

# Hybrid Mobile Apps Quality, Speed to Market, and Adoption: A Look At React Native

Ryan Turner

May 2018

Traditionally, mobile application development is dominated by the languages and APIs provided by device manufacturers. Hybrid mobile application development has existed for nearly a decade (Taft 2009), however new advancements have been made in ease of development and performance with the introduction of Ionic and React Native. There are few academic studies on these more modern hybrid frameworks, and the frameworks are not widely taught in school. React Native is significantly different in the how development happens, its performance, and its feature set compared with the existing researched platforms. Established techniques will be repeated with React Native to offer a comparison. The rest of this writing will focus first on the existing research, why React Native is significantly different and demands the research be repeated, and end with the tasks that will take place in order to answer these questions.

## Existing Research on Hybrid Mobile App Development

There are many conference and journal papers today that document the hybrid web app approach that Phonegap introduced (Que, Guo, and Zhu 2016, Huynh, Ghimire, and Truong (2017), Latif et al. (2017), Jobe (2013)) and Cordova has continued. These materials introduce the general concepts of mobile development, app development, and hybrid app development. Then, discussion and benchmarking on ease of coding, developer tools, capabilities, and performance takes place. Some materials also include other interpreted mobile frameworks like Xa-

marin and CodeNameOne for similar analyses, but fewer exist for these as they are commercialized solutions. These studies exist in the older areas of hybrid frameworks, but not with the new industry-leading framework React Native.

The prevalence of hybrid apps in the app stores has also been researched (Martin et al. 2017, Ali, Joorabchi, and Mesbah (2017), Viennot, Garcia, and Nieh (2014)) extensively. This research also reviews app listing attributes like category, content rating, and size. These same approaches have not been used to find React Native applications. Many publications however do introduce tools and techniques that can be extended to identify other frameworks. Additionally, some research combines this data with manual analysis of blogs and developer surveys to determine prevalence.

## Distinguishing React Native from Other Hybrid Frameworks

The motivation for this work is that React Native is not covered in research like other hybrid frameworks, and it has significantly distinguished capabilities from others. Performance, potential for cross-channel reuse, and integration with UX/UI and testing tools all are areas in which React Native has advantages over researched platforms. In this research, the author hopes that academics consider React Native for mobile development courses and use in projects.

## Research Approach

In order to conduct this research, the goal-question-metric approach will be utilized. The goal of this research is as follows: *inform an academic audience of the traits and distinctions of React Native mobile apps from the learning engineer's perspective.*

Consistent with that goal, the driving question asked is:

**How do developers learn the React Native framework, and how is it perceived by its developers?**

**RQ1: Do engineers find learning React Native development simpler than learning traditional native application development?**

**RQ2: Where do developers learn React Native development?**

**RQ3: How widely is React Native used in undergraduate university studies and projects?**

The specific technique to research these questions at this point are using the following tasks:

- Fetch apps associated with universities, reverse engineer them to find use of React Native; a few open source platforms exist for this activity, however the specific approach to identify React Native's presence needs to be made
- Survey developer community for comparison of React Native to traditional dev; compare these results against existing research
- Survey developer community for how they learned React Native

## References

Ali, Mohamed, Mona Erfani Joorabchi, and Ali Mesbah. 2017. "Same App, Different App Stores: A Comparative Study." In *Proceedings of the 4th International Conference on Mobile Software Engineering and Systems*, 79–90. MOBILESoft '17. Piscataway, NJ, USA: IEEE Press. <https://doi.org/10.1109/MOBILESoft.2017.3>.

Huynh, Minh, Prashant Ghimire, and Donny Truong. 2017. "Hybrid App Approach: Could It Mark the End of Native App Domination?" *Issues in Informing Science and Information Technology* 14:049–065.

Jobe, William. 2013. "Native Apps Vs. Mobile Web Apps." *International Journal of Interactive Mobile Technologies (iJIM)* 7 (4):27–32. <https://doi.org/10.3991/ijim.v7i4.3226>.

Latif, M., Y. Lakhrissi, E. H. Nfaoui, and N. Es-Sbai. 2017. "Review of Mobile Cross Platform and Research Orientations." In *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 1–4. <https://doi.org/10.1109/WITS.2017.7934674>.

Martin, W., F. Sarro, Y. Jia, Y. Zhang, and M. Harman. 2017. "A Survey of App Store Analysis for Software Engineering." *IEEE Transactions on Software Engineering* 43 (9):817–47. <https://doi.org/10.1109/TSE.2016.2630689>.

Que, P., X. Guo, and M. Zhu. 2016. "A Comprehensive Comparison Between Hybrid and Native App Paradigms." In *2016 8th International Conference on Computational Intelligence and Communication Networks (Cicn)*, 611–14. <https://doi.org/10.1109/CICN.2016.125>.

Taft, Darryl K. 2009. "PhoneGap Simplifies iPhone, Android, Blackberry Development." 2009. <http://www.eweek.com/development/phonegap-simplifies-iphone-android-blackberry-development>.

Viennot, Nicolas, Edward Garcia, and Jason Nieh. 2014. "A Measurement Study of Google Play." *SIG-METRICS Perform. Eval. Rev.* 42 (1). New York, NY, USA: ACM:221–33. <https://doi.org/10.1145/2637364.2592003>.