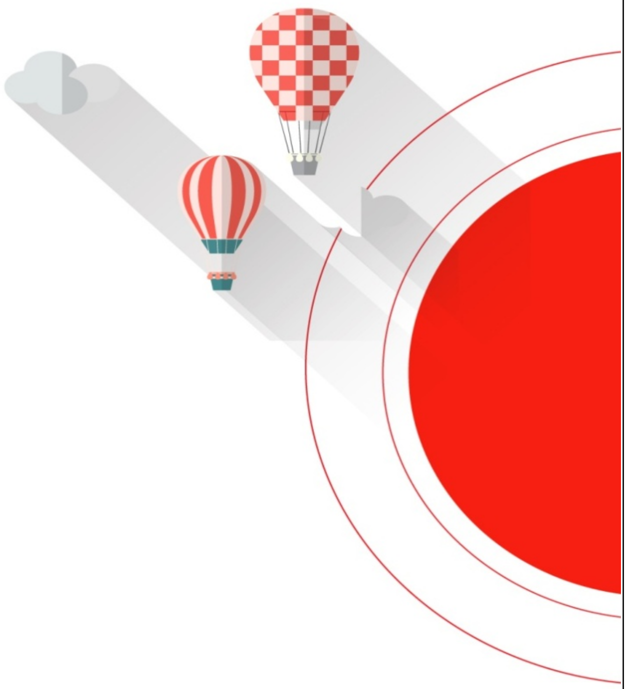
A Tutorial on Xamarin



redbytes™

Appify Your Business

Introduction to Xamarin

The Emergence of Cross Platform Ecosystem in Mobile app Development

Until recently, mobile app development specialists had to follow a stringent slot of either iOS, Android or Windows. Due to differing ecosystem, language constraints and style, the process of software development on each platform was entirely different. It was therefore extremely challenging for developers to build separate mobile apps for these platforms.

However, Xamarin came to transform the process and altered the course of the situation through enabling unique facility known as **cross platform** development. Xamarin allows developers to build application for iOS and Android with C# and .NET libraries. To establish the performance and feel of native apps, **Xamarin mobile development** enables developers to leverage full range of native APIs and UI controls available from the OS and hardware of the respective devices.

What is Xamarin?

Xamarin offers unique ability to iOS and Android developers to work their way towards writing mobile applications in C# and then run them on any desired mobile platform or OS.

Earlier, native languages like Objective-C, Swift and Java were the only limited choices for creating iOS and Android applications. In a past few years, Xamarin has bred novel ecosystem of platforms for **cross platform** mobile application development. It offers single language (C#) and runtime that seamlessly functions across iOS, Android and Windows and compiles native apps.

Why to Choose Xamarin?

Each platform has its own discipline of features and methods of writing native applications where native code interoperates and interacts smoothly with underlying Java base. Some platforms support apps written in HTML and JavaScript whereas some allow C/C++ code. In such traditional setting, there should be separate codebases for each device or your app run via web browser, which imposes major concerns of maintenance and cost.

In **Xamarin mobile development**, developers eliminate typical issues as faced in native app development. For instance, web browser apps do not run natively as its UI is not familiar to device ecosystem and its performance is also compromised due to the absence of native code.

Xamarin's **cross platform** capability is superior because:

- It consists of native features: UI, API access and performance
- Native UI of each device makes your app user-friendly

- You can perform the same coding results as achieved using Objective-C, Java and Swift
- Due to access to device's individual APIs, you can experience native performance
- Enjoy shared codebase across all devices

Xamarin products

- Xamarin.iOS
- Xamarin Android
- Xamarin.forms

Specialities of Xamarin.iOS and Xamarin.Android

Developers can leverage the power of UI toolkits Xamarin.iOS and Xamarin.Android in order to create an app using each platform's APIs .NET framework. While developing apps with these two products, you can relish customizations for UI for each platform.

In both UI toolkits, you will have the benefit of separate UI approach. Xamarin.iOS and Xamarin.Android comes with shared code base in C# which is shared across platforms with separate UI for each.

Xamarin mobile development with Xamarin.iOS and Xamarin.Android seems suitable for:

- Apps where native-like behaviour and interactions are expected
- Development of apps with platform-specific native APIs
- Mobile apps with UI customization scope

Specialities of Xamarin.forms

The best part about using **Xamarin.forms** is that while building an app with this UI kit, developers will need to write code only once. At the moment of runtime, this written code is translated to platform's native environment.

Another speciality of **Xamarin.forms** is that you can enjoy the benefit of the maximum code sharing which is as much as 90% with this UI kit.

Hence, **Xamarin.Forms** is suitable when:

- You don't require platform-specific functionality for apps
- Code sharing is more crucial than custom UI

Xamarin components

With the launch of Xamarin 2.0 in March 2013, the Component Store was introduced to help developers save time and accelerate **Xamarin application** development process and enjoy potential features. The store contains the stunning catalog of UI controls, charts, themes, cloud services and other great features that you can integrate in your app in matter of a few clicks.

To speed up the process of adding **Xamarin components** to your app, the Store is integrated into both Xamarin Studio and Visual Studio. To get detailed insight into how exactly you can create, bundle, test and submit your components to the Xamarin Store, refer to the full Xamarin Component Store Guide.

Here is the quick overview to **Xamarin components** and submission.

Submission Overview

After component creation and packaging and testing, you will need to submit your components and **Xamarin plugins** to Xamarin component store. Before you proceed, it is believed that you have some experience with yaml and building a component. If you are not aware of how to produce a component, start going through its insightful Xamarin Component Store guide.

Once you have finished building a component and are happy with it, you can submit your creation to Xamarin Store where it will be reviewed and approved.

You will need to set up your own Xamarin Account in case you don't have one. To proceed for creating the Xamarin account, visit Xamarin Home page and click "Sign In" at the top right corner. Follow the instructions there after clicking on "Create a new account".

If you are not willing to keep it open source, you can place the specific charge for the use of any of **Xamarin components** submit by you on the Store. To officially set up the charge for each component and **Xamarin plug-ins** you create, you will need to provide your account details. The sales will be credited to your account provided that 30% of the revenue is deducted as commission towards Xamarin Store.

To understand the process of updating the existing component – which is different from creating a new one – you should study the guide on Xamarin store website.

Approval Process

After you have submitted the new or updated component, you wait for Xamarin Store to review it on the basis of Component Review Guidelines. If they discover any issues or errors, you will be contacted with the details of needed fixes and improvements.

If there is not issue found in your component or **Xamarin plug-ins**, it will receive the final approval and be made available officially on the component store where people can explore and download when required.

Major Xamarin components

Some of the most powerful **Xamarin components** you may find on its online library for **Xamarin application development** are:

- **SNMP library:** Solid easy to use, open source SNMP library for iOS and Android
- **AchartEngine:** Charting library for Android applications
- **ActionTray:** A dockable customizable slide-out, tray view controller for iOS and Android
- **Advance Colour Picker:** An open-source colour picker component for Xamarin.iOS
- **Alliance Calendar:** Beautiful Android calendar control with elegant features
- **BubbleTree:** Represent tree structures in mobile apps. Search an element in bug tree structure in an easy and intuitive way.
- **Built.io:** Mobile and web back-end and app development platform by raw engineering
- **Card.io for Android and iOS:** Credit card scanning for mobile apps for both platforms
- **Dolby Audio Component:** Leverage the power of Dolby audio in your Android and Games
- **Download Manager:** Handles the downloading of files from the internet using either HTTP or FTP protocols
- **Eggs-To-Go:** Cross-platform mobile library for implementing Easter Egg gestures
- **Google App indexing for iOS:** Get your app found in Google Search
- **Google Cardboard:** It enables VR on your smartphone for experiencing virtual reality in simple, fun and affordable way

Features of Xamarin

Xamarin's unique quality is that it combines most quintessential power of native platforms with its own set of comprehensive features to enable Xamarin application development. Some of its great features include:

1. **Leverage the full potential with the underlying SDKs**

   For iOS and Android developers, Xamarin binds the whole underlying platform's SDK to establish native experience in its full capacity. In both the platforms, these bindings work effectively and are strongly-typed. They are easy to use and navigate and offer robust compile-time type checking during the development process, which translates into minimum runtime errors and high-quality mobile apps.

2. **Integration of Objective-C, Java, C, and C++ libraries**

   Xamarin application development allows you to directly take advantage of existing iOS and Android libraries written in Objective-C, Java or C/C++. This facility gives you the freedom and power to integrate a wide variety of 3rd party code already created in this library. Also, using declarative syntax, developers can easily bind native Objective-C and Java libraries.

3. **Sophisticated Language Constructs**

   Xamarin applications are all written in C# which is much more sophisticated and improved language than its previous predecessors Objective-C and Java. Its potential improvements are Dynamic language features, Functional Constructs (Lambdas), LINQ, Parallel programming features, modern Generics, etc.

4. **Massively feature-rich Base Class Library (BCL)**

   Xamarin applications benefit greatly from .NET BCL, which is a massive collections classes that contain comprehensive and much powerful features such as Database, XML, Serialization, IO, Networking support, String and much more. Moreover, by compiling existing C# code in an application, you can gain access to enormous number of libraries that allow you to perform more actions that BCL basically offers.

5. **Modern (IDE)**

   The inimitable quality of Xamarin is that it uses Visual Studio on Mac OS X and on Windows. The Visual Studio for both platforms have modern IDEs with features such as a sophisticated Project and Solution management system, code auto completion, integrated source control, a comprehensive project template library, and more.

6. **Mobile Cross Platform Support**

   To facilitate frictionless Xamarin application development for leading mobile platforms (iOS, Android and Windows), Xamarin offers sophisticated cross-platform support. Xamarin.Mobile library has a unified API to avail common resources across these platforms. You can write Applications with up to 90% of shared code. Xamarin's cross-platform support saves substantial amount of cost and time needed to market products for three popular platforms.

These are some of Xamarin's incredibly powerful and compelling features that contributing to enhancing the Xamarin application development. It is due to this powerful feature set that application developers can now fill the void experienced during the development process. They can finally capitalize on the strength of modern language and comprehensive platform to develop cross-platform mobile applications.

Xamarin App Development Tools

Xamarin is one of unstoppable buzzwords among developers who avidly gravitate towards cross-platform frameworks to build cross-platform applications across major platforms (Android, Windows, iOS, OS X) with native UI, experience and performance. Hence, if you are looking for highly productive tools to ease your Xamarin development process, this section will walk you through all the essential development tools used for creating Xamarin apps.

1) Visual Studio

Visual Studio is Microsoft's advanced and widely used sophisticated IDE. Known best for its features for mobile, desktop and web application creation, it supports mobile development of:

- .NET (Android, iOS, tvOS, watchOS)
- Mac desktop apps
- .NET Core applications
- ASP.NET Core Web application
- Cross-platform Unity games

Some if its significant features include debugger, rich editor, integrated source control and native platform integration with iOS, Mac and Android. The latest Visual Studio for Mac incorporates a range of features including cloud, web development and potential improvements in cross-platform mobile development.

2) Xcode

Dedicated to Apple's various OS platforms, Xcode IDE facilitates the entire Apple development ecosystem. It creates immensely productive environment for developers to build apps for Mac, iPad, iPhone, Watch OS and Apple TV. Even though you can use Xamarin to develop iOS apps on Visual Studio running in a Window system, you can't deny the role of Mac computer because you will also require an access to Mac system to perform the remote actions of compilation and debugging.

3) NuGet

NuGet is another essential tool that enables mechanism for modern developers to create, share and consume reusable code. NuGet is a main repository of over 100,000 code packages produced by .NET developers. You can also host packages in the private cloud or network or on your local storage system. This way, code bundles will be accessible to developers who have permission to the host, maintaining in place the limited, user-specific availability.

4) Xamarin Inspector

When it is time to run and test apps for bugs, Xamarin Workbooks and Inspector offers an interactive C# console. The Xamarin Inspector integrates with the app debugger of your IDE so that users can capitalize on its tools to diagnose, inspect and debug the app elements smoothly.

5) Prism:

Prism is a framework used to build testable XAML applications in WPF, Windows 10 and Xamarin Forms. Basically Prism offers implementation of design patterns for writing well-structured and maintainable XAML apps. It includes MVVM, commands, dependency injection, EventAggregator, etc. Prism provides shared code base in a Portable Class Library which targets these core platforms while enabling integration of these design patterns with the target platform.

6) MFractor

In order to make Xamarin application development easier and more simplified, MFractor offers XAML IntelliSense, 100+ XAML inspections and re-factorings, image tooling and more. It also contains an enhanced XAML editor that boosts your Xamarin.Forms development by increasing the speed of discovering code issues and ultimately fixing them in a few clicks.

MFractor also helps you with missing essentials such as:

- Image importing wizard
- Localization assistance
- Handy mobile navigation short-cuts
- Code snippet support
- Configuration engine
- Project-specific customization options

7) MVVM Frameworks

MVVM, which is also known as Model-View-View-Model, is a reliable technique tailored for XAML-based environments in order to architect an application. It does so by separating UI from the underlying code used to integrate business logic. You can use MVVM frameworks in your Xamarin Forms while opting for Xamarin application development. Some of them are:

- MvvmCross:https://github.com/MvvmCross/MvvmCross
- Mvvm Light:http://www.mvvmlight.net/
- Fresh MVVM:https://github.com/rid00z/FreshMvvm

8) ReSharper

If you are looking for an accurate on-the-fly code quality analysis in different languages, ReSharper is a Visual Studio extension you may consider. Designed for .NET developers, ReSharper supports .NET MVC, C#, ASP.NET, VB.NET, XAML, JavaScript, HTML, CSS and XML. Based on its analytical capability, developers can determine whether or not their code needs further improvement.

- You can be warned of any problem in your code
- You can take advantage of hundreds of quick fixes to automatically resolve the issue
- Leverage variety of quick fix options to pick and choose the best from

It is also equipped with many code editing helpers such as:

- Extended IntelliSense
- Instant code transformations
- Auto-importing namespaces
- Code rearrangement
- Documentation display

9) Sketch

Sketch is a vector-based, powerful design tool for Mac and its focus is mainly on screen design. Developers and designers can leverage Sketch for creating website structures, icons, and UI for both desktop and mobile devices. Sketch is handy and powerful to both beginners and established professionals who wish to specialize in one domain – which is design.

Thick with lots of pros, Sketch is tailored to address the needs of digital artists who expect to build spectacular applications. Due to its alluring feature set, Sketch as a design tool has become a widely popular choice among dedicated design companies and start-ups today.

10) Balsamiq

Balsamiq is another design aid that helps designers get faster, better and smarter in their design building experience. For once, it uses computer to reproduce the experience of sketching on a whiteboard and making wireframes faster. This ultimately helps creative designers generate more ideas and think of the best solutions possible. Here in Balsamiq, one can just drag and drop to elements for resizing, make fast changes and bring clarity to their work.

11) Adobe XD

Providing all-in-one UX/UI solutions for website and mobile apps designs, Adobe XD enables faster concept-to-prototype journey. Adobe XD offers the following major benefits:

- Helps designers achieve maximum precision, performance and task efficiency
- Best complements the effortlessness of UX/UI designers
- Its intuitive tools are built to suit exact needs of designers

- It is equipped with speed boosting features such as Repeat Grid and flexible artboards
- Developers can create fully interactive prototypes in shortest of time
- Switch from static layouts to interactive prototypes in a single click
- Edit your design, view prototype updates automatically without syncing
- One can review and share prototypes with transitions on iOS and Android devices

12) Visual studio Unit Testing Framework

The Visual Studio Unit Testing Framework is Microsoft's unit testing tools suite that comes integrated into Visual Studio versions. Developers and QA testers familiar with Visual Studio can easily apply their experience in testing apps with the unit tests created in the framework and execute the same in Visual Studio. It is defined in Microsoft.VisualStudio.QualityTools.UnitTestFramework.dll.

13) xUnit.net

Written by the creator of NUnit v2, xUnit.net is the latest technology in testing tools. It is an open-source and community-focused unit testing tool working in the .NET framework. It supports unit testing for multiple languages including C#, F#, VB.NET, and other languages of .NET family. xUnit.net complies with the code of conduct of .NET foundation and works well with ReSharper, CodeRush, TestDriven.NET, and Xamarin.

Xamarin.iOS and Xamarin.Android

Apps built using Xamarin cross-platform development framework are unique for individual device platforms (Android and iOS) as Xamarin understands features for each platform quite well. Here is what Xamarin.iOS and Xamarin.Android offers to cross-platform mobile application developers:

100% API access for each platform

It enables developers to create an app using independent APIs that belong to each underlying device platform. Xamarin.Android and Xamarin.iOS offer 100% API access with all the merits of .NET APIs, which implies that you have sheer power to do anything you wish in Android or iOS with Xamarin in C#.

This is an added advantage of building apps using Xamarin cross-platform technology, which means it is easy to deliver exceptional native experience for mobile applications built with Xamarin.

Smart Separate UI approach

Developers can leverage the approach of having separate UI design toolkits known as Xamarin.iOS and Xamarin.Android. This indicates that you can capitalize on the power of shared code base in C# with Xamarin.iOS and Xamarin.Android, resulting in a controlled structure where common business logic is conveniently shared across multiple platforms with separate UIs. Thus the two products help Xamarin developers tailor-build UI designs for each platform involved.

Xamarin.Android and Xamarin.iOS unleashes the complete potential of Android SDK to .NET developers in order to craft perfectly native Android apps in C# in Visual Studio.

Suitability

While developing cross-platform mobile application using Xamarin technology, if you are confused as to where Xamarin.iOS and Xamarin.Android are most suitable, here is where both the tools come handy:

- Building apps with native-like behaviour and interactions
- Developing apps with platform-specific native APIs
- Mobile apps with custom UI development

Windows

As long as Windows applications go, the Windows is already built in C# and supports C# for development along with native APIs.

Xamarin.Forms

Xamarin.forms is another UI toolkit that allows cross-platform developers to write code only once and then share application UI across all platforms. At the time of running the mobile application, the written code blends into platform's native structure.

Xamarin.Forms also sets free a complete UI toolkit for .NET developers to build fully native Android, iOS, and Windows apps in Visual Studio.

Another pro of using Xamarin.forms is that developers can relish the luxury of sharing maximum code (which is up to 90%) with this UI kit. With Xamarin.forms, all it takes is the strength of single team of Xamarin developers to build apps for multiple platforms.

Xamarin.Forms is equipped with:

- UI building blocks (pages, layouts, and controls)
- Complete set of UI controls defined in XAML
- Data binding
- Custom renderers
- Navigation
- Animation API
- Dependency Service
- Messaging Centre

Advantages of Xamarin.Forms

- Native app feel
- Shared Business Logic
- Shared UI
- More efficiency & less development time

Suitability

During cross-platform mobile app development, Xamarin.Forms seems suitable in situations like:

- Building apps where platform-specific functionality is not compulsory
- Cross-platform projects where code sharing is more emphasized than custom UI

Steps to Create a Mobile App Using Xamarin

Xamarin is one of unstoppable buzzwords among developers who avidly gravitate towards cross-platform frameworks to build cross-platform applications across major platforms (Android, Windows, iOS, OS X) with native UI, experience and performance. Hence, if you are looking for highly productive tools to ease your Xamarin development process, this post will walk you through all the essential development tools used for creating Xamarin apps.

Visual Studio

Visual Studio is Microsoft's advanced and widely used sophisticated IDE. Known best for its features for mobile, desktop and web application creation, it supports mobile development of:

- .NET (Android, iOS, tvOS, watchOS)
- Mac desktop apps
- .NET Core applications
- ASP.NET Core Web application
- Cross-platform Unity games

Some if its significant features include debugger, rich editor, integrated source control and native platform integration with iOS, Mac and Android. The latest Visual Studio for Mac incorporates a range of features including cloud, web development and potential improvements in cross-platform mobile development.

XCode

Dedicated to Apple's various OS platforms, XCode IDE facilitates the entire Apple development ecosystem. It creates immensely productive environment for developers to build apps for Mac, iPad, iPhone, Watch OS and Apple TV. Even though you can use Xamarin to develop iOS apps on Visual Studio running in a Window system, you can't deny the role of Mac computer because you will also require an access to Mac system to perform the remote actions of compilation and debugging.

NuGet

NuGet is another essential tool that enables mechanism for modern developers to create, share and consume reusable code. NuGet is a main repository of over 100,000 code packages produced by .NET developers. You can also host packages in the private cloud or network or on your local storage system. This way, code bundles will be accessible to developers who have permission to the host, maintaining in place the limited, user-specific availability.

Xamarin Inspector

When it is time to run and test apps for bugs, Xamarin Workbooks and Inspector offers an interactive C# console. The Xamarin Inspector integrates with the app debugger of your IDE so that users can capitalize on its tools to diagnose, inspect and debug the app elements smoothly.

Prism

Prism is a framework used to build testable XAML applications in WPF, Windows 10 and Xamarin Forms. Basically Prism offers implementation of design patterns for writing well-structured and maintainable XAML apps. It includes MVVM, commands, dependency injection, EventAggregator, etc. Prism provides shared code base in a Portable Class Library which targets these core platforms while enabling integration of these design patterns with the target platform.

MFractor

In order to make Xamarin application development easier and more simplified, MFractor offers XAML IntelliSense, 100+ XAML inspections and refactorings, image tooling and more. It also contains an enhanced XAML editor that boosts your Xamarin.Forms development by increasing the speed of discovering code issues and ultimately fixing them in a few clicks.

MFractor also helps you with missing essentials such as:

- Image importing wizard
- Localization assistance
- Handy mobile navigation shortcuts
- Code snippet support
- Configuration engine
- Project-specific customization options

MVVM Frameworks

MVVM, which is also known as Model-View-View-Model, is a reliable technique tailored for XAML-based environments in order to architect an application. It does so by separating UI from the underlying code used to integrate business logic. You can use MVVM frameworks in your Xamarin Forms while opting for Xamarin application development. Some of them are:

- MvvmCross : https://github.com/MvvmCross/MvvmCross
- Mvvm Light : http://www.mvvmlight.net/
- Fresh MVVM : https://github.com/rid00z/FreshMvvm

ReSharper

If you are looking for an accurate on-the-fly code quality analysis in different languages, ReSharper is a Visual Studio extension you may consider. Designed for .NET developers, ReSharper supports .NET MVC, C#, ASP.NET, VB.NET, XAML, JavaScript, HTML, CSS and XML. Based on its analytical capability, developers can determine whether or not their code needs further improvement.

- You can be warned of any problem in your code
- You can take advantage of hundreds of quick fixes to automatically resolve the issue
- Leverage variety of quick fix options to pick and choose the best from

It is also equipped with many code editing helpers such as:

- Extended IntelliSense
- Instant code transformations
- Auto-importing namespaces
- Code rearrangement
- Documentation display

Sketch

Sketch is a vector-based, powerful design tool for Mac and its focus is mainly on screen design. Developers and designers can leverage Sketch for creating website structures, icons, and UI for both desktop and mobile devices. Sketch is handy and powerful to both beginners and established professionals who wish to specialize in one domain – which is design.

Thick with lots of pros, Sketch is tailored to address the needs of digital artists who expect to build spectacular applications. Due to its alluring feature set, Sketch as a design tool has become a widely popular choice among dedicated design companies and start-ups today.

Balsamiq

Balsamiq is another design aid that helps designers get faster, better and smarter in their design building experience. For once, it uses computer to reproduce the experience of sketching on a whiteboard and making wireframes faster. This ultimately helps creative designers generate more ideas and think of the best solutions possible. Here in Balsamiq, one can just drag and drop to elements for resizing, make fast changes and bring clarity to their work.

Adobe XD

Providing all-in-one UX/UI solutions for website and mobile apps designs, Adobe XD enables faster concept-to-prototype journey. Adobe XD offers the following major benefits:

- Helps designers achieve maximum precision, performance and task efficiency
- Best complements the effortlessness of UX/UI designers
- Its intuitive tools are built to suit exact needs of designers
- It is equipped with speed boosting features such as Repeat Grid and flexible artboards
- Developers can create fully interactive prototypes in shortest of time
- Switch from static layouts to interactive prototypes in a single click
- Edit your design, view prototype updates automatically without syncing
- One can review and share prototypes with transitions on iOS and Android devices

Visual studio Unit Testing Framework

The Visual Studio Unit Testing Framework is Microsoft's unit testing tools suite that comes integrated into Visual Studio versions. Developers and QA testers familiar with Visual Studio can easily apply their experience in testing apps with the unit tests created in the framework and execute the same in Visual Studio. It is defined in Microsoft.VisualStudio.QualityTools.UnitTestFramework.dll.

xUnit.net

Written by the creator of NUnit v2, xUnit.net is the latest technology in testing tools. It is an open-source and community-focused unit testing tool working in the .NET framework. It supports unit testing for multiple languages including C#, F#, VB.NET, and other languages of .NET family. xUnit.net complies with the code of conduct of .NET foundation and works well with ReSharper, CodeRush, TestDriven.NET, and Xamarin.

How Xamarin Benefits Cross-platform App Development

As Microsoft resolutely states, "Apps built using Xamarin look and feel native, because they are native indeed!" Let's start with this one as we go to describe all the incredible benefits delivered by Xamarin for cross-platform app development.

All things Native

With Xamarin application development, you are confident that any app built using this platform will reflect native-like qualities, such as:

1) Native UI:

Xamarin cross-platform IDE is built with high priority shown to app performance and UI quality. Hence, Xamarin-built apps have native UI controls and look and behaviour that end users truly expect.

2) Native API Access:

Since Xamarin application development provides access to device-specific functionalities and platform-specific capabilities (ARKit and Android Multi-Window mode), developers can foster unique ability to emulate native app technologies used universally.

3) Native Performance:

Apps built using Xamarin technology is enriched with platform-specific speed and are designed to generate organic native performance. This is because during runtime UI components are converted into device-specific elements, avoiding the code interpretation, eventually producing native experience.

4) Fast Pace of development Cycle :

One of the main attractions of Xamarin for cross-platform app development is that its code can be reused and recycled. As up to 90% portion of code can be reused across different platforms, developers find it easy to economize on both time and vigour which they can invest later in other innovations. Thus the simplified code sharing significantly impacts the pace of development cycle.

5) Extract great Value from C#:

Having spent much of time in mobile application development, developers might be familiar enough to use programming languages from the C family. C# which is a much sophisticated, general-purpose language that offers type-friendly comfort for Windows platform. It has object-oriented coding and contains all the improvements and advancements of C++.

6) Shared app Code:

With Xamarin cross-platform app development, it is possible for developers to share app code across multiple mobile platforms and OS. Xamarin has unique edge since its major UI logics such as input validation, database interactions, web service calls, back-end API integrations are written in one language only which is C#. So you can use the same APIs for 75+% of app code across all mobile development platforms.

7) Outstanding flexibility:

Xamarin gifts you outstanding flexibility when it comes to using the same APIs and UI controls as used for development of Android, iOS and Mac applications. You can develop apps easily across these platforms without changing much since Xamarin supports platform specific programming languages viz Java, Objective-C, and Swift. Due to this allowance, users can have new features installed as soon as they are launched for the respective device.

8) Agility and Economy:

This particular benefit appeals to both enterprise-grade businesses and developers equally. The reason why Xamarin cross-platform mobile application development contributes to business growth is its Rapid Mobile App Development (RMAD) capability. All credit goes to Xamarin UI form libraries that enable simplified, swift and frictionless development. It provides developers with an added advantage of agility they need to expedite the process, and businesses, too, benefit from Xamarin's low-cost development model.

9) Great Hardware Compatibility:

Since Xamarin cross-platform development gives you full access to its powerful components and APIs, your app will conduct native interactions with the device OS. This ultimately lowers any compatibility issues you face earlier in the app development. Xamarin gives developers an opportunity to capitalize on and connect with applications' common device functionalities for various platforms on a larger scale.

10) Enhanced Accuracy

Xamarin contains quintessential type-safety feature which is beneficial to developers who want to spend minimum time correcting the errors they face while writing the applications. This rich feature in Xamarin application development allows you to avoid possible type-specific syntax errors and inaccuracies. Developers can simply avoid the involvement of boilerplate or verbose type annotations, thus augmenting convenience and speed, and eliminating occurrences of flaws.

11) Incredible Customization

It is advantageous for any mobile app developers to be able to tweak and add custom feature to applications. Xamarin cross-platform app development tool enables facility for you to use and work on native libraries so that you can create applications with incredible customization.

From above discussion, it is crystal-clear that Xamarin is super-beneficial to both developers and businesses bodies. Not only Xamarin mobile applications offer competitive advantage to enterprise structures, but it also equip developers who wish to work on intuitive, sophisticated and time-tested cross-platform development tools.

Summary

Xamarin mobile application development is a considerable cross-platform solution that offers simplified coding and native user experience across multiple device platforms. The robust cross-platform tool helps developers build a single app, reuse code for common business logic and optimize the speed of app development cycle.

Due to rapid app development capability, hardware compatibility and API flexibility, Xamarin is a platform of choice for modern developers who seek real-time value while building, testing and deploying apps with just one tool. Finally, we can firmly say that a wide community of talented Xamarin professionals and the use of advanced C# definitely make Xamarin a comprehensive application development platform for both business enterprises and developers.

Contact Us

125/2, Sainiketan Colony,
Kalas road, Vishrantwadi,
Pune, Maharashtra-411015,
Email : info@redbytes.in
Fax : 904 701 6310
+91 8113 869 000