

**IT7x22 - Assignment One**

**Research Report**

**Semester 2, 2018**

**Research Question:**

**Is Xamarin cross-platform development cost effective & viable for mobile development in start-ups vs. native or web mobile development?**

**School of Information Technology**

### Abstract

Mobile application development is a rapidly evolving area with massive economic and scientific interest. Now with the rapidly increasing amount of mobile platforms, developing apps becomes harder for companies as they would need to re build for each platform just to increase their target market. This phenomenon can be (partially) solved using Cross-platform development, which (can) lowers development time & cost leading to a more profitable for companies.

This report focuses on trying to give clarity to Start-ups and/or larger companies on if Xamarin’s cross-platform development is the solution suited to their goals/resources, or if native or web development approach, maybe better suited to their goals/resources. Other purpose is to give hard data to backup statements.

Web Apps are applications that get access through a web browser over a network like the internet. Main point of web apps, develop once and you can deploy your code across all of the major platforms, this lowers cost & development time.

Native app is a mobile application developed solely for one mobile OS. It has the advantage of faster performance. Gives a more streamed lined use and feel plus consistency with other native apps. The user is more willing to learn how to navigate and use the app faster. Native apps have other great advantage such as full utilization of built-in capabilities of the platforms device, like GPS, camera and many more.

Hybrid applications are HTML5 websites within a native container. Hybrids give a look and feel like a native app, but outside of the simple frame of the application (restricted to the controls /navigational functions), more economical then native.

Is Xamarin the right choice for start-ups? Yes and no, every Literature, blog, YouTube video gives the same conclusion, depends on the application development scope, time and capital. So choose native approach if you need a high performance and full API access. Web approach if you require deployment on all platforms, cheap development cost and scope of app is simplistic. Hybrid if you want the look and feel of a native app to give better “user experience”, cheaper cost then native and for Xamarin if your company already uses C# and has everything build for C# environment.

### Introduction

Mobile application development is a rapidly evolving area with massive economic and scientific interest.

According to Statista the Number of app downloads worldwide in 2017 178.1 billion, 2018 205.4 billion and future estimate in 2022 will be 258.2 billion (Statista, 2018). This is amazing news for start-ups because it’s an upward trend of expansion giving a lot of room for Start-ups success in the App market (if you have the right idea).

Now with the rapidly increasing amount of mobile platforms, developing apps becomes harder for companies as they would need to re build for each platform just to increase their target market.

The traditional way of going about development for native apps is the sort of right way to go about mobile development but there is one big inordinate cost of native framework: its fundamentally impossible to re-use code for another platform; redevelopment from scratch thus leading companies the choice Native to a “at a loss” and/or “unprofitably”.

Native apps are fully developed using Integrated Development Environment (IDE) e.g. Unity, Android studio, Xcode and ect, that houses all the necessary development tools for app development & debugging. Native apps are more complicated & require a higher level of expertise and tech literacy than other types of applications.

Native apps provide the richest & stream lined “user experience”. Source code is most capable, with full access to all the devices API’s (Application programming interface), has the closes to the devices hardware so it has faster performance, Stream lined look and feel. The “user experience” means to experience of an user on how to use the device. “User experience” is an utmost key factor for app development for the user must be able to operate the app straight after installation & has an expectation that the app functions in a standard way e.g. like if the user is looking for the back button he would immediately look at the top left.

The end goal of Xamarin/cross-platform mobile app development is to gain the Native app speed, access to the API’s and to run on as many platforms as possible. This Report on “Is Xamarin cross-platform development cost effective & viable for mobile development in start-ups vs. native or web mobile development?” aims to give a detailed inquiring of this question. Main focus will be on giving a review and comparative analysis on Xamarin and other cross-platform (hybrid) mobile app development & a compile of “opinions” of expertise on using Xamarin I the real world.

### Methodology

This reports methodology use to underpin work and methods to collect data for this report involves collecting and analysing literature, reviews & opinions by using search engines like “Google Scholar” & other search engines like DuckDuckGo, and by using keywords to stick to the scope of this report such as Xamarin, Cross-platform development, cost, Mobile development, Start-ups, Native, Web mobile development. I found many literature, YouTube reviews and Blogs/Websites full of relevant information.

For literature “A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications” (Xanthopoulos & Xinogalos, 2013), “acmqueue Mobile Application Development: Web vs. Native” (Charland, Leroux & Nitobi, 2011), “Native Apps vs. Mobile Web Apps” (Jobe, Leroux & Nitobi, 2013).

For blogs/websites, “Hybrid vs Native Mobile Apps – The Answer is Clear” (Ymedialabs, 2018), “Native, Web or Hybrid Apps? What’s the Difference?”(Mobiloud, 2017), “Some thoughts after (almost) a year of real Xamarin use” (Estaun, 2015).

### Mobile Apps & Platforms

Smart Phone devices are descried as a computer in your pocket, that runs a OS (operating system) with already installed applications. How persons deploy their Apps is done through “app stores”, sometimes for free or some fees. The two biggest app stores are the Google Play & Apple App Store, is owned by the company that manufactures the OS. App Store houses third-party applications that can be downloaded & installed with a simple click by the users. These Apps are developed using refined APIs and fully device functionality through direct access to the OS components. The most common OS install on devices is Android (Google), IOS (Apple) and others like Linux Maemo and MeeGo, Symbian (Nokia), Microsoft Windows Phone, webOS (Hewlett-Packard), Samsung Bada, and BlackBerry OS (RIM). Statistics given by Statista “The smartphone platform war is over” The OS that possess the most market share is Android (Google) with a share of 84.8% of installations and IOS (Apple) with 14.4%. The other OS are very smaller with less than 0.8% for them to share.

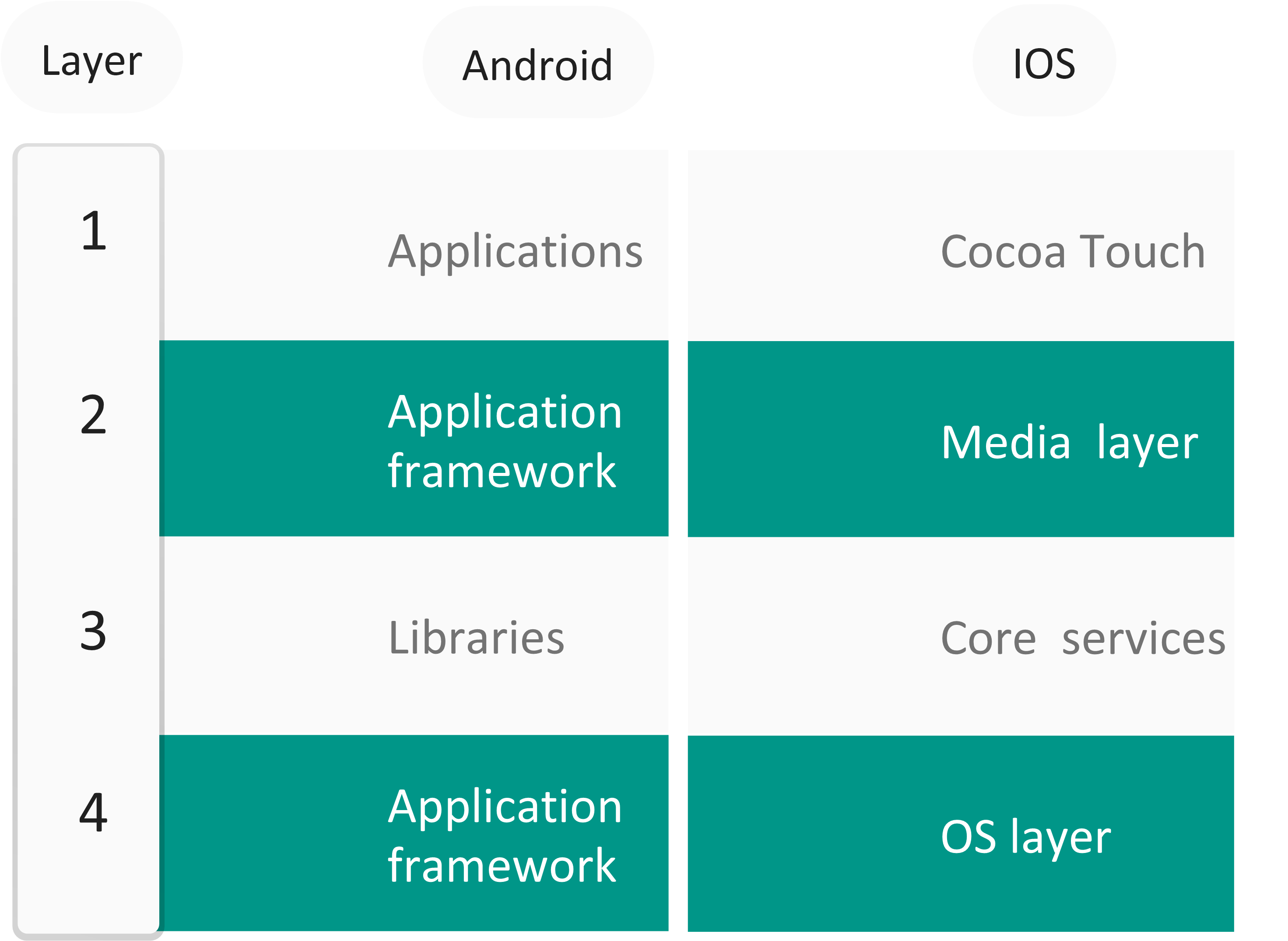
Each one of the platforms utilizes different programing language, environments & programming models grounded on that platforms specific APIs. Objective-c for iPhone, Java for Android, C # for Windows Phone and C++ for blackberry and many other platforms with other programming languages. This essentially means that you need to recreate your app each time for each platform you need to write your code one for Android, one for iPhone redo the whole thing again and again for each of these platforms. This phenomenon is call fragmentation, a phenomenon that hikes up development cost & time.

As the idea of “do it once & run everywhere” cannot be adopted when building native apps, the alternative for Start Ups or other companies is the cross-platform development for mobile apps. Cross-platform development simplifies maintenance & deployment processes, and saves development time & stress.

The idea of platform in its simplified form houses technology of a device’s hardware and software that defines how its operated and determines what other kinds of software can be used. The case of smart phones devices the Smart Phone platform layers have the OS & necessary hardware components. The OS is in control of managing the device’s heath/hardware & giving a framework for housing native apps. The OS has various built in apps, like web browser (google chrome for Android). Software development kits or SDKs, provide the needed tools & resources for the installation, development, and testing of apps.

Smart phone technologies can be understood as a set of layers in a hierarch.

Table 1 Layers



First layer of the Smart Phone platform acts as the middleman or intermediary layer between the user apps and the hardware. The layer houses every already installed & third party apps that lengthens the devices performance.

Second layer houses the media like audio & video functionality, a multimedia experience in other words.

Third layer houses all the necessary system services, that all the apps use indirectly or directly, e.g. security & data storage.

Fourth or last layer houses the fundamental technologies & OS interfaces that are mandatory for other layers, e.g. files, drivers & memory management.

### Analysis of cross-platform development

Market has a very strong growth for new software development tools and applications for smart phone devices. The growth in the smart phone market inspired the innovation of cross-platform software development environments that makes development easier & more economical. Main types of applications made by these software environments are hybrid, generated, web, progressive web apps and generated apps.

Which approach is best, it depends on the situation/problem. If a company has small capital and only needs minimal performance Web app is best, but if you are a company that needs a high performance application with all the devices functions like GPS, camera and many more native app is best. Hybrid is best if you have a need to access to all platform markets and the application to look and feel & look like native applications with little API access need.

# 5.1 Web / HTML5 App Development

Web Apps are applications get access through a web browser over a network like the internet. Main point of web apps, develop once and you can deploy your code across all of the major platforms. Programing languages include CSS, JavaScript & HTML5. Unkind to Android or IOS apps, there is no software development kit (SDK) to use when developing. Also unlike native apps, development time is relatively quick & simple (depending on your scope). It’s a nice and cheap way of prototyping your mobile app. Web Apps have no need to update nor need to install onto the device but requires internet access

### 5.1.1Table of Advantages & Disadvantages.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Easy to build (HTML/CSS/JS)  Easy to maintain  Use any technology/language  Cheaper than native and hybrid apps  Single app for all platforms | **Need to run in browser (poor user experience)**  **Slower than native apps**  **Less interactive and less intuitive**  **No icon in desktop**  **Cannot be submitted to app store**  **Cannot interact with device utilities** |

# 5.2 Native App Development

Native app is a mobile application developed solely for one mobile OS, objective-c for iPhone, Java for Android, C shop for Windows Phone and C++ for blackberry etc. Since there is a standard for technical and user experience set of rules of the OS (swipes, defined gestures etc), It’s not only has the advantage of faster performance but also history with use, giving a more streamed lined use and feel plus consistency with other native apps. The user is more willing to learn how to navigate and use the app faster. Native apps have other great advantage such as full utilization of built-in capabilities of the platforms device, like GPS, camera and many more. Basically, native apps are native to the OS.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Very fast  Built to run on specific platform  Distributed in app store  Interactive & intuitive  Interact with device utilities  Full access to device’s APIs | **Single platform**  **Harder languages**  **Very expensive**  **Hard to maintain** |

# 5.3 Hybrid App Development

In essence, Hybrid applications are HTML5 websites within a native container. Hybrids give a look and feel like a native app, but outside of the simple frame of the application (restricted to the controls /navigational functions); they are powered by a company’s website. Simply, hybrid app is a web app developed using HTML5 & JavaScript, wrapped by a native container that loads most of the content on the page as the user navigates throughout the application.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Easy to build – HTML/CSS/JavaScript  Much cheaper the native app  Single app for all platforms (PhoneGap)  No browse needed  Can usually access device utilities using APIs  Faster to develop than native apps | **Slower than native apps**  **More expensive than web apps**  **Less interactive than native apps (getting better every day)** |

### Literature review/relevant literature

# 6.1 A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications (Xanthopoulos & Xinogalos, 2013)

For this report, paraphrasing their Conclusion is the best idea.

Smart phone app development is a rapidly changing environment that has interest all over “In 2015, iOS users downloaded 25 billion apps, and Android users downloaded 50 billion” (Artyom dogtiev, 2018).

The most used phone apps was, not to recently, native apps. Native Smart phone apps are developed for one platform like Android & IOS and has the best “user experience” to offer, this is true on the basis on the idea that Native has full access.to the devices hardware and APIs. Not only that but also the fact you don’t require internet access all the time to use and maybe that the user can simply click the app icon to access the app.

However, the big growth of mobile platforms has mad the development inefficient in terms of time & development/maintenance cost. Though this is, somewhat true android’s market share is over 84.8% just in 2016 so if you’re idea is to give best “user experience” & have access to 84.8% of Smart Phone users native development for Android seems good.

The aim of the paper was to undergo an investigation of the cross-platform approaches that goals include full deployment for all platforms. Their review gave clarity that the current trend in developing cross-platform mobile apps focus on only four main app types, web, hybrid, interpreted & generated apps. Tough now in 2018 there has been more cross-platform apps types, like “Progressive Web Apps” which are simply Web apps but they behave like native apps.

Brief comments and explanation of their case study:

Their goals for the case study was to build an app using Titanium (same concept as Xamarin, hybrid approach), the apps scope was simplistic, create a simple RSS feed client, targeted at both Android and iOS platforms. The RSS client will query for the latest Apple’s press info news (http://www.apple.com/pr/feeds/pr.rss) and show the info on the device. The source code is less than 150 lines, and they tested on both OS (Android & IOS) devices, they confirmed the same source code works on Both Android and IOS.

Their goal for the case study was to see if one source code base could work on two different platforms, they succeeded but not too sure if it provides any relevancy other than it can be done.

# Taxonomy of Cross-Platform Mobile Applications Development Approaches

For this report, paraphrasing their Conclusion is the best idea.

Cross-Platform smart phone development solutions extend the SDLC or system development lifecycle by doing development once and deploying it on all platforms to have greater economic benefit. Unlike native, which you would need to re develop for every platform, you wish to deploy on.

The paper covered many new Cross-platform approaches, like Cross-Compilation (A cross compiler capable of creating executable code for other platform other than the one which it is running on), Virtual Machine approach, Model-Driven Development approach, simple Web-Based and many more. The paper gives a detailed brief for each approach; there is a lot of information within the paper.

The paper also comes to the same conclusion like the other literatures, that native are widely used but also says hybrids Apps, because they can be download from the App store of that platform.

Titanium & Xamarin are commercial solutions that are used widely thought out the industry to make native Apps. Could not find any stats to prove the statement but could find some popular apps made with Xamarin like “Olo Food app”, “MRW Mobile app”, “CA Mobile app”, “Novarum Reader” and many more, though theses apps aren’t widely used in New Zealand it still can be done.

# Native Apps vs. Mobile Web Apps

For this report, paraphrasing their Conclusion is the best idea.

Their research conclusion are that native apps are best choice for apps that require intensive hardware apps. Just like A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications, they both came to the same understanding that if your app needs full access to all the native devices APIs & hardware Native is the right choice.

Nevertheless, websites or apps the mainly consume data/content can be successfully re done with web apps. Like a small survey, you can simple use a website rather than designing and developing a native app for each platform.

Since this paper was in Kenya, which is a developing country, they had to focus on the fact that using web apps for content can drastically reduce development cost & finding expert developers. For example, (it may not be the best example to Kenya nether the less) in the United States of America, the Salary for a HTML5 Developer is “$59, 731” USD (Payscale, 2018) & the cost of a native/android developer is “$80,730” USD(Payscale, 2018). Now for Kenya their developer cost may drastically increase since theres no real competition or decrease depending on your point of view.

### Summary/conclusion

Mobile application development is a rapidly evolving area with massive economic and scientific interest. Now with the rapidly increasing amount of mobile platforms, developing apps becomes harder for companies as they would need to re build for each platform just to increase their target market. This phenomenon can be (partially) solved using Cross-platform development, which (can) lowers development time & cost leading to a more profitable for companies.

This report focuses on trying to give clarity to Start-ups and/or larger companies on if Xamarin’s cross-platform development is the solution suited to their goals/resources, or if native or web development approach, maybe better suited to their goals/resources. Other purpose is to give hard data to backup statements.

Web Apps are applications that get access through a web browser over a network like the internet. Main point of web apps, develop once and you can deploy your code across all of the major platforms, this drastically reduce development cost & development time.

Native app is a mobile application developed solely for one mobile OS. It has the advantage of faster performance. Gives a more streamed lined use and feel plus consistency with other native apps. The user is more willing to learn how to navigate and use the app faster. Native apps have other great advantage such as full utilization of built-in capabilities of the platforms device, like GPS, camera and many more.

Hybrid applications are HTML5 websites within a native container. Hybrids give a look and feel like a native app, but outside of the simple frame of the application (restricted to the controls /navigational functions), develop once and you can deploy your code across the entire major platforms thus more economical then native.

In conclusion is Xamarin the right choice for start-ups? Yes and no, every Literature gives the same conclusion, it depends on the application development scope, time and capital. So choose native approach if you need a high performance and full API access. Web approach if you require deployment on all platforms, cheap development cost and scope of app is simplistic. Hybrid if you want the look and feel of a native app to give better “user experience”, cheaper cost then native and for Xamarin if your company already uses C# and has everything build for C# environment. The future of Xamarin, looks bright as long as core concepts are kept e.g. Using the same language, data structure and APIs on different plaforms and target them with c# codebase. Main point greater profit out come in limited time period.

### References

Xanthopoulos, S.X & Xinogalos, S.X. (2013). A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications . A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications , 1(1), 213-220.

Charland, A.C, Leroux, B.L & Nitobi, N. (2011). Mobile Application Development: Web vs Native . Web apps are cheaper to develop and deploy than native apps, but can they match the native user experience?, 1(1), 1-9.

Jobe, W.J, Leroux, B.L & Nitobi, N. (2013). Native Apps vs Mobile Web Apps. PAPER NATIVE APPS VS MOBILE WEB APPS , 7(4), 27-32.

Ymedialabs. (2018). Https://ymedialabscom/hybrid-vs-native-mobile-apps-the-answer-is-clear. Retrieved 16 August, 2018, from <https://ymedialabs.com/hybrid-vs-native-mobile-apps-the-answer-is-clear>.

Mobiloud. (2017). Native, Web or Hybrid Apps? What’s the Difference?. Retrieved 16 August, 2018, from <https://www.mobiloud.com/blog/native-web-or-hybrid-apps/> .

Estaun. (2015). Some thoughts after (almost) a year of real Xamarin use. Retrieved 16 August, 2018, from <http://www.estaun.net/blog/some-thoughts-after-almost-a-year-of-real-xamarin-use/> .

connor iny, C.I. (2017). What are the best SDKs for mobile platforms?. Retrieved 6 August, 2018, from https://www.quora.com/What-are-the-best-SDKs-for-mobile-platforms

Hemendra singh, H.S. (2017). Native vs Hybrid vs Web Application: Which one should you choose?. Retrieved 6 August, 2018, from <https://theninehertz.com/blog/native-vs-hybrid-vs-web-application-which-one-should-you-choose/>

Jscrambler, J. (2017). 12 Frameworks for Mobile Hybrid Apps. Retrieved 6 August, 2018, from <https://blog.jscrambler.com/10-frameworks-for-mobile-hybrid-apps/>

Payscale. (2018). IOS Developer Salary. Retrieved 7 August, 2018, from <https://www.payscale.com/research/US/Job=iOS_Developer/Salary>

Payscale. (2018). IOS Developer Salary. Retrieved 7 August, 2018, from <https://www.payscale.com/research/US/Job=Web_Application_Developer/Salary>

Pingdom. (2007). A history of the dynamic web. Retrieved 17 August, 2018, from <https://royal.pingdom.com/2007/12/07/a-history-of-the-dynamic-web/>

Artyom dogtiev, A.D. (2018). App Download and Usage Statistics. Retrieved 17 August, 2018, from <http://www.businessofapps.com/data/app-statistics/>

Matt mansfield, M.M. (2018). STARTUP STATISTICS – The Numbers You Need to Know. Retrieved 9 August, 2018, from https://smallbiztrends.com/2016/11/startup-statistics-small-business.html

Statista. (2018). Share of Android OS of global smartphone shipments from 1st quarter 2011 to 1st quarter 2018\*. Retrieved 14 August, 2018, from <https://www.statista.com/statistics/236027/global-smartphone-os-market-share-of-android/>