <https://unity3d.com/public-relations>
- [19]. Woods, V., van der Meulen, R.: Gartner says worldwide smartphone sales grew 9.7 percent in fourth quarter of 2015 (2016), <http://www.gartner.com/newsroom/id/3215217>