INFRAGISTICS

Xamarin and Xamarin.Forms:

MULTI-PLATFORM APP DEVELOPMENT THE SMART WAY

Introduction

As of July 2014, data from Statista showed there were well over one million apps in the Apple and Google app stores and more than three hundred thousand in the Windows Phone Store. Enterprise app growth is similarly positive, Gartner predict that by 2017 25% of businesses will have their very own Enterprise app store. Employees are to be given instant mobile access to their companies' CRMs, Intranets and collaboration systems with a significant boost in productivity expected.

The rise of enterprise apps that help with business processes is one of the most remarkable trends in the professional technology market in recent years and is set to keep on growing. More and more large-scale enterprises are opting for internal app development. It is becoming clear that, more than being a simple fashion, apps are part of a global shift in business practices. Technology has allowed employees to work on the move and from a distance more easily, and outsourcing and freelancing are increasingly common human resourcing practices. Apps play a central role in this movement.

However, given the huge range of mobile devices and platforms available, developing customized apps for businesses and their specific requirements is far from a simple task. Except in the unlikely scenario that all employees use the same mobile devices, developers will have to go through the complex and time consuming process of cross-platform development - writing different apps on iOS, Android and Windows Phone.

As this Whitepaper establishes, cross platform development can be significantly improved by employing Xamarin. Utilizing the C# language, Xamarin allows a single code base to be used for all of the major mobile platforms. What is more, combining Xamarin with Xamarin.Forms not only allows an application's business logic to be shared between Windows Phone, iOS and Android, but the UI design and logic as well.

This Whitepaper will look in detail at the value Xamarin and Xamarin.Forms can add to the app development process, and how they help companies build cross platform apps more effectively.

The Business Benefits of Enterprise Apps

Since Apple first opened their app store in 2008 there has been an explosion in app development for almost every conceivable use. From games to entertainment to business tools, developers are called on to build apps for a vast range of needs and activities. The enterprise market in particular has been quick to recognize the business case for internal apps in order to share internal news and comms, access systems and increase employee engagement. Developers are increasingly being asked to build tools for internal data analysis, Business Intelligence, CRMs, collaboration, HR portals and beyond.

Whatever the business process or use case, companies are increasingly aware that apps can no longer be seen as an optional extra. Whether as a means of increasing revenues and productivity, spreading awareness of products and services, and promoting employee engagement- apps have a critical role to play.

Since employees and consumers will likely download and access an app from a range of platforms – from iPads to Windows Phones to Android devices – widespread adoption depends on availability on these different devices. Apps must therefore be available on all platforms but also need to be easy to use and feel natural on each. It would be frustrating for iOS users to have to wait months for an update already received by colleagues with Android devices for instance, yet this is very hard to achieve. While offering internal apps to employees has clear benefits, it is not surprising that many companies are put off by the complexities involved in building them. shared between Windows Phone, iOS and Android, but the UI design and logic as well.

Each Platform Has Its Own Language and Related Development Tools

Traditional cross platform development can be highly complex, require a large upfront investment, and most significantly, considerable ongoing maintenance effort. It is expensive, time consuming and inefficient for a number of reasons:

1 Because developing for the different platforms requires developers to write code in different languages, a company hoping to launch an app simultaneously on each of the major platforms would have to hire three separate teams excelling in:

- C# for Windows Phone
- Java for Android
- Objective C or Swift for iOS

Engaging an extensive group of professionals to build, test and then maintain three sepvarate codebases comes at a high cost and can easily cause what started off as a simple project to run over.

2 The Windows Phone, Android and iOS operating systems each have their own particular interfaces and features. Each one allows users to interact with apps differently and this requires different considerations on the part of the developer. Apps need to consider, if not mirror, the specific UI characteristics of each if they are to feel 'right'. This can require a lengthy process of design, implementation and testing.

3 Testing and comparing the functionality and UI of apps written in different languages is complex. App developers need to ensure their builds work correctly in terms of performance and functionality on many different systems. Reviewing thousands of lines of code and finding bugs is, of course, painstaking and time consuming.

4 Besides the not insignificant issue of costs, maintaining different codebases over the lifetime of an app can be a major investment. Each app version will require updates and bug fixes. These need to be coded and maintained separately.

Options for Cross Platform Development

One of the principal means for addressing issues with cross platform app development has been to use traditional web technologies. Essentially a web site is wrapped up in a dedicated mobile app 'container'. The app container is specific to that platform, but the core web based components can be shared amongst platforms.

There are a variety of frameworks that allow developers to build apps in HTML and JavaScript and then host the results in an app container. Tools such as Infragistics' IgniteUI help in the creation of responsive web based designs.

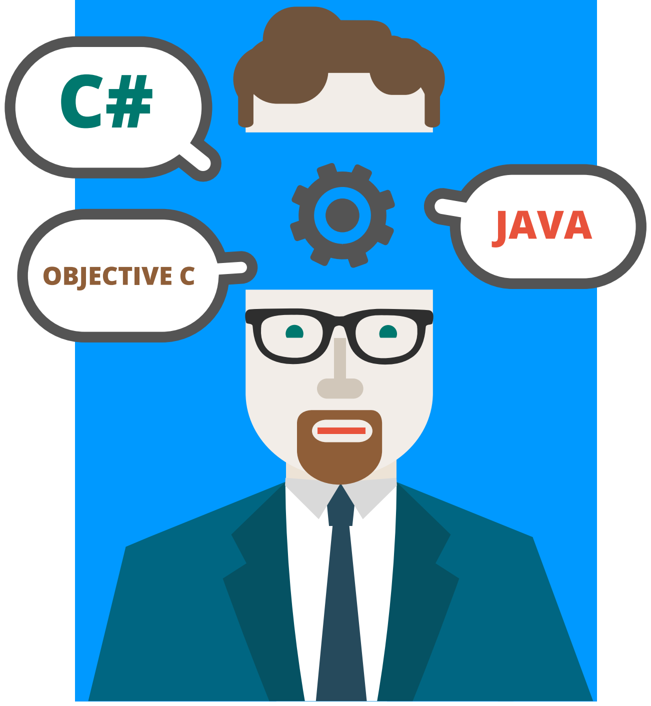
The advantage of a web based approach is that everything is built from one codebase and allows the app to reuse existing website assets. Essentially, end users download an app, which lets them view web based content, tailored to their device. Such tech giants as Facebook used this approach in the early days of their app strategy to provide a cross platform app, which shared many common assets.

With the above approach, developers still face issues relating to the app's performance on different devices. This is unsatisfactory, since with more complex apps, performance will vary between operating systems and the capabilities of the devices that run them and mean different users will have very different experiences of the tool.

On a more technical level, the tools and languages used in this kind of development – HTML and JavaScript – are geared for web development and while powerful, still do not correspond with the specific characteristics of mobile devices. Web hybrid development means performance is lower than a platform targeted app and means end users can often feel frustrated. Facebook eventually rowed back from this approach because its apps simply didn't perform in terms of speed and functionality.

A more targeted approach was needed.

Cross platform app development is a highly complex discipline for app developers, requiring specialist knowledge of different languages and toolsets.

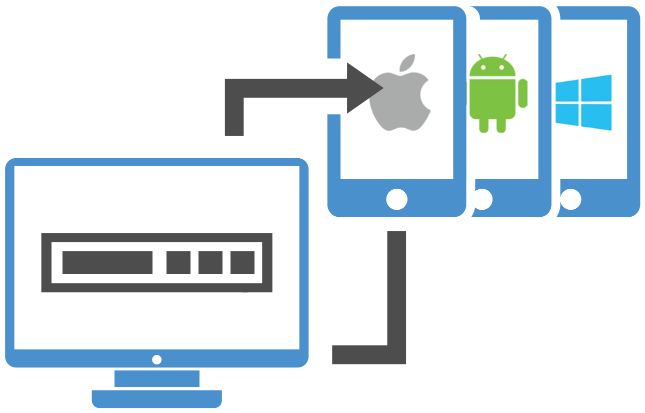


Refocus App Development with Xamarin

As we have established, cross platform mobile app development can be hard to get right but Xamarin offers a real alternative. It presents a compelling case for any organization hoping to develop apps quickly, affordably and in a way that can be easily maintained and updated across different operating systems. In contrast to other multi-platform app solutions, Xamarin allows native apps to be written in one language (C#), whilst still having access to the unique features and APIs of the host device.

A Single Development Language (C#)

Xamarin's key strength is that it allows developers to write apps using a single language, the widely adopted and industry standard C#. Rather than building apps using three languages for three different systems, Xamarin considerably simplifies the development process. Furthermore, while the code may not be written in the native languages of each operating system, apps built in C# with Xamarin have the same speed and functionality of apps that are. Xamarin even provides access to a full range of platform specific APIs.



With the C# language at its core, Xamarin makes it easy to develop any mobile app using Visual Studio. This industry standard development environment is known and loved by developers the world over, especially those working in an enterprise setting. Both C# and Visual Studio are well supported online, with a wealth of community forums, experts and add-on software. Xamarin lets developers take advantage of all of this.

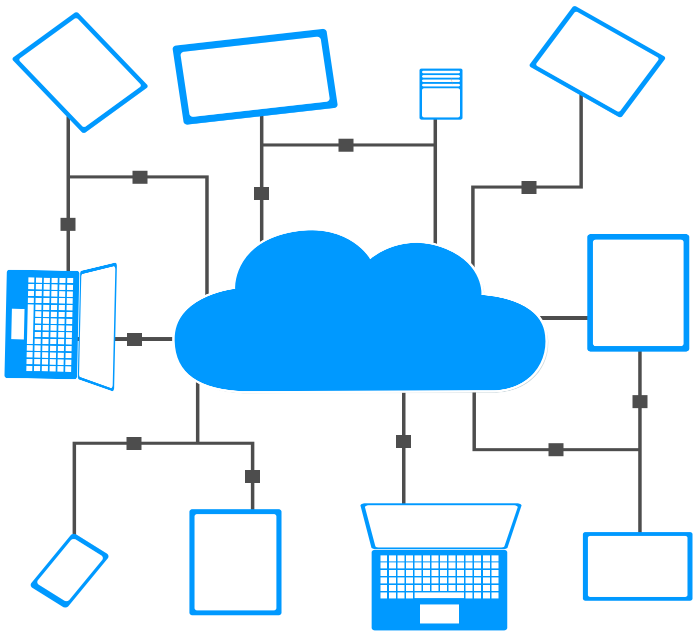
Drawing on the Strengths of Different Systems

On a global scale, Android is by far the most widely used operating system on mobile devices, followed by iOS and then Windows Phone . Windows Phone natively uses C# and Xamarin extends this support to Android and iOS app development. This means developers can share a large proportion of their application's code base across the three most popular operating systems on the market today.

Xamarin achieves this by making native APIs available for each platform, via C#. This means developers can use the specific features of each platform, creating apps that end users will feel instantly at home with, as if built specifically for their device. As users navigate around the app, as they pinch, flick and tap on their mobile or tablet screens, the app will feel 'normal' for that system. This is something that just isn't possible using other techniques, such as the web app 'containers' described above.

Test Builds on Thousands of Devices in the Cloud

Being confident that an application is user-ready prior to release is an essential stage in the development process. Discovering a bug or error once users have already installed an app can have serious consequences for the app's success. Business users will be put off by a faulty app and this can cause frustration and disengagement – feelings that internal apps are specifically intended to overcome. While having to release a patch for a failed build is not a disaster, it costs time and money and impacts ROI.



Good testing is therefore essential. However, having to maintain multiple codebases makes any testing harder since there is simply more code to test. Testing these different versions across an even wider spectrum of hardware devices makes the problem even harder.

With Xamarin's Cloud Testing service, developers are able to access over one thousand popular devices where they can test their builds in advance of release. The service offers a simplified approach with which to visualize and explore how an app would appear on different operating systems and how it would physically sit on devices of different shapes and sizes. This is particularly useful with the Android platform, due to the great variation in operating system versions, hardware features, and screen sizes.

The Xamarin Cloud Testing Service Allows Users to:

- Check for bugs
- Produce reports to alert developers of potential problems on different systems
- Save time spent manually seeking bugs and locating errors amongst thousands of lines of code
- Produce real-time reports
- Leave more time to focus on actual development.

Xamarin's Broad Benefits

Xamarin offers a real leap forward for app development. It means previous painstaking and complex work replicating apps across platforms can be considerably simplified.

Organizations making use of the Xamarin platform to build multi-platform apps can make savings in a number of areas:

1. Upfront development costs are reduced, as teams can be smaller and more compact – only knowledge of a single programming language is needed.
2. Ongoing app support budgets are reduced because developers only need to maintain and fix one codebase.
3. It is possible to reuse existing tooling, such as Visual Studio and related utilities.
4. Time spent testing is reduced thanks to direct and instant access to thousands of devices on the Cloud meaning developers can instantly run the app from Visual Studio and find bugs directly.

The Xamarin platform makes cross platform mobile app development a reality. It gives companies peace of mind that their apps are performant across operating systems and hardware, that they will use all of the features of the target platforms, and that they can implement a sustainable support strategy within their budgets and ambitions.

Xamarin.Forms – Leverage the Power of Xamarin for UX Design and Implementation

The advantages of the Xamarin platform are significant. Not only does it simplify upfront cross platform app development, but it means that only one application codebase needs to be maintained. This translates as significant time and cost savings over an app's lifetime.

Xamarin.Forms extends Xamarin by allowing developers to share not only application logic, but UI logic and design across platforms too. This provides developers with close control over an application's look and feel on each individual target platform. While Xamarin allows code and assets to be shared amongst mobile app operating systems, Xamarin.Forms does the same for the development of the look and feel of the app.

Working like a translator, the Xamarin.Forms API ensures resulting code is optimized for each platform. For instance, when designing a text field with Xamarin.Forms, the developer would type Entry. Xamarin.Forms then maps this command to platform specific native UI elements – Entry become UITextView in iOS, EditText on Android and a TextBox on Windows Phone.

Xamarin.Forms Takes an Advanced Approach to UI Design and Build

1 Developers can now build apps that share even more code than with Xamarin alone. Previously it would have been necessary to make changes to apps for different platforms when it came to UI and UX. Xamarin.Forms reduces this effort.

2 Xamarin.Forms use XAML as a markup language for expressing UI layout (although simply using C# is also possible). This means any learning curve for Xamarin.Forms is low for experienced Microsoft developers.

3 With Xamarin.Forms, developers have flexibility in how they wish their apps to ultimately look on each platform. It is possible to take an 'adaptive UI' approach, that is to say, each app UI makes use of the idioms of the target platform. As a result, apps look and feel specific to each platform. Alternatively, it is possible to take a 'consistent UI' strategy, meaning the app looks and feels the same across all platforms.

So, for example – if you are building an app for a company Intranet portal, it would be preferable that the app looks and feels the same on Android, iOS and Windows Phone. Users will expect consistency across company devices in terms of navigation and content interactions.

On the other hand, an app aimed more at the consumer market might be better suited to adopting the look and feel of the different platforms - users will rarely see it on another platform.

4 Xamarin.Forms is part of an evolution in app development. It manages to use the strengths of alternative approaches (such as hosting web content inside a platform specific app), but avoids their limitations. For example, Xamarin. Forms lets developers reuse UI design and logic, but with no loss to performance.

5 Developers are not tied to expressing UI content using Xamarin.Forms - they can also express UI content natively using the Xamarin platform. If Xamarin.Forms does not provide a specific piece of functionality they need they can also opt to call platform specific APIs.

Xamarin.Forms – The Core Benefits

The Xamarin.Forms framework, built on top of the power of the Xamarin platform, allows developers to share UI logic and designs between platforms. Developers can be confident that, no matter how they wish individual app designs to manifest on the host platform, the process of creating and maintaining them will be as simple and manageable as possible.

When deadlines are pressing, budgets are limited and it's simply not possible to entirely rebuild a UX for each platform, Xamarin.Forms stands out as an indispensable tool for developers.

Next generation Cross-Platform Mobile App Development

For a long time, the solutions available for companies building cross-platform apps have been largely unsatisfactory. Either they have had to engage different teams to write the same app for different platforms in different languages, or they have been forced to use tools and techniques – like those designed for traditional web development – that didn't quite fit the bill. A better approach was needed.

Xamarin is the natural evolution of cross platform app development. By adopting one language, the time and effort to build apps across all the major mobile platforms is massively reduced.

Xamarin.Forms builds on the powerful architecture of Xamarin and allows UI logic and design to benefit from a single implementation. This not only means apps built using Xamarin function in an optimal way on different platforms, they also look 'right' on different interfaces too (whether 'right' is a consistent look across the board or each app blending in with its native operating system style).

Enterprises using Xamarin and Xamarin.Forms to build cross platform apps will not only see shorter development time and better resulting software, but will be able to support and maintain apps much more effectively over their life time.

Infragistics has the solution

Infragistics is pleased to support the Xamarin platform, and we offer a range of UI controls for use with the Xamarin.Forms framework. By combining Infragistics' expertise in UX with Xamarin's cross platform development architecture. Infragistics' Xamarin.Forms UI Controls go one step further to helping the business app developer achieve their targets.

There's no need to stress over the training and time costs involved with learning a new platform. With Infragistics Xamarin.Forms, developers are able to use their current C# & XAML skills to architect truly native apps in the same way as traditional cross-platform applications: with a single codebase.

To learn more about Infragistics Xamarin.Forms, visit http://www.infragistics.com/products/xamarin-forms, or contact **Senior Vice President of Developer Tools, Jason Beres, at** JasonB@infragistics.com.

REGIONS	TARGET	FORECAST %	FORECAST	ACTUAL	PREVIOUS	YOY
Americas	$1.9M	81	$1.6M	$1.6M	$1.3M	+21%
Asia	$1.2M	84	$995.3K	$966.8K	$770.1K	+26%
Australia	$743.4K	82	$608.4K	$606.3K	$473.8K	+28%
Europe	$950.4K	80	$764.4K	$777.6K	$634.3K	+23%
TOTAL	$4.8M	82	$3.9M	$3.9M	$3.2M	+24%