XAMARIN CROSS PLATFORM DEVELOPMENT

Mobile phones have become an unavoidable friend in human life. It has become an integral part of our daily lives. We use them at all the places be it at home, travel, office or while hanging out with friends. One of the main reason why we are so much dependent on mobile phones is the ease of use and power which has been bestowed upon us through various mobile apps. With the mobile app, life has not just become easy the world has become small.

Today it is clear that the number of mobile users is greater than the number of desktop users! So, the businesses class have realized the need to effectively use mobile channels for attracting and retaining their customers. So, companies are always in search of the best platform on which they can build their mobile apps.

When it comes to platform for development there are some pre-requisites. That's the balance between the scalability, effectiveness of the platform and the cost of production. We cannot compromise on the user experience at the same time it should be cost effective for the investor putting the money in to it and also it should cover the vast user base. An app provider can no longer take the privilege of focussing one only one platform if he has to keep up with the competition because the mobile users are spread across various platform. When this is the situation cross platform application development is the only effective solution. And when we use the word 'cross platform development' the first sound which resonates with it is **Xamarin**.

Before Xamarin the practice that was followed was the Silo Approach where
we had to build independent app for each platform with no code being shared. Of course, the end user was happy but investor wasn't because to maintain an app we had to maintain separate team for each platform. After that HTML based mobile website came into picture and frame works like phone gap etc. helped developers build app in HTML and JavaScript. Here the investor was happy but the end user wasn't. because of the limited native API access, slow performance and poor user experience.

Xamarin is your complete mobile solution. The Xamarin approach to building cross-platform native apps combines the essential characteristics of native apps–native UI, native performance, and native device access–with the efficiency and time-to-market advantages of code sharing. In addition to that, Xamarin offers a wide range of products for developers to go mobile including Test Cloud, Hockey App, and a way to learn all of mobile with University.

Xamarin is considered the complete mobile solution because it provides you the too and environment to

1. Develop
2. Build
3. Test

4. Monitor & Distribute

Let us look into the details

1. **Develop**

Usually what you hear about Xamarin is that it is a platform for building native iOS, Android, Mac, and Windows Apps all in C#. When we say Native the minimum expectations are it should have Native user Interface, Full SDK Access to the platform features and the performance. These are the areas where a native app clearly stands out from a nonnative mobile app.

Xamarin has two approaches.

I. Xamarin Native

Here the UI is built natively per platform leveraging the C# capabilities. So, if it is Xamarin.iOS app it would be C# + XIB which means you can build the UI using storyboard in XCode or visual studio and write the app logic in C#. Similarly, for Xamarin.Android app it would be C# + XML which means you can build the UI using Android studio or visual studio and write the app logic in C#. And of course, for windows application we already have C# + XAML in visual studio.

Here since we are writing the app logic models, validation, web services, persistence, enterprise integrations in c# it can be shared across the three platforms.

Source:https://onedrive.live.com/?authkey=%21AACPzgZ6reUScpE&id=E5EDEAE8DF1FB78D%21900&cid=E5EDEAE8DF1FB78D

With c# being used you get all the benefits like LINQ support, ability to work with JSON and XML, Event handling and Delegates, Async/Await etc. The code is shared using a PCL or Shared project. In addition to that Xamarin offers huge support through its component store and NuGet libraries. The libraries, services and the controls we get like this will also cross platform. The Component Store, coupled with our NuGet support, brings more than 20,000 libraries to mobile development with Xamarin

And the assurance that Xamarin gives is "Anything you can do in Objective-C, Swift, or Java can be done in C# and Visual Studio with Xamarin.". You can not only have all of our .NET namespaces and libraries, but Xamarin also give us 100% api coverage of each iOS/Android API in its SDK that we access with C#. Even after all these magics we do with c# at the end we are getting a native app itself. If we use Xamarin.iOS it does full 'Ahead Of Time (AOT)' compilation to produce an ARM binary for Apple's App Store. At the same time Xamarin.Android uses 'Just In Time (JIT) compilation' on the Android device. So, the performance is not compromised and the app gets the native look and feel because it is native.

II. Xamarin Forms

With Xamarin forms it has gone a step further. Using Xamarin.Forms developers implements their app's UI in shared C# code, this increases code sharing to 90% or more while still delivering a native experience. Here the screens are written in once in C# or XAML. At runtime, these screens and their controls are mapped to the corresponding native UI elements, creating a native user experience on each platform that adheres to the design principles and user expectations of that platform.

Source:https://onedrive.live.com/?authkey=%21AACPzgZ6reUScpE&id=E5EDEAE8DF1FB78D%21900&cid=E5EDEAE8DF1FB78D

With Xamarin Forms we have over 40+pages, layouts and controls. Since we are using XAML we can have two-way binding. So, we can use the MVVM pattern for development. It supports different type of navigation. It provides various animation APIs. It supports dependency service to write platform specific implementation. And it also has a Messaging center. We can also write platform specific code in logic in both C# and in XAML. Basically, it includes everything you need to get an app up and running to build out full native applications.

For example, to show a text box in Xamarin forms we use an Entry Field. In iOS, it is rendered as UITextField, EditText in Android and as Textbox in Windows Phone.

Given below is a tabbed page in Xamarin forms which get rendered as UITabBarController in iOS, Tabbed page in Android and as a pivot in windows phone.

Source:https://onedrive.live.com/?authkey=%21AACPzgZ6reUScpE&id=E5EDEAE8DF1FB78D%21900&cid=E5EDEAE8DF1FB78D

In addition to these Xamarin form provides various plugins to access device features like Camera, GPS, Settings, Notifications Battery. For all these the developer uses the same APi, based on the running platforms the corresponding resource is accessed.

2. **BUILD**
   The ease of development always depends on the IDE used for development. Xamarin provides the best IDE with it's integration and support for Visual Studio.

With visual studio, the developer can design, develop, debug, and deploy great mobile apps. With Visual Studio developer finally have freedom to write code for all major platforms under one roof using their programming language c#. Now the cross-platform version of Visual studio has been released which can now be used in OSX platform too. Now developers can perform interactive debugging on an app that is running in the Android emulator, the iOS simulator, or even directly on hardware. The debugger supports breakpoints, catchpoints, watch expressions, stepping, and inspecting threads and local variables. We can also use Visual Studio Team Services to manage and monitor the development.

3. **TEST**

Quality of the final app is something which we a user can never compromise upon. So, testing becomes an important part in the development lifecycle. But testing an app developed in a cross-platform environment involves many challenges. The app will have to support wide variety of devices, multiple OS version, different languages and locale and various screen size.

With Xamarin we have a wide variety simulators and emulators where developers can execute their apps in a runtime environment without leaving their development environment. In addition to that Xamarin provide a cloud based test environment called Xamarin test cloud. Xamarin Test Cloud lets teams test every feature on more than a thousand devices. So catching bugs before release shortens the development cycles and allows more time for innovation.
Through Xamarin test we get:

- Complete test coverage
- Comprehensive device testing
- Fast troubleshooting
- Accelerated cycles with continuous integration
- Comprehensive support for all native and hybrid apps.
- Flexible automation

4. **DISTRIBUTE AND MONITOR**

While releasing an app to the market there should be a well-formed plan for monitoring and fixing an app which is considered the key to happier users, faster fixes, and more time for innovation. App crashes has to be closely monitored because it often results in bad user review. Also, a proper framework should be there for analytics to know how the user is using the app. For all these Xamarin provides a wide variety of options

- Hokey App : support various features like Collecting crash reports, User Metrics, Update Ad-Hoc / Enterprise apps, Update notification for app store, user Feedback, Authenticate testers.
- Xamarin Insights
- Google Analytics

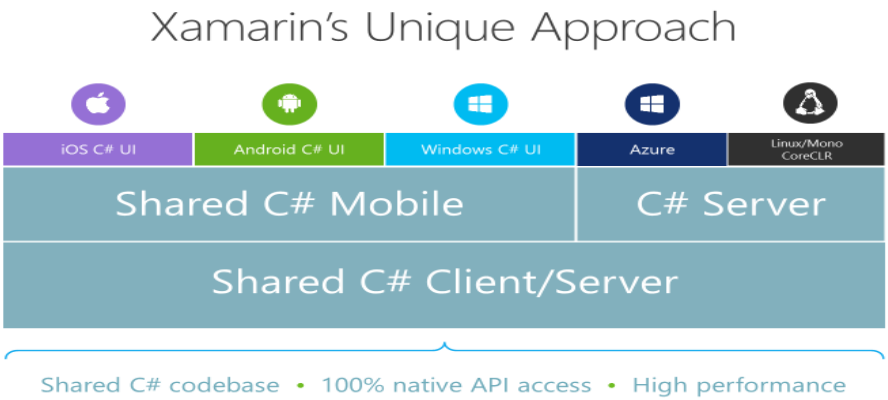
CONCLUSION

So, is Xamarin a solution for all mobile based needs? No. Only if you want consumer-facing apps with heavy UI where code sharing isn't much important we can go for the native approach.

Based on our experience we would suggest that the best use-case for Xamarin is enterprise mobile solutions. Because in enterprise application the UI will we standard and what is most important is the business logic and the content shown. For that the core product logic can be easily shared across the platforms. So, the platform customization will only take 5-10 percent of the engineering effort.

There is no doubt that Xamarin is one of the best tool for app developers to develop multi-platform apps which includes android, windows, iOS and Mac apps. So, if you are a java developer who develop android apps using android studio or an iOS developer developing using XCode you can use Xamarin Android / Xamarin iOS where you continue to write the app logic using the c# using the same API calls you made in java/objective c. If you are a c# developer coming from XAML background (WPF), you can immediately start developing app for iOS, android and windows using Xamarin forms.

With Xamarin it just isn't your front end in C# it is your full backend server as well. With Azure or even on Linux running Mono or the CoreCLR your app is fully C# end to end!



Source:https://onedrive.live.com/?authkey=%21AACPzgZ6reUScpE&id=E5EDEAE8DF1FB78D%21900&cid=E5EDEAE8DF1FB78D

You can learn Xamarin from Xamarin University. So happy coding to everyone who would like to try out Xamarin.