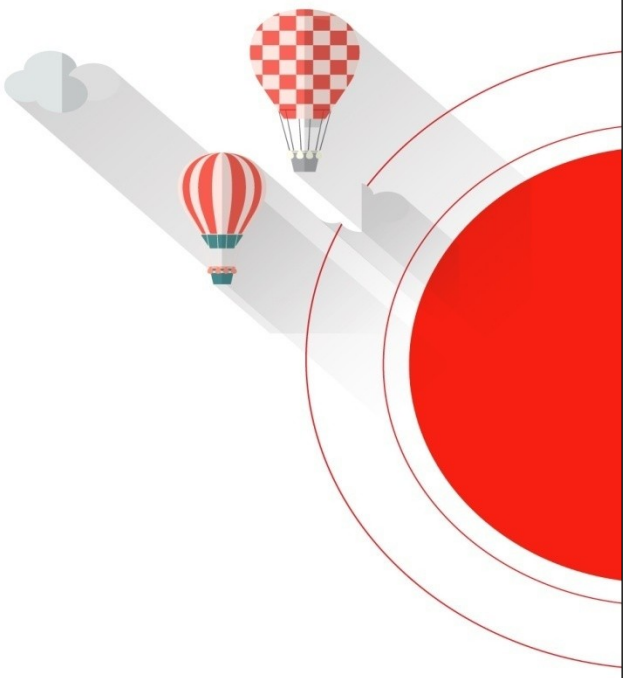
Introduction to Xamarin Cross –Platform Mobile App Development



redbytes
Appify Your Business

+91 8113 869 000

Summary:

In this document, we talk about the unique ability to create native iOS, Android, Mac and Windows apps using C# making **Xamarin**, a top contender in cross-platform development. Mobile is no longer about a single platform and its dominance. Users keep forward high demands and mobile app development companies need to be able to match those demands by providing native apps that work on operating systems simultaneously and offer a unified experience regardless of the device used. This, and a lot more, is now possible with Xamarin. Read on to find out more.

Introduction to Xamarin and its Unique Approach:

Since the advent of computers, machines have been created and designed using various methods and architectures. The recent growth in technological advancements in the digital world has made us capable of doing anything using a smart phone.

At present, the major market share is topped by Android, followed by iOS and Microsoft's Windows. If you are a budding app developer, then at a point of time you might be confused where and what to start with. And this is where Xamarin takes in action.

Xamarin has eventually turned out to be one of the popular cross-platform mobile app development frame work over the decade. As the platform works under the concept of Write once, Run everywhere (WORA),

Xamarin is already one of the preferred framework for Android, iOS or Windows owing to the fact that it helps in developing apps for multiple platforms using a single programming language. This platform is based on C# programming language and can be used in the cross-platform mobile app development with native UI and APIs to deliver better performance.

There is a way to address all the issues of cross-platform development mentioned above. Xamarin Studio (IDE) ensures a holistic strategy to develop mobile applications for multiple platforms. Xamarin Studio comprises of code completion in C#.

iOS mobile apps are designed with the integrated Xcode Interface Builder. Similarly, an Android UI builder is included for designing user interfaces without the need for XML. It also comes with a debugger for testing of mobile applications.

The advantage of Xamarin includes native UI, access to specific-device functions, and native performance. It's also hugely beneficial in code reuse & sharing which shortens app deployment time. Achieve cost-effective cross-platform development along with the flexibility to handle any app use case and user experience requirement.

What is Xamarin?

Xamarin, the cross-platform framework is a Microsoft-owned San Francisco-based software company founded in the year 2011. According to reports, over 1.4 million developers have used Xamarin products across 120 countries around the world.

And recently in 2016, this platform was acquired by Microsoft. Apps created with Xamarin have native features and also shares a common codebase. As per Xamarin stats, more than 15,000 companies rely on their tools and the list even includes Fortune 500 companies.

Number of Elements in the Xamarin platform:

Xamarin platform consists of a number of elements that allow the development of applications for both Android and iOS:

1) C# language: With this one can use the same kind of syntax and established features like LINQ, Generics and the Parallel task library.

2) Compiler: The main function of the compiler is to create a native app or a .Net application with run time and also optimizes many features for the app deployment process.

3) Mono .NET Framework: This framework provides a cross-platform implementation with various features.

4) IDE Tools: In Xamarin**,** the Visual Studio on Mac and Windows allows you to create, build, and deploy Xamarin projects.

System Requirements for a Xamarin Application:

1) Mac:

- Requires a Mac computer running OS X Yosemite or even higher
- Apple's Xcode IDE and iOS SDK
- Xamarin iOS SDK
- Xamarin Studio

2) Windows:

- Visual Studio (Professional/higher)
- Xamarin for Visual Studio
- Requires a computer with at least 2GB RAM & running Windows 7/higher

Before Xamarin was purchased by Microsoft, this platform had a number of issues related with:

- The license was not totally free
- Lack of support and resources
- The learning process seemed to be quite long as the native libraries for each platform need to be known very well

Benefits of using Xamarin for App Development:

1) Native User Experience: The unique approach of Xamarin provides the best experience in every platform, with the benefit of a native interface, i.e., access to features of the device

2) Open-source platform: After Xamarin was acquired by the Microsoft, some of the components were made open-source, which means that the developers have the freedom to modify or change the codes according to the app feasibility.

3) Reuse the same API: This is one of the major advantages of using Xamarin as it lets the developers use the similar APIs for the apps in iOS or Android in their specific program based languages. This way the same code can be used to develop an app on multiple platforms, which in turn reduces the time spent on cross-platform app development.

4) App Logic is shared: Shares the same code to multiple platforms

5) The advantage of using C#: A simple yet premier programming language C# facilitates simple data types and comparatively it's highly cost-effective to carry out a Xamarin app project.

6) Xamarin Component Store: The Xamarin component store is rich in terms of cross-platform libraries, UI controls, and third-party web services. Within this, Xamarin developers have the freedom to choose from free/paid components.

7) Xamarin Test Cloud: iOS and Android apps can be tested automatically using Xamarin Test Cloud, on multiple devices. Test Cloud offers continuous integration, reports, test for fragmentation, and object-based UI testing.

The integration between Microsoft and Xamarin is targeted at supporting developers in extending their apps across multiple devices. With Xamarin, developers combine all of the productivity benefits of C#, Visual Studio & Windows Azure with the flexibility to quickly build for multiple device targets.

Features of Xamarin:

- Development of Xamarin apps is the combination of native platforms along with many features of its own
- This platform can directly make use of Objective-C, Java and C++ libraries which will give the ability to reuse 3rd party code bases which are in Java, Objective-C or C++.
- Xamarin uses C# which is a modern language which has many improvements over Java and Objective-C
- This platform reduces the time and cost of mobile app development
- This has a massive collection of class libraries
- Secure & scalable backend
- Fast development and fast time-to-market
- It provides robust compile-time checking which will, in turn, fewer the run-time errors and also give high-quality apps
- Xamarin offers cross-platform support for platforms like Android, iOS, and Windows and can share up to 90% of their code base during development.
- Facilitate the feature to combine codes for native platforms
- Xamarin supports Apple and Android Watch devices. This lets engineers use Visual Studio for Mac/Windows to build future-centric apps for wearables.
- **For Eg:** In the case, if you want to run an app in iOS, then select the particular code, compile and then you get the apk file for iOS same Android/Windows. So, the advantage of this feature is that there is no need for code in the same app for different language as the single code can be executed in all platforms

Business Benefits to Enterprises

1. There are around 4.5 billion mobile devices in the world and this is expected to reach 5 billion by 2019
2. Ideal for native apps & cross-platform development
3. Has around 700, 000 developers community support
4. Easy development and deployment
5. 28 % of Fortune 500 companies use Xamarin for mobile app development as it quickly finds its way to the market
6. Xamarin has integration with SAP and Salesforce platform and it is also backed by Microsoft and has collaborations with companies like IBM, Samsung etc.
7. Reuse 60-80% of the code resulting in a shorter development cycle

Comparison of Xamarin vs Native vs Hybrid

	Xamarin	Native	Hybrid
Tech stack	One tech stack, single codebase (C#, .Net framework + native libraries)	Different tech stacks for each platform.	One tech stack, single codebase (Javascript, HTML5, CSS)
Code sharing	Yes (up to 96% with Xamarin.Forms)	No, different code bases	Yes, 100%
UI/UX	Complete UI customization is possible for each platform (with Xamarin.iOS and Xamarin.Android)	Completely platform-specific UI	Common UI for all platforms (limited customization capabilities)
Performance	Good, close to native	Excellent	Medium - Poor
Hardware capabilities	High - Xamarin uses platform-specific APIs, and supports linking with native libraries	High - Native tools have complete support for system capabilities out of the box.	Medium - The capabilities can be accesses through third-party APIs and plugins, although there are some risks due to the poor quality and unreliability of most of these tools.
Time to market	With Xamarin.Forms the time to market is fast due to the limited customization and extensive code sharing. Xamarin.iOS and Xamarin.Android require slightly more time as the amount of custom code increases.	The time to market for iOS or Android native app might equal that of Xamarin.Forms or Hybrid tools. Yet, building apps for multiple platforms, will require you to either prolong time to market or increase the number of developers involved.	Hybrid solutions offer the fastest time to market thanks to the single code base and minimum customization. These tools are even used for prototyping and proof of concept projects.

Certifications:

1) *Xamarin Mobile Certification:* Achieving a Xamarin Certification is a great way to showcase your potential and also learn the core principles of this cross-platform tool.

To complete the certification, one needs to complete *various courses as listed below:*

1) Getting Started with Xamarin
2) Introduction to Xamarin.iOS
3) Introduction to Xamarin.Android
4) Introduction to the Xamarin Designer for iOS
5) Activities and Intents
6) Introduction to Cross-Platform Mobile Development
7) Introduction to Xamarin.Forms
8) XAML in Xamarin.Forms
9) Layout in Xamarin.Forms
10) Resources and Styles in Xamarin.Forms
11) Consuming REST-based Web Services
12) SQLite and Mobile Data
13) Preparing for Publishing
14) Fundamentals of TableViews
15) ListViews and Adapters in Android
16) Customizing TableViews
17) Toolbar and App Bar
18) Navigation Patterns
19) Android Navigation
20) Patterns for Cross Platform Mobile Development
21) Data Binding in Xamarin.Forms

22) Using ListView in Xamarin.Forms

23) Mobile Application Architecture

24) Xamarin.Forms Effects

25) Diagnosing Memory Management Issues

Various Self-Guided Learning

This also includes the same content that is in the live classes, but only the difference is that in this you can learn according to your own pace.

1. Introduction to Xamarin.Android

2. Activities and Intents

3. Introduction to Azure

4. Building an Azure Mobile App Service

5. Building an Azure Mobile App Client

6. Authentication with Azure

7) Introduction Xamarin Designer for iOS

8) Introduction to Universal windows Platform

9) Introduction to Cross-platform Mobile app development

10) Introduction to Xamarin.Forms

11) SQLite & Mobile Data

12) Layout in Xamarin.Forms

Inference:

Even though Xamarin is loaded with features and comes like a dream come true for developers looking for cross-platform development, it's a bit expensive. When the very first time when Xamarin was introduced into the market, it wasn't that popular and developers were still trying to figure out more of its features and to go for Web Vs Cross-platform apps. But as Microsoft took up Xamarin, everything changed and with its front end cross-platform development approach, developers could deliver mobile apps successfully in half a time.

Xamarin continues to build a significant presence in the enterprise market, as well as be a growing partner ecosystem, both in terms of strategic partnerships, integrations (SAP), as well as a significant pool of implementation and channel partners. Xamarin provides UI/UX

and performance advantages to make a mobile app successful. In the past, Xamarin was considered a small vendor, but now with Microsoft's acquisition, many of the clients globally are ready to make significant investments in developing apps using Xamarin.

Enterprises usually prefer cross-platform app development in order to extend their reach and cater to a wider audience. Xamarin is built for that reason with its ability to share code, app logic while saving on additional resources & development time. Based on various results, it's quite safe to say that **Xamarin** has a steady position in the mobile app market.