Power, Speed and Quality

Key Strategies for Mobile Excellence

Xamarin

Introduction to Mobile

Post-PC devices are the fastest growing and most disruptive technological innovation of our time. Smartphone adoption is happening ten times faster than the PC boom of the 80s, two times faster than the internet boom of the 90s, and three times faster than the recent social networking explosion[1]. Smartphone shipments eclipsed PC shipments in less than two years, and started outselling PCs two to one in early 2012[2]. Tablet growth is happening at an even faster pace, with iPad adoption outpacing the iPhone's growth rate three to one[3].

Mobile software is enabling a new generation of context-aware applications that were not possible in the pre-mobile world, transforming customer relationships and business processes. Your smartphone is always with you: it knows where you are, it has access to a wealth of personal and corporate data sources, and it drives more communication engagement than previous online methods—97% of text messages are read vs 15% of email[4]. Successful businesses are creating mobile-unique experiences to enhance productivity and enable new kinds of engagement—from redefining retail to making the full power of a company's backend computing infrastructure available to employees anywhere.

By 2016, 70% of the mobile workforce will have a smartphone, and 90% of enterprises will have two or more platforms to support[5]

The impact on business is profound. Facebook experienced a 54% year-over-year growth in mobile active users, which account for 30% of its year-over-year revenue growth[6]. U.S. mobile commerce is now growing nearly three times faster than transactions from desktop PCs[7].

Companies that embrace mobility today have a vital opportunity to expand the value and reach of their business, while those that are slow to act will lose relevance as more agile competitors jump ahead.

Users Demand Native App Experiences

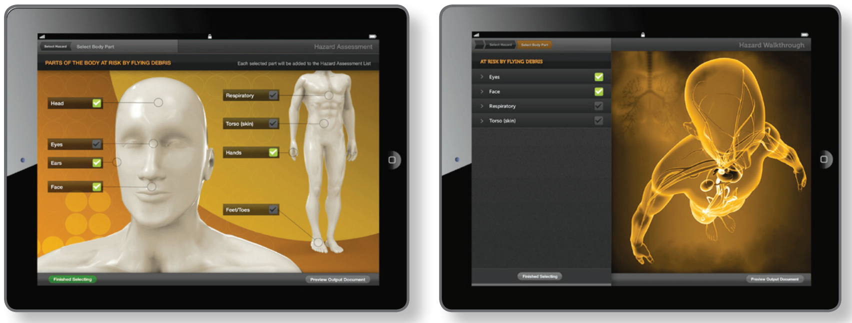
Whether accessing past order history on a sales call or checking the schedule to catch the next train, users expect mobile software to make contextually-relevant information accessible instantly, in a format that takes full advantage of the device's capabilities and form factor. Applications that can't fulfill a desired function promptly and intuitively simply don't get used.

There are many factors that contribute to whether or not users engage with an app, including performance, reliability, design, and ease of use. In all of those areas, a fully native application has significant advantages over non-native implementations.

Defining Native: Three Important App Characteristics

1. Native applications are built with standard, native user interface controls in a manner that fully conforms with each platform's design conventions.
2. Native applications offer optimal performance, leveraging platform-level hardware acceleration to deliver unmatched responsiveness.
3. Native applications have access to the full spectrum of functionality exposed by the underlying platform and device, including platform-specific capabilities such as stylus support and face tracking.

Device platform vendors are fiercely engaged in a battle over market share. What is at stake here is not just today's smartphone and tablet revenue. The battle is over capturing a vast digital economy that will be conducted on devices used in the living room, workplace, car, and even as apparel as new wearable device categories emerge. This intense competition between platform vendors is greatly accelerating innovation—devices are becoming faster and smarter, with vendors adding differentiating capabilities at an astonishing pace. Businesses that are quick to integrate these new capabilities in their customer interactions and business processes command a significant competitive advantage.

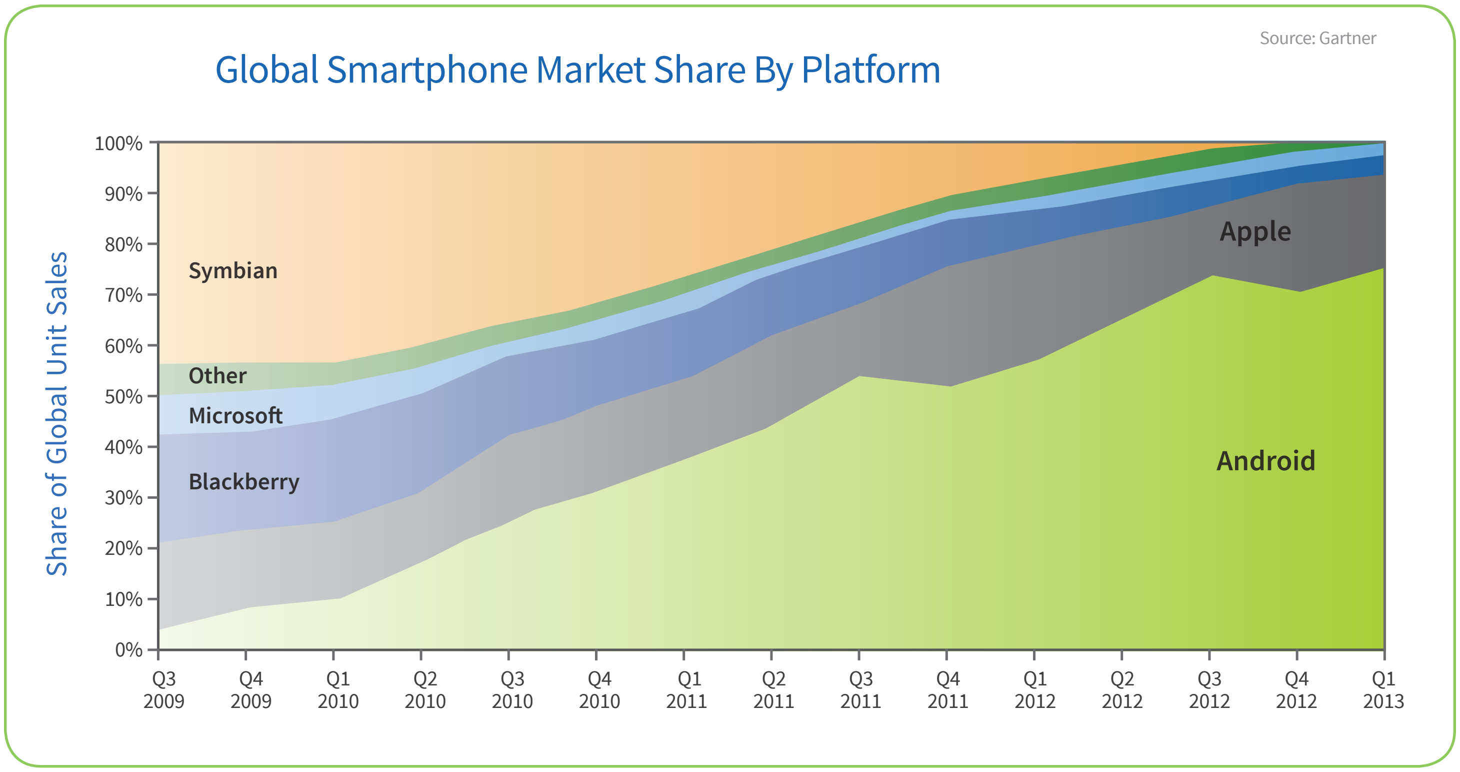


Native iPad app for Kimberly-Clark's field sales team, built using Xamarin

The Diverse Device Landscape

The mobile computing landscape is heterogeneous and is far more fluid and fragmented than the the prior PC era. From 2009 to 2012, Android vaulted from 4% smartphone market share to a whopping 66%. While Android dominates in device volume, Apple remains king in user engagement with iOS users spending over twice as much time on device[8].

In order to formulate a sustainable mobile strategy that will outlast the latest trends, companies need a cross-platform approach that enables them to take the high ground in the escalating platform wars.



Current Cross-Platform Approaches

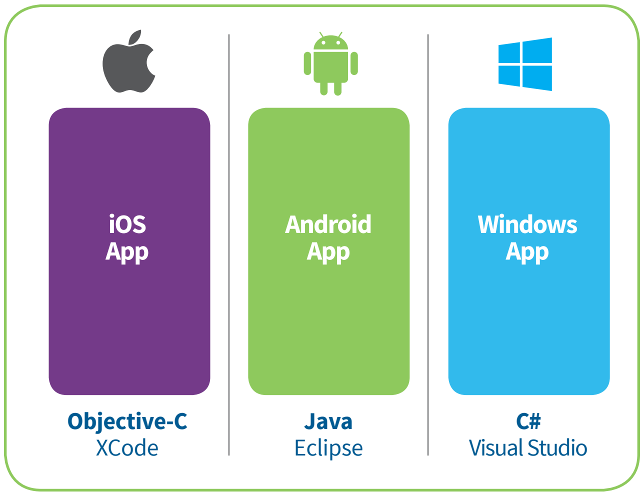
Because the mobile device landscape is heterogeneous and highly dynamic, companies are faced with the challenge of reaching multiple mobile operating systems with a single app.

Unfortunately, current approaches to cross-platform mobile development have given the whole concept a bad name. Let's take a moment to examine the problems with current approaches.

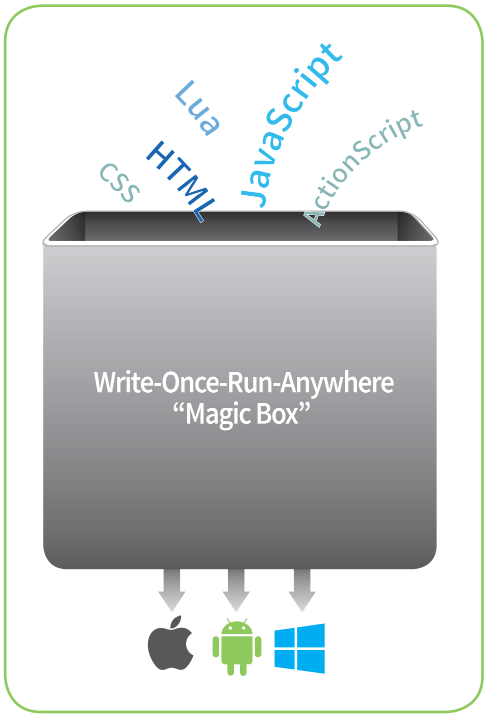
Write Apps Three Times

Amazingly, one of the most common techniques used by companies today to build mobile apps for multiple mobile device categories is to rewrite the app from scratch for each vendor's mobile operating system.

The overhead of this approach slows innovation which hurts both businesses and end users



Companies taking this "vendor-specific silo" approach to mobile development have to cope with separate languages and tools, different teams with different expertise, and the friction of having to implement every feature multiple times. If they want to support any other emerging device platform—such as Samsung's Tizen—they'll have to replicate the entire app again. Building applications in each platform's language and toolset comes at a great cost.



The "Magic Box" Approach

Another method taken by many cross-platform framework vendors is the write-once-run-anywhere (WORA) approach. The idea of WORA app development is that you write your app once, in a single codebase, and then drop it into a "magic box" which adapts the app to the operating system and form factor of each device. This approach is familiar to people who have experience with Adobe Air, Java SWING, and other cross-platform toolkits.

Abstraction frameworks can't express the patterns and variations of each platform's design language

Particularly with mobile, users rely on platform-specific stylistic cues to guide their interactions with software. These cues are stripped in apps built with "Magic Box" tools. The result is a disonnant, lowest common denominator user experience that threatens user adoption and puts the business objectives behind the app project at risk.

Xamarin's Unique Approach

Overview

Xamarin is redefining cross-platform development and enabling the best of both worlds—providing the advantages of native UI, access to device features, and native performance coupled with the time-to-market advantages of code sharing and reuse. Companies that build mobile software today with Xamarin achieve cost-effective cross-platform development and have the flexibility to handle any app use case and user experience requirement.

Xamarin's new model for cross-platform development:

- **Accelerate time to market:** Xamarin enables significant code reuse, reducing the time and overhead of producing rich applications for multiple platforms.
- **Start now with existing teams:** developers who are at home with C# and Visual Studio are productive with Xamarin from day one. There's no need to learn a new programming language or hire developers with specialized expertise.
- **Engaging, native experiences:** build rich applications that deliver optimal performance and leverage every last inch of functionality exposed by the underlying platform and device.
- **Integrate with existing enterprise architecture:** leverage .NET's extensive framework of libraries for calling web services and interacting with data sources, and share the same application logic written in C# across client and server.

"The results from our new field sales app are phenomenal—our sales people love the app and are able to engage customers and close sales more effectively. Key to the app's success is the beautiful, fast user experience made possible by Xamarin."

Kimberly-Clark

Kim MacDougall | Senior Capability Manager

Native Capabilities

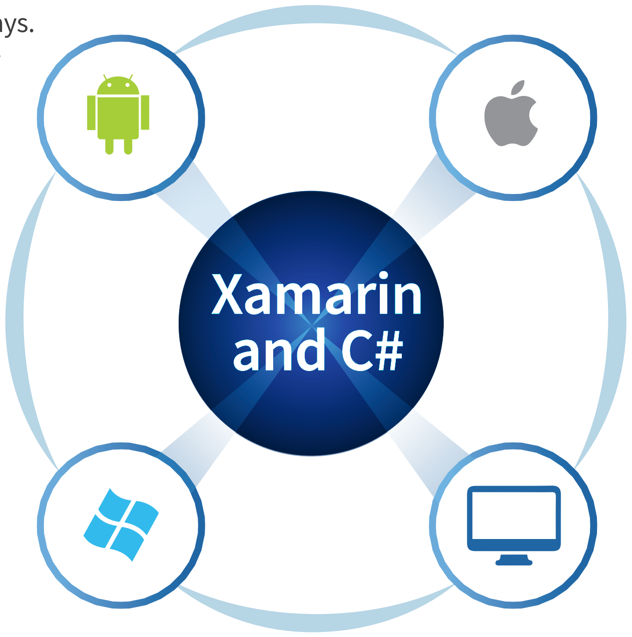
Xamarin apps look native because they are native. Through our unique binding technology, developers have complete access in C# to all of the same APIs and user interface controls used to build iOS, Android and Mac apps in the platform-specific languages. The full feature set of the underlying platform is exposed, including capabilities like face tracking, calendar integration, NFC and Core Bluetooth—providing support for any app use case or user experience requirement.

If you can do it in Objective-C and Java, you can do it in C# with Xamarin

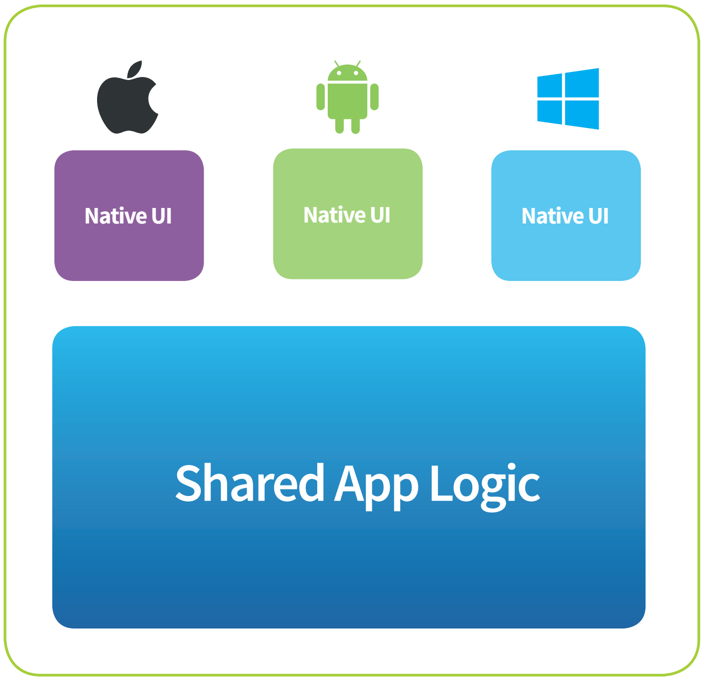
Xamarin's binding technology makes it possible for Xamarin to quickly deliver support for new features as they are introduced in the device operating systems. Xamarin released updates for iOS 5.0, iOS 6.0, and iOS 6.1 all within 12 hours of their public release—providing full support for new platform capabilities such as PassKit and full compatibility with new device features such as the larger screen size on the iPhone 5. In the highly competitive mobile application landscape, users increasingly expect apps to support the latest features on day one. With Xamarin, you never have to worry about getting left behind.

Reach 2.6 Billion Devices with Existing Teams and Code

Xamarin provides a frictionless glide path for migrating existing C# skills, teams, tools and code to the world's most popular mobile platforms—making it possible to use C# to reach 2.6 billion devices. Companies that transition their existing .NET teams to Xamarin achieve mobile productivity within a matter of days. Xamarin makes it easy for any C# developer to become a mobile developer, eliminating the need to staff multiple, platform-specific teams.



Companies that initially built apps in platform-specific languages are now turning to Xamarin as a scalable solution for achieving broad reach across platforms without sacrificing native user experiences and performance. This combination of rich, fully native development and extensive code sharing across device platforms is not possible with any other cross-platform solution.



The Xamarin Approach

Write apps entirely in C# with access to 100% of each platform's APIs

Deliver a device-specific native user interface while sharing an average of 75% app logic code across device platforms

Compile apps as native binaries for fast performance

Code Sharing Advantages

When building software with Xamarin, developers create distinct native user interfaces with platform-specific elements, while sharing their application logic—such as input validation, web service calls, database interactions, and backend enterprise integrations—across operating systems. Xamarin developers share an average of 75% of their code across platforms. This means that all of the code below the user interface layer is written once and reused—reducing the surface area for testing and leaving less room for bugs.

Xamarin also gives developers a practical path for extending the reach of their existing .NET skills and code to modern mobile environments. Companies that have existing desktop and web applications built with C# use Xamarin to bring much of the underlying application logic to Android and iOS. Xamarin makes it possible for organizations to leverage their existing C# investment instead of starting over from scratch.

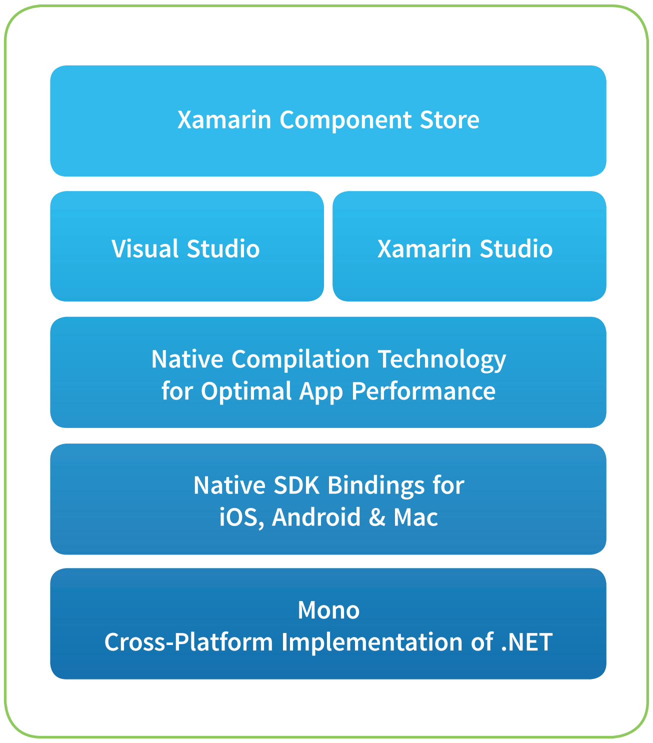
Xamarin is a particularly compelling choice for developers who want to build mobile frontends for existing ASP.NET applications. In addition to supporting all of the native controls exposed by the underlying platform, Xamarin also makes it easy for developers to build hybrid applications that display HTML content in native web view controls. It's even possible to use ASP.NET's Razor templating engine in a Xamarin mobile application to generate HTML on the client. In some cases, it's possible to share the same exact data models between the frontend and the backend.

This immensely powerful blend of native platform functionality and .NET libraries offers developers the perfect toolset, accelerating time-to-market and enabling amazing user experiences.

Xamarin Platform

Overview

Xamarin offers a compelling blend of powerful developer tools, efficient cross-platform code portability, full access to the underlying device platform APIs, and a rich ecosystem of third-party libraries. Those characteristics make Xamarin a great choice for modern mobile developers who want to build successful applications that reach users on every major platform.


Xamarin Component Store
Visual Studio
Xamarin Studio
Native Compilation Technology for Optimal App Performance
Native SDK Bindings for iOS, Android & Mac
Mono Cross-Platform Implementation of .NET

Rich IDE Support

With support for both Visual Studio and Xamarin's own Xamarin Studio IDE developers have everything they need to design, develop, debug and deploy great mobile apps.

Xamarin integrates with Visual Studio, making it possible to build iOS and Android apps in Microsoft's preeminent development environment. Visual Studio users finally have the freedom to write code for all major platforms under one roof—using the programming language they know and love—with convenient access to Microsoft's ecosystem of extensions and tools like Resharper and Team Foundation Server.

Xamarin also comes with Xamarin Studio, a powerful cross-platform IDE for mobile development on Mac and Windows for developers who do not use Visual Studio. Xamarin Studio combines a strong foundation of general-purpose C# programming capabilities, specialized mobile development features, and tight integration with Xamarin's frameworks and build toolchain. The result is an IDE tailored for building apps with Xamarin, offering a good balance of power, extensibility, performance, and ease of use.

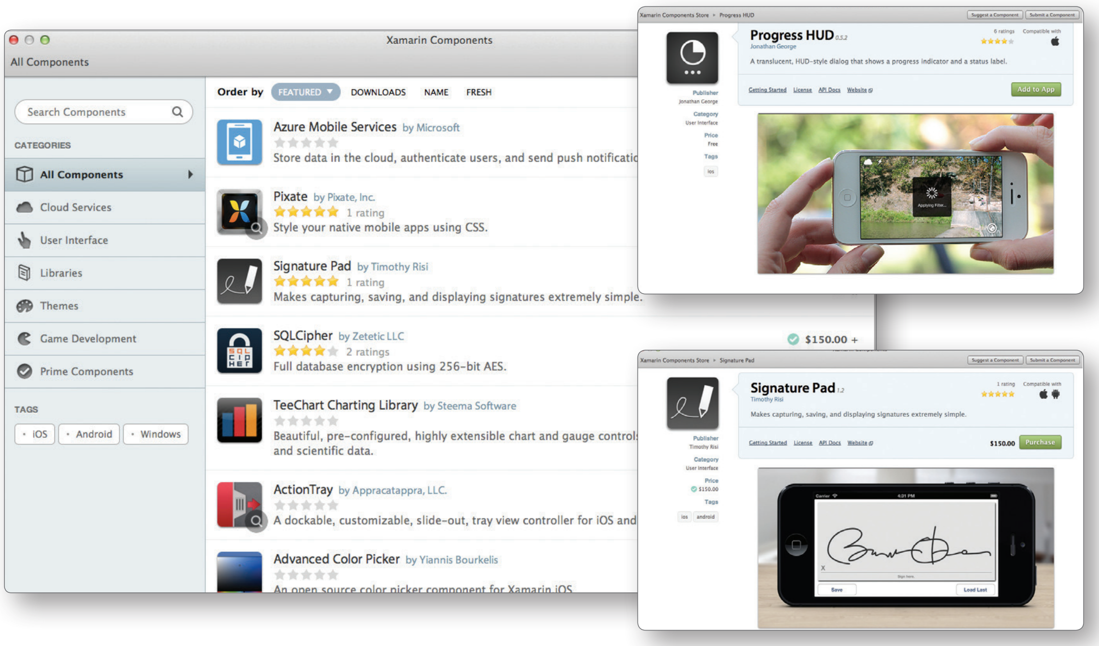
Superior tools increase agility and enable developers to focus on innovation and quality—great tools lead to great apps

Xamarin's IDE support includes a built-in design surface for building Android user interfaces and a new iOS designer for building layouts and configuring transitions between views. These design tools leverage the same conventions and file formats as the design tools supplied by Apple and Google. They enable drag-and-drop user interface construction and are tightly integrated with the Xamarin Component Store and the rest of the Xamarin toolset.

Xamarin Studio and Xamarin's Visual Studio extensions include advanced debugging tools that work across the full spectrum of supported platforms and environments. Developers perform interactive debugging on an application that is running in the Android emulator, the iOS simulator, or even directly on hardware. Xamarin's debugger supports all of the modern features that users expect—setting breakpoints, inputting watch expressions, and accessing local scope variables at runtime. Xamarin also supports a number of advanced debugging features, such as conditional breakpoints.

Component Store

The Xamarin Component Store is a searchable catalog of free and paid components that add beautiful UI controls, libraries, and 3rd party web services to apps with a few lines of code. If an application requires complex features like bar-code scanning or a sophisticated user interface control like a signature pad, developers can simply install the desired components and then focus on wiring them together with their own application-specific glue code.



Azure Mobile Services by Microsoft
Store data in the cloud, authenticate users, and send push notificatio

Pixate by Pixate, Inc. 1 rating
Style your native mobile apps using CSS.

Signature Pad by Timothy Risi 1 rating
Makes capturing, saving, and displaying signatures extremely simple.

SQLCipher by Zetetic LLC 2 ratings $150.00 +
Full database encryption using 256-bit AES.

TeeChart Charting Library by Steema Software
Beautiful, pre-configured, highly extensible chart and gauge controls and scientific data.

ActionTray by Appracatappra, LLC.
A dockable, customizable, slide-out, tray view controller for iOS and

Advanced Color Picker by Yiannis Bourkelis
An open source color picker component for Xamarin.iOS

Progress HUD 0.5.2
Jonathan George
6 ratings
Compatible with
A translucent, HUD-style dialog that shows a progress indicator and a status label.
Getting Started License API Docs Website
Publisher: Jonathan George
Category: User Interface
Price: Free
Tags: ios

Signature Pad 1.2
Timothy Risi
1 rating
Compatible with
Makes capturing, saving, and displaying signatures extremely simple.
Getting Started License API Docs Website
$150.00
Publisher: Timothy Risi
Category: User Interface
Price: $150.00
Tags: ios android

There are a variety of user interface controls available in the Component Store, such as a graphing view and a Path-inspired satellite menu. Cross-platform libraries are also available, making it easy to integrate authentication, social network sharing and other popular SDKs into apps. The Component Store also makes it easy to integrate with hosted backend services like Parse and Azure Mobile Services.

The Component Store is built right into Xamarin Studio and Xamarin's Visual Studio extensions, making it possible for developers to find, use, and manage components from within the IDE. The Xamarin Component Store is carefully curated by Xamarin, ensuring that all components work reliably, and include documentation and sample projects that demonstrate proper use.

Enterprise Class Support, Training, and Services

Xamarin has a thriving ecosystem of support and services offerings that that accelerates app development for mission-critical projects. Xamarin's support offerings include response SLAs, access to the latest hotfixes, technical training and access to resources for code troubleshooting.

In addition, Xamarin offers technical training courses that help developers understand core mobile development concepts, the nuances of each device platform, how to maximize code sharing and optimize performance, among other topics.

Customers can also tap into the worldwide network of Premier and Authorized Xamarin Consulting Partners. The Consulting Partners program includes over fifty partners that have first-hand experience helping clients ship great apps with Xamarin. Additionally, Xamarin offers a range consulting services—delivered by Xamarin Premier Consulting Partners—designed to accelerate development at any stage of the app lifecycle.

Fundamentals and Advanced training classes provide excellent preperation for Xamarin Mobile Developer Certifications

How it Works

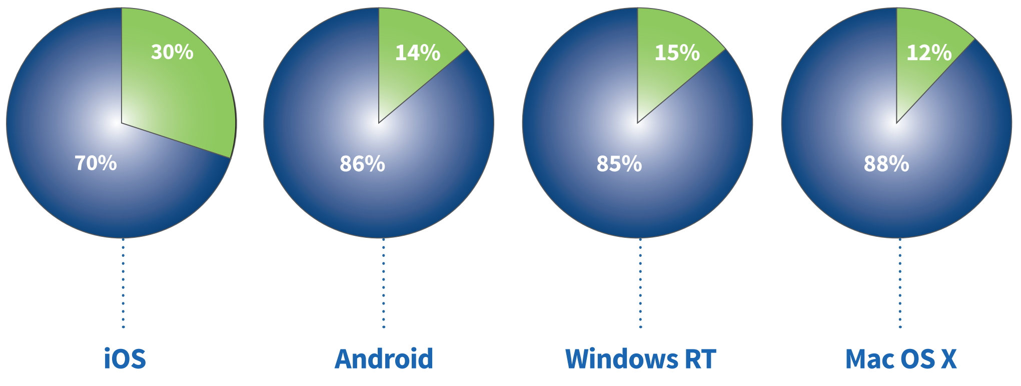
App Architecture

The best practice when building mobile applications with Xamarin is to isolate core application logic in a portable layer of platform-neutral code that uses .NET frameworks and APIs. The developer then builds a separate user interface for each platform on top of the shared code, using native controls and native platform APIs that Xamarin exposes in C#. When done properly, only a small amount of platform-specific code is typically needed to implement the user interface behavior for each platform and bind it to the functionality from the shared library.

Structuring a Xamarin application for maximum code reuse is easy. There are a number of well-understood design patterns that developers can adopt to ensure that their user interface layer is optimally decoupled from application logic. Some common approaches include Model View ViewModel (MVVM) pattern endorsed by Microsoft or the more traditional Model View Controller (MVC) pattern.

Independent analyses performed by Xamarin customers offer insight into the amount of code that can typically be shared across platforms when building applications with Xamarin. Using an automated tool that measures code reuse, developer Frank Krueger determined that the Android, iOS, Windows RT and Mac OS X ports of his sophisticated iCircuit application share between 70 and 88% source code across platforms.

Code Sharing Statistics from the iCircuit App



🟦 App Logic 🟩 User Interface Code

Full API coverage

Xamarin provides access to 100% of the native APIs available on each mobile platform, ensuring that developers aren't confined to an arbitrary subset of features. The C# bindings provide 1:1 mappings to native APIs, while adding support for features like LINQ, delegates, events, and other idioms C# developers expect.

Fully Extensible

The .NET base class libraries offer support for a wide range of common operations, including networking, data serialization and persistence, file I/O, text processing, threading, and more. Between portable .NET libraries and platform-specific libraries with C# bindings, Xamarin application developers have access to an extraordinarily rich ecosystem of third-party code.

The Xamarin Component Store makes a tremendous range of third-party libraries and SDKs available for drop-in adoption in Xamarin projects, but it's also very easy to generate Xamarin bindings for existing third-party Java and Objective-C libraries. Xamarin provides tools, such as Objective Sharpie, that largely automate the process and generate high-quality bindings that are idiomatically copacetic with C#.

There are also a growing number of libraries that provide universal C# abstraction layers over platform-specific capabilities. Some examples include PushSharp—a cross-platform library for push notification—and Xamarin's own Xamarin.Mobile library—which provides cross-platform support for features like geolocation and the system address book.

"Xamarin offers the best of all worlds. We deliver performance-driven native apps that, until Xamarin, were only possible with Objective-C and Java. Sharing over 50,000 lines of code across platforms gives us more time to spend on great user experiences."

rdio™

Matt Crocker | Senior Software Engineer

Production-Proven Technology for Mission-Critical Apps

Xamarin is built on top of Mono—an open source implementation of the .NET runtime and framework class libraries, and a C# compiler. Mono is highly mature with over a decade of innovation and successful production use in a variety of environments, from embedded devices to enterprise backends. Xamarin's advanced compiler technology uses highly sophisticated techniques to ensure optimal performance and reliability while bringing the power and stability of Mono to iOS, Android, and Mac devices.

High Performance and Low Memory Footprint

The Mono runtime—combined with Xamarin's advanced compilation technology—offers excellent performance. Xamarin applications handle computationally-intensive operations at speeds comparable to that of iOS and Android applications built with Objective-C and Java. In fact, internal Xamarin benchmarks suggest that Xamarin C# code can often outperform equivalent Android Java code.

Xamarin provides rich support for threading and parallelism, allowing developers to build applications that utilize the limited processing power of mobile devices as effectively as possible. By using the standard user interface controls supplied by the underlying platform, applications built with Xamarin achieve optimal responsiveness and benefit from platform-level support for hardware-accelerated rendering.

Xamarin's Compilation Technology

Unlike other cross-platform frameworks, Xamarin apps are compiled as native binaries, not interpreted. This results in high performance apps under even the most demanding scenarios like complex data visualizations and high frame-rate simulations.

Xamarin.iOS uses Ahead of Time (AOT) compilation, which preemptively compiles an application down to a native ARM executable during the build process. When a C# application is built with full AOT compilation, the output is an architecture-specific binary executable that contains no bytecode and does not require JIT compilation when it runs. The Xamarin.iOS compilation toolchain uses the LLVM optimizing compiler, the same compiler that Apple ships in the official iOS SDK. The user's application code, the core of the Mono runtime, and the frameworks and libraries on which the application depends, are all statically compiled into the binary as native code. From Apple's perspective, an iOS application built with Xamarin is just like any other native application that runs on the platform.

Android has its own runtime environment called Dalvik, which is used to execute conventional Java-based Android applications. Much of Android's platform-specific functionality is made accessible through Java APIs that are tied to the Dalvik environment.

Xamarin.Android applications use the Mono and Dalvik runtimes at the same time, allowing developers to take advantage of the full range of functionality offered by both environments. Mono itself operates on Android much as it would in any other computing environment—it uses standard Mono JIT compilation. Fully transparent interaction between Mono and Dalvik is made possible by leveraging standard Java Native Interface (JNI) calls. Android's open platform model enables Xamarin to leverage the full power of Mono while providing deep interoperability with the platform's native development stack. As a result, Xamarin applications run just like regular Java-based Android applications, but with full access to all of the functionality provided by Mono.

"Our customers rely on National Instruments for their mission- critical systems, and we in turn rely on Xamarin to extend our capabilities to mobile devices. Xamarin enabled our engineering team to become native mobile developers almost overnight."

NATIONAL INSTRUMENTS™

David Fuller | VP, Application and Embedded Software R&D

The Formula for Mobile Excellence

Mobile technology is reshaping the way that people connect, work, and play. It enables new kinds of customer engagement, makes distributed workforces vastly more efficient, and further lubricaties the flow of data that serves as the lifeblood of our information economy.

Connected devices have implications for companies across the entire spectrum, not just those in the technology industry. The importance of mobile technology will continue to increase as hardware improvements and declining costs propel us into a world where connectivity is present in everything around us. We're at the start of a virtuous cycle where the growing pervasiveness of mobile integration is making mobile technology more vital and indispensable, driving more adoption and more integration. No company can afford to fall behind.

Businesses in the modern mobile era now have a solution that provides resiliency against industry changes, while providing the flexibility to leverage the emerging capabilities of top mobile devices

Companies that have adopted a Xamarin-based mobile strategy are winning, rapidly delivering quality, platform-specific experiences to customers, employees and partners—apps that take full advantage of the latest capabilities across a heterogeneous device landscape. Enterprises that are still formulating their mobile strategy can rely on Xamarin to accelerate their ability to move from strategy to this level of mobile excellence.

References

[1] http://blog.flurry.com/bid/88867/iOS-and-Android-Adoption-Explodes-Internationally

[2] https://intelligence.businessinsider.com/smartphones-outpacing-pcs-two-to-one-2012-11

[3] http://www.slideshare.net/kleinerperkins/kpcb-internet-trends-2013

[4] https://www.digitalmarketingassoc.com/social-media/closer-mobile-marketing/

[5] http://www.gartner.com/technology/research/symposium-keynotes/
http://www.crn.in/news/software/2012/10/26/gartner-two-thirds-of-enterprises-will-adopt-mobile-device-management-solution

[6] http://techcrunch.com/2013/05/01/facebook-sees-26-year-over-year-growth-in-daus-23-in-maus-mobile-54/
(Facebook Q1 2013 earnings)

[6] http://au.businessinsider.com/mobile-is-taking-more-e-commerce-2013-5

[8] http://www.netmarketshare.com/

About Xamarin

Xamarin was founded in 2011 by Nat Friedman and Miguel de Icaza, technology veterans with a successful track record in enterprise software. The company employs a passionate team of 70 highly skilled software engineers, has a robust training and developer certification program, and world-class support and documentation.

Xamarin's worldwide partner network is comprised of over 50 global system integrators, national and regional consulting companies, and digital agencies who have deep experience building enterprise and consumer apps with Xamarin.

Fulfilling the promise of cross-platform mobile development without compromises, Xamarin's breakthrough products have attracted a community of over 350,000 developers. Xamarin today has over 17,000 paying customers, including Microsoft, Kimberly-Clark, Clear Channel, Schindler, McKesson, Bosch, Halliburton, Cognizant, GitHub, Rdio, and WebMD.