

3.2 Sources of Data Collection

Primary and secondary sources were also used to gather information for this research project. The data collection instruments issued for this study were Internet, books, related literature review, seminar documentations, online developers forums and highly rated organisation that provides statistics of usage, performance, users of various technologies (statista.com, github.io, iee.org)

3.4 Eligibility Criteria

Every research study has guidelines for who can or cannot participate and what can or cannot be included in the study.

Enrolling inclusion and exclusion criteria ensure that the results will be due to what is under study and not other factors. In this way, eligibility criteria help researchers achieve accurate and meaningful results.

Inclusion Criteria

Language: English

Timeframe: 2014-2020

Title: Is the title relevant to the study?

- i. Yes, review Abstract, Questions, Keywords and Conclusion. Does it answer any of our questions, or contain relevant information.
- ii. No, exclude the paper

Exclusion Criteria

Non-peer reviewed

We began the literature study with a starting set of six different papers. These papers were defined as relevant and useful using the Inclusion and Exclusion criteria.

In the article “Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering”, Claes Wohlin described what the characteristics for a good start set is. In this study we chose to interpret these characteristics as:

- i. If the relevant papers may come from different communities, then it is important to have these covered in the start set. The reason is that the papers may be in independent clusters, i.e. in clusters of papers not referring to each other.
- ii. It is important to get the right amount of papers in the start set. The number differs depending on the area of study, in a smaller, more focused area the amount requires fewer papers than a bigger area of study.
- iii. If the search result of papers is too large, for example due to having to general search terms, then an alternative is to identify a number of relevant and highly cited papers.
- iv. The start set should include different publishers, years and authors, i.e. diversity.
- v. The start set ought to be formulated from keywords in the research questions, preferably also take synonyms into account.

4. ANALYSIS OF RESULTS AND FINDINGS

This section contains the research perspectives and analysis of result which include the comparison of the two development approaches and more details based on the newest technologies and updates in these two approaches. The section discusses the native application development and hybrid application development based on the latest updates and upgrades which former publications covers less or non, giving detailed explanation, features, and limitation of these two development platforms and then gives comparison of the two platforms and finally providing findings from the research work.

4.1 Native Application Development

‘Native’ is a term used for software development in which the developer uses the main language, tools, and a framework for the platform being targeted, while using an Integrated Development Environment (IDE).

Native apps are typically built using development tools and languages (XCode and Objective-C for iOS apps, Eclipse, Android Studio; Java for Android, Visual Studio; and C# for Windows) that the respective platforms support, and they run only on those platforms. Since native apps are written for specific platforms, they can interact with and take advantage of operating system features and the other software programs installed on the platforms.

4.1.2 Features of Native Application Development

- i. Multi-touch – double taps, pinch-spread, and other compound User Interface (UI) gestures
- ii. Fast graphics API – extremely speedy graphics
- iii. Fluid animation – crucial in gaming, highly interactive reporting, or intensely computational algorithms for transforming photos and sounds
- iv. Interaction – interacts with other apps and provides for widgets on the homepage, also, can respond to hard keys, i.e. the Android’s search button and volume control
- v. Documentation – there are nearly 3,000 books on iOS and Android development, along with several online articles, blog posts, and technical threads

As Reviewed from existing works in chapter two the limitation of native application development did not change even with the inclusion of the latest feature as compared to the hybrid approach. Despite regular updates, native applications development for various devices has not changed from previous approach but rather makes the existing functionality do better and adaptable to new technologies and approaches, this makes the system more reliable and maintains its existing advantages and, in some cases, creates more and the downfall, its limitations almost the same.

4.2 Hybrid Application Development

Hybrid application development empowers the developer to create an application using a single language or tool set, and instantly deploy it across a variety of platforms. In general, any program that can run on more than one device with different systems is a cross-platform program and hybrid applications is one approach of development cross-platform application, leaving web approach as the other.

4.2.1 Hybrid Application Development Technology

As reviewed from previous works in chapter two hybrid application development technology is still evolving and one of the types of hybrid approach which previous work state as “still in development phase” is now fully functional and has change the face of hybrid application development. This approach is explained as follows;

4.2.1.1. Cross-Compiled Hybrid Approach

In the cross-compiled approach (or generated approach), developers write codes with the use of any common programming language. These codes are transformed by cross compilers to a specific native code Fig.5.

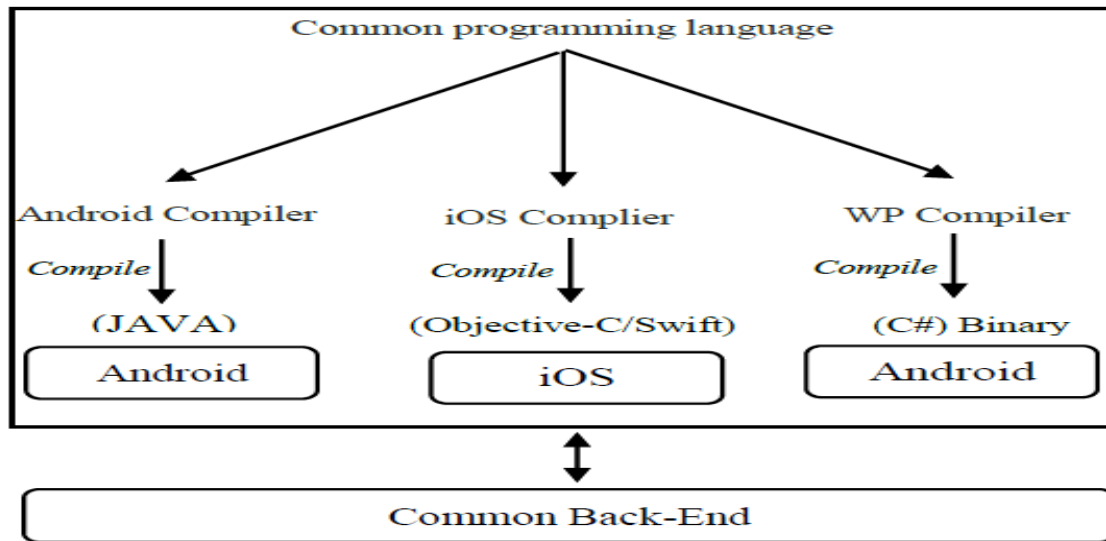


Fig.5: Cross-compiled Approach

The main benefit of this approach is that the applications are able to attain native performance and deliver all the features of native applications along with its native interface components.

There exist two powerful platforms based on this approach; the first one is Flutter which uses Dart shared codebase, the second is Xamarin, which uses C# shared codebase and lastly React-Native. Developers can use Xamarin tools to write native Android, iOS, and Windows apps with a shared code across a number of platforms and native user interfaces. However, Xamarin requires writing a specific code in order to benefit from the platform specific features; contrary to the Flutter and React that also uses Code-Name. Code-Name presents more advantages by having a free and open source version, which is more generous than the free Xamarin version (Nfaoui, Es-Sbai and Abdellah, 2019).

4.2.2 Features Hybrid Application Development

- i. Familiar languages/relatively simpler language – apps can be created with HTML, CSS, and JavaScript. Simpler language C#, Dart.
- ii. Integration – integrates with existing JSP and .NET infrastructure
- iii. Single code for building mobile apps for iPhone and Android platforms
- iv. Advanced capabilities – leverages features such as GPS, camera, etc.
- v. Flexible – applications adapt to different resolutions, screen sizes, aspect ratios, and orientations
- vi. Multiple devices – can be used to build for desktop, tablet, and mobile web devices
- vii. Single-page architecture – generates self-contained web apps that execute locally on the device.
- viii. Hybrid application simplifies marketing by enabling the use of multiple media with generalized messages targeting potential customers.

4.2.3 Limitations Hybrid Application Development

- i. While cost saving can be one of the advantages of cross-platforms, the real meaning behind the term should be fully understood. A typical mobile application development process has cost overheads related to requirements gathering, analysis, and high-level design. It also has platform-specific design considerations, such as form factor, capabilities of operating systems, and hardware besides other similar concerns.

- ii. Pertaining to single code base, if a particular issue is found and fixed, or a new capability/feature on one platform is added, the entire suite of target applications should be retested fully. This is a serious concern. The same is applicable even if the change is required for only one platform. The fact that the code is used for all platforms introduces a mandatory overhead – i.e. to test on each and every platform every time a change is ready to be submitted. Any change for a particular platform may have unforeseen effects on an unrelated platform.

4.3 Comparisons

From the collection and analysis of existing data, the two key elements of our research are compared with each other, giving both the pros and cons of each element and discussing their major features in detail.

Consideration	Native	Cross-Platform
Multiple OS Support	No	Yes
User Interface Quality	High	Medium to High
Performance	High	High
Cost of Ownership	High	Medium
Application Updates	Native Market	Native Market
Application Maintenance	High	Medium
Development Languages	Java, C, C++, Objective C, Objective C++	Dart, Java,HTML,CSS, JavaScript,

Comparisons Between Native and Mobile Platforms

4.4 Results

After reviewing relevant publications, articles and research works and also performing an investigative research on the latest development in the areas that covers native and hybrid application development but the once that are published and those that are in use but too recent to have any research publication and also, comparing the two platforms by the publications and from the publications the following results were realised:

- i. Performance: with the introduction of the new base languages that are supported by some devices and also the introduction of the cross-compiler, the performance of hybrid application has skyrocketed to be as much as that of native.
- ii. UI: hybrid application now does not necessarily needs the web view in androids or the UI web view for iOS. It compiles to the native UI equivalent and render it on the device.
- iii. Access Control: Hybrid applications has now covered the problem of not having access to most functionality of device such as camera, GPS, Hardware Accelerator, OS. That is now covered, hybrid application development platforms such as flutter, React Native, Xamarin can now have access to functionality that native does.
- iv. Design: Hybrid application development platforms offers more flexibility and design properties providing for more dynamic and beautiful designs.
- v. Offline: hybrid applications now provide almost all functionality that native applications provides.

4.4 Findings

As time goes the disadvantages of hybrid application is becoming its advantages and hybrid application development is evolving rapidly and will continue to evolve providing more feature and functionality that covers both what native can do and even more.

Missing features of hybrid application such as; hardware access, Gps optimizations, OS interactive access, Encryption, Security, Utility access and optimization are now covered and are being done by hybrid platforms efficiently.

Hybrid application platforms are becoming closer and closer to native and with the introduction of the dart language by google and the flutter platform, hybrid applications produced are in all ways equivalent to native application and flutter having its own language and doesn't run on the web view of its target device and also having its own compilers means a big breakthrough for the hybrid community.

All a developer needs is to have a taste of hybrid development and falling in love with it is an unforeseen event that will come to pass. More and more communities and forums are being created and they keep getting bigger and more interesting and from developers' point of view, the moment flutter and react native breaks out the love for mobile application development in general has increased drastically.

Even though hybrid application development comes with all its ever increasing features from all angles of this research and previous ones, hybrid application development will not replace native application development but rather might even work hand in hand to provide more quality and advanced applications for mobile devices.

5. SUMMARY, RECOMMENDATION AND CONCLUSION

5.1 Summary of Result

As apps continue to play an important role in the business world, developers and organizations struggle to find the best development approach, but most of them realize that both approaches have their advantages and disadvantages.

There is likely to be further fragmentation of mobile devices and technologies, all of which will play a huge role in escalating costs and time frames. At the same time, they will be adding to the complexity of the development process. There will also be more issues related to security, integration, and upgradation. There could also be new distribution channels, wherein developers can market their apps directly to consumers, instead of going through app stores. Social media and its power will continue to increase, and its effect will be palpable in the future mobile space. Therefore, organizations should adopt a flexible approach, for which support, scalability, and integration become factors to consider. When that happens, decisions about the right platform or approach will fall into place.

5.2 Recommendation

As this study shows that Hybrid application is not just another way of developing mobile application but also provides efficient and quality features that native approach cannot provide and even more, we recommend the following;

- i. To companies, developers and students that want to develop or learn how to develop not just for hobby but also has a target and a customer base, adopting hybrid application development will increase quality, productivity and reduce cost.
- ii. Keeping up with latest technology and new upgrades is a key feature that every developer and organisation should have and this feature should be a key element for every hybrid application development.
- iii. Having knowledge of and using it, are two different things that can comfortably be separated in this instance. Due to fragmentation there are many approaches to solving a

mobile application development problem and having a good knowledge of them all is very important but using them all is not necessary and so, even if native has been around and is not doing bad at all for a while and hybrid is now doing better individual developers should try to learn more than one approach and not necessarily use all and organisations should be able to handle all approaches.

5.3 Conclusion

Although native apps benefit from an optimal integration into the respective mobile operating system and good developer support, the analysis showed that cross-platform approaches are a viable alternative. As soon as mobile apps must be developed for multiple platforms under tight budgets, with small developer teams, and in a short time frame, a cross-platform approach is necessary. However, these approaches are more than a second-best alternative. Developers might prefer using a cross-platform solution even in the absence of these constraints.

Cross-Platform apps constitute an ideal starting point for new developers and a focus for native developers, because they do not require advanced knowledge and enable developers to start implementing the app right away. Cross-Platform apps are a simple approach benefiting from very good support by devices on all platforms. Furthermore, they can be easily ported to other cross-platform approaches.

However, the results of our evaluation are only general guidelines that can be adapted and interpreted for each project individually. The Result can be used to support decisions, for example in semi-formal multi-criteria decision methods. Basic decision support can be obtained by weighing the criteria according to the requirements of a given project and calculating a weighted grade, carefully interpreted and analysed for sensitivity. The result might yield first insights on which solution best matches the requirements at hand.

REFERENCE

- Anmol K, Rashmi G, B. Sindhya, May 2015. "An Introduction to Hybrid Platform MobileApplication Development" Accessed 2019-12-08 <http://research.ijcaonline.org/volume118/number15/pxc3903463.pdf>
- B. Siegfried, "Enhanced Student Technology Support with Cross-Platform Mobile Apps," in 39th annual ACM Special Interest Group on University and College Computing Services, San Diego, California, 2017.
- Behrens, H.: Cross-Platform App Development for iPhone, Android & Co. (2010), <http://heikobehrens.net/2010/10/11/cross-platform-app-development-for-iphone-android-co-%E2%80%94-a-comparison-i-presented-at-mobiletechcon-2010/>
- Bernard Kohan and Joseph Montanez, January 2015. "Native vs Hybrid / PhoneGap App Development Comparison" Accessed 2019-11-03 <http://www.comentum.com/phonegap-vs-native-app-development.html>
- Claes Wohlin, "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering" Accessed 2019-11-18 <http://www.wohlin.eu/ease14.pdf>
- El Habib Nfaoui, Najia Es-Sbai and Sidi Mohamed Ben Abdellah Conference Paper, March 2019, Cross platform approach for mobile application development: A survey <https://www.researchgate.net/publication/304371091>
- Felix MohammadiKho'i and Jawed Jahid, April 2019. "Comparing Native and Hybrid Applications with focus on Features" Accessed 2019-12-08

- Henning Heitkötter, Sebastian Hanschke, and Tim A Majchrzak. “Comparing crossplatform development approaches for mobile applications.” In Proceedings 8th WEBIST, pages 299–311. SciTePress, April 2052.
- Iván Tactuk Mercado, NuthanMunaiah, and Andrew Meneely. The impact of cross-platform development approaches for mobile applications from the user’s perspective. In Proceedings of the International Workshop on App Market Analytics, WAMA 2016, pages 43–49, New York, NY, USA, 14 November 2016. ACM.
- Jeff Whatcott, 30 Nov 2011, “HTML5 and the Rise of Hybrid Applications”. Accessed 2019-12-18 <https://blog.brightcove.com/en/2011/11/html5-and-rise-hybrid-applications>
- JQuery Mobile Accessed 2019-12-18 <https://jquerymobile.com/>
- L’uboštaráček and Valentino vranić. 2017 “MDA Based Multiplatform Mobile Application Modeling with Platform Compliant User Interfaces”.
- Luis Corral, Andrea Janes, and TadasRemencius. Potential advantages and disadvantages of multiplatform development Frameworks–A vision on mobile environments. In Procedia Computer Science, volume 10, pages 1202–1207. SciVerse ScienceDirect, 9 August 2012.
- Mohamed Ali and Ali Mesbah. Mining and characterizing hybrid apps. In Proceedings of the International Workshop on App Market Analytics, WAMA 2016, pages 50–56, New York, NY, USA, 2016. ACM.
- Native vs Hybrid / PhoneGap App Development Comparison, January 2015. Accessed 2016-04-03 <http://www.comentum.com/phonegap-vs-native-app-development.html>
- Paulo R, Adriano B, February 2015. “Cross Platform App A Comparative Study” Accessed 2019-12-08 http://uv3sv3ds3g.search.serialssolutions.com/?ctx_ver=Z39.882004&ctx_enc=info%3Aofi%2Fenc%3AUTF8&rft_id=info:sid/summon.serialssolutions.com&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft.genre=article&rft.atitle=CROSS+PLATFORM+APP+A+COMPARATIVE+STUDY&rft.jtitle=International+Journal+of+Computer+Science+%26+Information+Technology&rft.au=Paulo+R.+M.+de+Andrade&rft.au=Adriano+B.+Albuquerque&rft.date=20150201&rft.issn=09753826&rft.eissn=09753826&rft.volume=7&rft.issue=1&rft.spage=33&rft.epage=40&rft_id=info:doi/10.5121%2Fijcsit.2015.7104&rft.externalDBID=DA&rft.externalDocID=oai+doaj+org+article+0a54a17ec193467d9c4b66c4d98f1d9¶mdict=en-US
- Tim Majchrzak, Andreas Bjørn-Hansen, and Tor-Morten Grønli. Comprehensive analysis of innovative Cross-Platform app development frameworks. In Proceedings of the 50th Hawaii International Conference on System Sciences, pages 6162–6171. scholarspace.manoa.hawaii.edu, 2017.
- William Jobe, Nov 12, 2014, “Native Applications vs. Mobile Web Applications”. Accessed 2019-11-08. https://www.researchgate.net/profile/William_Jobe/publication/268153001_Native_Applications_Vs_Mobile_Web_Applications/links/546346b30cf2cb7e9da765c3.pdf