Cross-Platform Development Tools

Kishan Prajapati
Computer Science Department
Institute of Nirma technology
Ahmedabad 382 481, India
18bce181@nirmauni.ac.in

Rohan Mittal
Computer Science Department
Institute of Nirma technology
Ahmedabad 382 481, India
18bce198@nirmauni.ac.in

***Abstract*-This paper describes the solution of the problem of modern-day developer's cross-platform development. It discusses why we need cross platform development, advantages, disadvantages and challenges of using it along with pros, cons and verdict and few cross-platform development tools to know which one to use for what application.**

Keywords—Cross-Platform, Cross-Platform tools, Different Platforms

I. INTRODUCTION

Cross-platform development is the way toward creating programming items or administrations to make it usable for numerous stages or programming conditions. Designers and engineers utilize different strategies to teach diverse working frameworks and climate for one application.

The thought behind cross-platform development is, that a product or software application ought to have the option to function admirably in more than one explicit digital environment. This is done to sell programming for more than one restrictive working framework stage, for example, to oblige use on both Microsoft and Apple stages or on Android, Blackberry and iOS. With the rise of mobile technology, the investigation of open-source advances like Linux, cross-platform development is trending, rising and necessary to learn and apply [5].

The main objectives of this review paper are following:

- Understanding concept of the cross-platform development tools
- Difference between native and cross-platform development tools
- Selecting the most appropriate tool
- Advantages and disadvantages of cross-platform development tools
- Challenges in cross-platform development tools

II. DIFFERENCE BETWEEN NATIVE AND CROSS-PLATFORM DEVELOPMENT

Native vs. cross development is very hot debate and tech community has not still come to conclusion. Some are the in favors of native app whereas some support cross-platform development. Both native and cross-platform app development technology are in a steady kingdom of evolution. This converting nature of technology indicators that those subjects must be revisited and then to test which of those alternatives is presently main the game.

The difference between native and cross-platform development app is given below:

TABLE I. NATIVE AND CROSS-PLATFORM APP COMPARISON

Native Apps	Cross-platform Apps
It involves high cost for development	It includes lesser amount of cost for development
It codes on a single platform	It codes on a multiple platform, which is easy to use
Without any traffic, Platform SDK able to access to device's API	Not able to access all the device's API's
Consistent with the UI components of the device.	It has lower consistency with the UI Components
Flawless performance, when the app is developed for the devices OS	It has high performance however lags and hardware compatibility problems are common

III. SELECTING THE MOST APPROPRIATE TOOL

How do you choose the most suitable one?

As there are various choices to look over, it very well may be hard to limit to know which ones are the best.

The following four main factors should be considered while making the decision.

- Speed Considerations
- Feature Support
- Framework Risk
- Development Cost

Analyzing these elements is very crucial for cross platform development tools.

IV. DEVELOPMENT TOOLS

A. *Adobe PhoneGap*

Considered as the best Mobile App Development instrument, just as open source, Phone Gap is an incredible choices for building cross-stage versatile applications utilizing existing web innovations. Phone Gap is most appropriate for versatile applications that don't utilize the phone's local highlights. It packages our applications inside a local application that permits JavaScript to get to device-level APIs a similar way ordinary applications do [5].

It's based on the open source framework Apache Cordova that gives us access to the complete set of Phone Gap toolset.

Pros

- Creation of hybrid apps like HTML5, CSS3 and JavaScript are possible which are easy to learn using cross-platform development
- Access to native device APIs and it can be extended using modules because cross-platform follows plugin-able architecture [11].
- Open Source and thus, rapidly improving

- Compiles all work in the cloud so no need to maintain SDKs

Cons

- For intense graphic app it does not show good performance

Verdict

- Its tools are easy to learn and use.
- Hence, easy-to-use, open source and not using native features of the phone makes it useful for fast and effective app development.
- However, not recommended much for Graphic-Intensive apps.

B. Appcelerator Titanium

The Titanium SDK works utilizing JavaScript. Their framework makes it truly simple to make local applications for the two iOS and Android [3]. We can move from the plan perspectives back to the code parts with consistent effort [3].

Pros

- Rapid Prototyping-it allows us to reuse between 60% and 90% of the same code for all of our apps.
- An open-source development tool.
- Growing community.
- Web-oriented & great design features.
- Feature of drag and drop & ready-made templates.

Cons

- The application complexity increase is more than proportional as compared to increase in the development complexities (and costs)
- No Freemium
- Flexibility limitations-certain tasks that can be easily done with solo platform development tools (like XCode) creates difficulty in Titanium.
- Apps developed with it are quite lagging because it is obvious that apps developed with a native development environment provide the most smooth, fast and comfortable user experience.

Verdict

- As it allows us to reuse 60-90% of our code it is a huge time saver. Because as a developer we know it takes so much time to develop something from scratch.
- The capacity to get a head begin and have the greater part of the code previously done is the reason Appcelerator Titanium is such a well-known choice.

C. Cocos2d

When we need to construct an application that is for a two-dimensional game, we should consider utilizing Cocos2d as our cross-platform development tool [5].

There is five different platform available to develop when you use this tool:

- Cocos2d-x
- Cocos2d-JS
- Cocos2dXNA
- Cocos2d-Swift
- Cocos2d (Python)

The contrasts between these forks depend on your favored programming language as a designer. Along these lines, in case you're acquainted with XCode and Objective-C, you'll need to utilize Cocos2d-Swift.For those of you that favor the C# language, you should utilize Cocos2dXNA [3]. For JavaScript, Cocos2d-JS is the most ideal choice. In the event that you like the C++ language, at that point Cocos2d-x is your smartest option [3].

Some well- known gaming apps that were built using Cocos2d include:

- Hill Climb Racing
- Badland
- Kingdom Rush
- Castle Clash
- Tiny Village
- Big Fish Casino
- Matching with friends

Pros

- Open Source
- Consistently improved by other developers of the community
- High quality cross platform game development

Cons

- Little complex to use

Verdict

- The programming language flexibility on Cocos2d is the thing that makes it such a well-known choice for designers. Working with a language that you're familiar with guarantees that you're finishing things as quick and adequately as could be normal the situation being what it is.

D. Unity 3D

Unity3d is used for developing a game. It is not limited to 3D games, you can develop 2D games, AR/VR apps/games, animation & Arch viz Incredible Graphics is the reason why this cross-platform development tool is so popular. After you build up the code utilizing Boo, C#, or Unity Script, you have the capacity to send out the game to 17 stages. A portion of the best ones incorporate android, iOS, Windows, PlayStation, Linux, Xbox, Wii, Web.Unity3d has an organization of designers and specialists called Unity Connect [11]. On the off chance that you join this organization you discover help and get questions replied in case you're experiencing difficulty with something [11].

Pros

- Absolute high edge graphics
- Easy to learn
- Uses easy to learn programming language-C#
- App/game distribution to 17 platforms
- Can be used to distribute apps/games to all stores using Unity itself
- Can track analytics and social shares of the app/game
- Open source
- Growing community and improvement done by developers
- Easily available help and tutorials

Cons

- It has less graphics compatibility compared to its competitor Unreal Engine
- High quality features like high graphics rendering requires a paid version of unity which is quite expensive compared to Unreal Engine

Verdict

- Easy-to-learn, delivering high graphic games & apps make it one of the best choices for creating graphics-oriented apps as well as games for both, professionals as well as beginners.

E. Sencha

Sencha Touch 2 is a high performance HTML5 mobile application framework, and is the core of the Sencha HTML5 platform [14]. Sencha Touch 2 is one of the frameworks that allows developers to build fast and impressive apps that work on iOS, Android, Kindle Fire, Blackberry and more to provide an amazing experience. It's a complete solution which means that no more figuring out solutions from many different places [13]. They have a support model present in case one hits roadblocks. It's one of the future proof technologies because it's based on html5 and JavaScript [17].

Pros

- It is a one solution for everything because it has a vast array of charts, pre-built components and a data-grid based solution.
- Security of the software comes out of the box.
- It supports mobile as well as tablet.
- Theme customization is possible
- Due to support from the community, problems can be solved very easily.

Cons

- Sencha has huge learning curve compare to html5 and JavaScript because of more number of available framework.
- It's quite difficult to plug in a custom pure JavaScript component, one has to only use the components that are part of SDK.
- As a beginner, it's quite difficult to learn it from scratch.
- It has high licencing cost.
- If one wants to change any implementation of a component which is already defined, it is not an easy task, it needs a lot of knowledge.

Verdict

- Sencha can be used for enterprise applications that are really huge.
- It can be used when development is done under huge teams, on-shore and offshore model, one can divide and conquer by splitting it in different modules very easily

F. Qt

As it is written in C++, it has capacity to execute a wide scope of various highlights you may require in your foundation. These are produced for each stage by utilizing local innovation given on that stage. This implies that, taking a model, in the event that one needs to make a sound playback application, utilizing Media Player QML component will utilize Media Foundation on Windows while utilizing Streamer on Linux. Highlights and usefulness are actualized and isolated as modules. That gives you absolute control on the off chance that you need to make a Web Socket worker to build up a game running on any upheld stage to be utilized for your elite application. There is likewise an authority Qt for python for the people who have diverse language inclinations [1].

Pros

- It is the best designed C++ GUI application framework, as it is well designed.
- It has high user base which makes easy to get solution of problem.
- It is very stable on almost every major platform.
- Very nice documentation.
- Qt is full grown and has been considered by significant players.
- It isn't only a GUI system yet in addition gives a cross-platform method of doing a ton of stuff that occasionally desktop applications need to do [6].

Cons

- A bit of complexity in the building process is produced by Metaobject compiler.
- All the ongoing work is about the JavaScript/Qt Quick/QML stuff and not about the center library, seems like its designers are removing Qt might be from C++ [6].

Verdict

- Multi-platform, easy, powerful, lots of libraries and Extension.
- C++ Knowledge is mandatory.

G. Corona

Corona, developed by Corona Labs Inc. in 2009, is a complete development kit which lets software programmers

build 2D mobile applications for iOS, Android, desktop applications for Windows and OS X, Kindle and connected TV applications for Apple TV, Android and Fire TV [8].

Pros

- To cover a wide array of Android and iOS devices, it is quite easy to change the skin of the Corona simulator.
- Usually, it takes comparably much less code to implement a given functionality in Corona if the same is done with Java.
- Compared to the local Android approach, the system utilized by Corona for deciding on the good asset for a selected tool and its good efficiency to auto scale.
- It takes very minimal time to set up the development environment and is much less complex than installing Android's Eclipse/plugin environment/SDK.

Cons

- Corona has a feature to find memory leaks.
- If one wants to debug Corona applications, it is not easy in corona. Due to not so good IDE or debugger, simple issues are complex to solve.
- One needs to spend quite a large amount of money for the Pro version in order to get access to all the features. It is not over yet, Corona has a subscription model which is to be renewed every year.
- There is zero or negligible support for some simple elements of user interface. This limited support can sometimes lead to a non-standard or clunky user interface.

Verdict

- Corona is not recommended for a traditional business app, despite the fact that the Corona website claims it.
- It complicates widget-heavy user interfaces even though it efficiently simplifies graphics manipulation.

H. Alpha Anywhere

Alpha anywhere allows you to develop mobile applications which fit perfectly for business, specifically tailored to current workflows, mobile environments and many others. Business users and IT developers have a benefit to work simultaneously which allows them to build applications much faster and thus preventing bottleneck encounters that are so common during the typical process in a business app development. Thus, it accelerates processes, as an effect which enhances efficiency and productivity too with no compromise in the quality and the accuracy of work [10].

Pros

- Accelerated App Development Workflow due to no expert level coding requirement enables to complete process in few minutes instead of few days
- No or Low Code Development makes it so efficient, much easy and a very good experience for users to build an application.
- It is cost-efficient as the need of hiring application developers is eliminated.
- It is centralized as it focuses on the front end as well as back end development process.
- Rapid Mobile App Development aka. RMAD, as it enables faster development of apps and no need of writing and debugging lines too.

Cons

- There may be a need to neutralize the base Alpha page setup, if one has to use one's own CSS and Html within a component.
- To build quite a complex application, it needs a very advanced knowledge of JavaScript and XBasic.
- The annual cost that Alpha Software charges costs too much due to its subscription model.
- The development platform of alpha anywhere is a bit unstable and the work may lock up in between.
- Coding sometimes can become irritating or frustrating, as the platform uses a number of programming languages such as, HTML, Jason, JavaScript, and Basic.

Verdict

- Development done with this software is extremely fast.
- It is the perfect software for both the novice and seasoned pro.
- Support of this software is great and the online Forum community is so helpful and active.

I. SAP

SAP Mobile Technologies has provided us with a huge set of mobile capabilities to connect your end users and all your systems. To develop native or cross-platform applications, one can use our preferred tools or a choice from software development kits. Though choosing any approach, one gets benefit from backend integration, consistent and adaptive user experience and user/device on-boarding and management [16].

Pros

- It is a very productive tool to develop applications.
- Real-time data information is provided by them, which improves business processes.
- High performance is experienced by users due to a wide range of interfaces.
- One can get instant improvements which enhances user experience.

Cons

- The service should be more flexible in the way to accept to the needs of its users.
- Proper training is required in order to use the system, the system may confuse people.

Verdict

- SAP Cloud Platform is quite a good tool to develop applications of different platforms with a modern infrastructure.
- The reports provided by SAP are very comprehensive and good, the interfaces of SAP are much broader and have an extraordinary unique design
- In conclusion, this service provided by SAP is high performing and very comfortable.

J. Native script

Native Script is originally developed by Progress. Native Script apps can be developed using JavaScript, or any other language which can be transpired to JavaScript, such as Typescript. Native Script supports Vue JavaScript as well as the Angular frameworks. Mobile applications that are developed with NativeScript give the experience the same as if it were developed in XCode or Android Studio [4].

Pros

- NativeScript provides the user complete Native API access for better development.
- Native gives us better performance, if we compare things like ionic or phone gap as Script doesn't use any Web View to render the UI in the application.
- NativeScript provides a huge repository of plugins which helps the user to boost your development speed.
- NativeScript provides us comfort by giving us a choice of four different architectures: JavaScript, Angular, Typescript and Vue.

Cons

- When you change in Js/Ts file there is not hot reload in NativeScript unlike Google's Flutter.
- NativeScript runs each and everything on just a single thread, the UI thread which may be sometimes problematic. Thus, to prevent these, one has to use web workers.
- In the current version of NativeScript, when the user builds with webpack, there is no live sync.

Verdict

- Native scape is suggested for building native applications for iOS or Android when the engineers originate from a web development foundation and when the organization can't manage the cost of independent groups for Android and iOS application development and when the group is keen on utilizing Vue.js.

V. Advantage

A. Faster Time-to-Market and customization

'Compose once, run all over' is the idea that is accepted while building cross-platform application development.

Due to that, Time-to-Market for development of an app reduces drastically along with a quick deployment.

Also, it's simple for the engineers to do the minor changes in a solitary code, on the off chance that you have to change or modify the application. Thusly, it permits engineers to convey product more effectively and quick than the contenders by improving client commitment [12].

B. Larger Reach

This advantage is very obvious because the more platforms you cover, the more people in the world you'll be able to reach. In the period where Google's Android and Apple's iOS are going after top positions around the world, the enormous number of cell phone clients is expanding step by step. In this way, the benefit of taking advantage of more prominent market potential is picked up by building up an application those sudden spikes in demand for both iPhone and Android.

C. Reusable Code Components

Due to hybrid app or cross-platform development approaches, the development teams don't have to write code again and again for different platforms. Thus, the same code can be reused [13].

D. Reduced team size

It is easier to manage and to work with a small team when you're developing one application that works on all platforms. Further, with devices like PhoneGap and Appcelerator, it has been a serious simple assignment to oversee one group of developers chipping away at a solitary multi-platform application than a few groups taking a shot at various platforms.

E. Uniform Feel and Look

As the development of an application for every platform is done via a single piece of code, the look and UI are the same at every platform. It is difficult to sync two different developers in cross-platform when you are designing an app.

F. Use of Known Technologies

In the event that we utilize a few instruments, for tools like Appellatory, you can undoubtedly code in HTML5 and convert similar code for various mobile platform. This implies you're utilizing assets you definitely think about and you are just changing over them for its ability over various platforms [4].

G. Controls Cost

Due to this cross-platform mobile app development, companies today only have to invest once to get their app developed as compared to earlier days when companies had to spend heavily on different tools and technologies [7].

VI. Disadvantages of Developing for Multiple Platforms

A. Different Languages and Tools

As there are various tools accessible in the market like Unity, Grapple, Ramp, Open Plug, Rhombille, PhoneGap, Titanium that make the errand simple for you, if any application must be specially develops, it very well may be difficult to ward off the distinctions in devices and dialects of every platform's API.

B. User Interaction

Android and iPhone comparatively have much diverse screen layouts. Therefore, if one wants to design an app that fits on both these platforms, it can be quite a difficult task.

C. Platform Integration

With regards to incorporating the application with inclinations and notices applications, the nearby settings, genuine difficulty can be caused when attempting to handle different platform. Indeed, even capacity frameworks are unique, in this manner one might be taking a gander at cloud choices and furthermore coordination of outsider cloud service with your own application [14].

D. Limited User Experience

Cross-platform applications are not able to take complete advantage, that native apps are able to provide excellent user experiences. It is because of different platforms, screen layouts, and functionality [9].

E. Limited Tool Availability

Cross-compliance during the application development phase makes the code a bit complex and also reduces the speed drastically. Also, another problem is that it becomes mandatory for the application developers to make use of tools that are suited limited to a particular app.

F. Efficiency of the Program

Cross-platform development can make a program less proficient. For instance, it might need to store repetitive cycles or record stockpiling organizers diversely for various frameworks that it should uphold. It might likewise necessitate that the nature of a program is diminished to oblige less complex programming conditions [15].

VII. Challenges Fased in Cross-Platform Devdelopment

A. Switching the Platform

Most cross-platform system utilize their JavaScript subject and this make issues while utilizing the reusable codes. Consequently, finding the issue among the entire coding turns into a difficult task that at last improves time just as the expense for the mobile application advancement.

B. Integration Process

Another regular issue that frequently emerges while utilizing cross-platform is the utilization of additional time. The joining cycle with nearby settings turns out to be very long with this platform [2].

C. Limited Updates

The operating system at times doesn't uphold all the highlights of a framework. For example, iOS platform presents another update or adds another component, you will be needed to refresh the iOS adaptation of your application appropriately, in any case, and you can't do same with Android until Google delivers a similar update.

D. Sluggish Code

Utilizing cross-stage application improvement isn't as simple as it appears, fundamentally for novice engineers. Henceforth, designers utilize cross-consistence during the improvement cycle that prompts a sluggish code and evidently bringing about a moderate application [2].

VIII. Conclusion

With the rise of new technologies, demands and human needs, it is now a core aspect that applications are to be produced with rapidity. With users spread almost equally among all platforms, neglecting one of them could lead to ignoring a large mass thus causing less impact and less profit (or even loss).

Thus, cross platform development is not a skill anymore, it is a mandatory process through which every developer and firm has to go if he wants his application to stay abreast in this evolving and competitive market.

References

[1] http://comparecamp.com/qt-creator-ide-review-pricing-pros-cons-features/

[2] https://blog.vertexplus.com/cross-platform-mobile-app-development-challenges-and-opportunities/

[3] https://buildfire.com/best-mobile-cross-platform-development-tools/

[4] https://codeburst.io/app-development-ionic-or-nativescript-40b118bd8e51?gi=b32ee5aa7835

[5] https://hackernoon.com/9-popular-cross-platform-tools-for-app-development-in-2019-53765004761b

[6] https://hackernoon.com/9-popular-cross-platform-tools-for-app-development-in-2019-53765004761b

[7] https://mfg.fhstp.ac.at/development/qt-software-development-made-smater-behaupten-sie/

[8] https://techbeacon.com/app-dev-testing/10-cross-platform-mobile-development-tools-enterprises

[9] https://www.arpatech.com/blog/corona-sdk-a-perfect-framework-for-cross-platform-mobile-game-development/

[10] https://www.digitaldoughnut.com/articles/2017/april/5-advantages-of-cross-platform-mob-app-development

[11] https://www.g2.com/products/alpha-anywhere/reviews

[12] https://www.hongkiat.com/blog/cross-mobile-platform-framework-wora/

[13] https://www.netsolutions.com/insights/cross-platform-app-business-benefits/

[14] https://www.rishabhsoft.com/blog/pros-cons-cross-platform-mobile-app-development

[15] https://www.socialmediatoday.com/content/benefits-and-disadvantages-developing-cross-platform-mobile-apps

[16] https://www.techopedia.com/definition/30026/cross-platform-development

[17] https://www.trustradius.com/products/sap-cloud-platform/reviews

[18] https://www.zuehlke.com/blog/en/using-sencha-touch-as-a-strategic-platform-an-evaluation/