xpandit

PASSION FOR **EXCELLENCE**

WHITEPAPER

CROSS-PLATFORM DEVELOPMENT WITH XAMARIN

1 Tackling the Cross-Platform Development Challenge

The adoption of mobile devices has been experiencing a huge growth in the last years, outpacing even the recent social networks adoption rate.
One of the biggest challenges of mobility nowadays is the number of platforms that should be considered in mobile initiatives and projects.
The last years led to a consolidation in terms of market leadership that resulted in a podium held by Android, iOS and Windows.

Personal preferences aside, the fact is that when a company decides to go mobile it has to consider which platforms should be the focus of its attention and, consequently, the focus for its investments.
From our experience, and being lucky enough to regularly meet customers with these type of projects in their hands or in their roadmap, it is our conviction that

most companies don't have a clear vision of what the challenges are when developing cross-platform apps.

We have found that if we look closely we can find bits and pieces of information scattered throughout the company regarding what is the understanding about mobility.
However, normally there's no corporate vision about it nor how to deal with it when it comes to creating something that can be used in any device.

The aim of this whitepaper is to shed some light among the biggest variables when considering the development of a mobile app.

We will focus on aspects like strategy, design, development and tests and how **Xamarin** tools address those aspects.

2 Does your company implement mobile strategy?

Having spent some years now asking this question we are always waiting for that perfect answer that makes us conclude that a company is facing mobile initiatives in a structured way.

The truth is that, with some honorable exceptions, what we usually find are companies in which mobile *"simply happened"*

Sometimes it start as a marketing initiative where an app is just part of a marketing campaign.
Maybe it is a mobile dashboard that the IT Department has to implement due to a Board request.
Or even a project that one of the Departments implements as part of their individual strategy.

The definition of a mobile strategy is not going to be detailed here because it was the focus of a previous eBook, but we can conclude is that there are usually mobility silos spread across the company without any kind of corporate alignment or shared vision.

If we bear in mind the challenge of cross-platform development, there are a lot of advantages in defining correctly the path the company is to follow in terms of mobility.

This is the only way to ensure that an investment is turned into capital for other areas or departments in the company.

3 It's all about User Experience

Assuming strategy is dealt with, ***the time comes to DEVELOPING the app.***

No one questions the importance of having a correct set of requirements for a project to be successful.
What we have found is that sometimes companies are not considering the importance of user expectations when dealing with mobile apps versus other types of customer facing applications.

We all know that designing a mobile app is not just adapting what we already have to fit a smaller screen.

But there are a lot of things to be considered in order to ensure a great User Experience that can be materialized in an awesome User Interface.
And again, if it is already challenging to change the company design team's mindset to deal with one vendor's guidelines in one single platform, having to develop cross-platform apps takes it to an even higher level.

Google's Material design, Microsoft's Modern design and iOS' Human Interface design guidelines – all have to be taken in considered when designing the Experience.

Ideally, the requirements and design phase of the project should be the first phase of the project.
After the functional definition is set, each platform's app should be studied, architected and designed.
This is the phase where we should privilege the details regarding to user interaction – your users will be expecting it.
The app should "feel native" and even when it has an innovative design it has to respect the guidelines of the platforms.

3

There are two different approaches to create the design:

Create a similar experience across all devices

It's the approach that can be created quicker.
The main UX/UI is the same across devices and platforms, with some minor differences and of course respecting the guidelines.
For instance, Android and Windows devices have a physical back button, while iOS has it in the apps.
But, appart from these exceptions, everything is similar.
This approach has another advantage related to the fact that people don't need to learn how to use the app when they use different devices, because most of the interface is the same.

Create a native experience in each device

This is the approach that takes more time to do.
However, it is also the one that can guarantee that the app is completely integrated with the platform, its different design paradigms and characteristics.
It will represent a bigger investment but it will also create something really focused on the user, providing the functionalities according to each platform's strong points.

3

Some examples of scenarios that fit each of the approaches are:

Scenario 1

A company developing an app for a BYOD (Bring your own device) scenario that wants to disseminate the app and ensure its adoption with a small investment in platform specific training.

Scenario 2

A mobile banking app that wants to take advantage of each platform will benefit from taking the second approach.

Choosing one alternative or the other requires an evaluation and should not be decided lightly.
Of course the investment and Time to Market is important but we must not lose focus from the user.

After all, it's him that will dictate if the app is a success or a failure.

4 What about Development?

This is usually where conversation about cross-platform development starts.
It's understandable because development time represents the biggest slice of the investment on a mobile project, if you don't consider the development of backend services.
It's usually the most discussed topic and there's no single answer for it.

Developing an app destined to different platforms can be done following different approches, each with advantages and disadvantages.
Below you can find a list of possible approaches.

Mobile Web App

This was the main approach for mobile development years ago. It represents a controlled investment and is easily manageable, since it's basically a web app developed focusing smaller screens. It follows the usual web paradigms and it can be built using the same technologies.

PROS

- Investment is reduced when compared to other alternatives.
- Centralized maintenance for corrections and upgrades.
- Similar experience across devices.
- Development team can be the same as for the existing websites.

CONS

- It's normally connection-dependent.
- Hardware integration may not be possible.
- User Experience is not optimized by platform.

4

Native App

Developing a native app using each vendor's SDKs and tools will ensure a great User Experience and performance. Each app will take advantage of platform specific components and functionalities but has to be built separately.

PROS

- Great User Experience.
- Native Performance.
- Hardware integration is seamless using the SDK.

CONS

- Investment has a multiplying factor related to the number of platforms that are to be supported.
- Teams need to have specific skills for each platform.
- No shared codebase.
- Time to market.

4

Hybrid App

This approach consists in developing apps using a mix of web languages like HTML, CSS and JavaScript and then deploying them to different devices. Phonegap is one of the most famous frameworks for doing this. It's also possible to have a web app that feeds a native container – the container is nothing more than a native app that ensures things like hardware integration, for instance, with web views for the web app.

PROS

- Investment is reduced when compared to purely native apps.
- Easy to update, because usually all the business logic is on the web app.
- Code for existing websites can possibly be reused.
- Time to Market is lower when compared to native development.

CONS

- Most of the times, apps are still connection dependent.
- User Experience can be affected.
- Performance is not the same as for native apps.

These were usually the different scenarios a company should evaluate when developing cross-platform apps. Through the years, we have been faced with different choices based on different evaluation criteria from our customers. Having the necessary skills to follow any of the approaches we have always been looking for a solution that could help us and our customers to deal with this challenge of creating an app for different platforms – without having to compromise anything from the start.

5 And now let's test it!

There is a world of devices out there, and this increases the difficulty in developing an app. Different specs on different devices can lead to different behavior.

However, what usually happens is that when developing a project, most of the customers test the apps using a small set of devices. Maybe the partner developing the app has 5-10 devices and the customer another 5-10 – and this is the best scenario we have ever seen in all the projects we're involved!
This is nowhere near what theoretically would be needed to ensure the minimums in terms of quality.
According to a recent study by OpenSignal there are almost 19.000 distinct Android configurations out there – a mix of device models and OS versions!

So, what usually happens is that the apps are tested on a subset of devices hoping that most of the problems are solved before publishing it. Then the company becomes reactive solving bugs as they come – whether related to functionality, performance or design.

Based on various studies users are not that tolerant when it comes to ***FAULTY*** applications

There is a reason why this is the most usually found scenario. There's nothing rational in investing in dozens of devices, it is just too costly. So, even knowing that we are making the decision to risk having problems with some devices or OS versions, there is no other viable solution. We think that a very relevant progress is being made by Xamarin that we will detail later.

6 Maintaining it

After all is set and done and the apps are published, another phase starts.
Apps needs to be maintained and possibly evolved to meet new user requirements.

Maintaining different codebases, testing them again & certificating the apps can be a ***PAINFUL PROCESS***

If our experience in mobility taught us something is that after a mobile app has been finished there will be things to be changed and others to be added.
Starting the day after publishing it.
Developing cross-platform apps also has its toll here, especially depending on what approach was followed on the development of the apps.

Maintaining different codebases, testing them again and certificating the apps can be a painful process.
It has to be taken in consideration when the development approach is being decided because it can multiply the investment when considering a 1 to 3 years' timeframe.

App maintenance can be simplified if some type of monitoring is implemented. This will provide information about the app's behavior after it is published and even when users don't complain. It's an excellent idea to include this component in mobile projects to ensure that we have as much information as possible when something comes up.

7 Going forward with Cross-Platform Development

Xamarin's numbers are pretty impressive.

In 3 years they have more than **200 employees** 200 partners & 750.000 + developers

Developing an app that is cross-platform is a requirement that we keep hearing over and over again. No company wants to be present on only one of the platforms. However, most of them have some insights regarding the challenges involved in this. And some have even looked for a solution that could benefit them and solve the issues without requiring a compromise.

Xamarin is a company created in 2011 by Miguel de Icaza and Nat Friedman.
Their objective is to give developers powerful tools that are fun to use and that allow them to create mobile apps in a modern way. Their numbers are pretty impressive, especially if you consider that they have only been around for 3 years – as we speak they have passed the 200 employees mark, they have more than 200 partners and more than 750.000 developers being a part of their community. Since the founders have been working for that last decade in things that allow companies to use Microsoft technology in non-Microsoft environments (like Mono) their track record is impressive. This led us to believe that there could be something very real there.
The fact is that we have projects being built as we speak and it is living up to the expectations – both in terms of development and more recently in terms of mobile testing.

7

We work with ***Xamarin Platform*** to develop in C# apps that target *Windows iOS* & *Android* devices

The value proposition consists in creating a shared code base that has all the business logic and that can be used by the different apps, and then mapping functionalities in the apps' interface.

More recently Xamarin released something called **Xamarin.Forms** that also allows sharing some of the interface's components among the different operating systems.

They have also announced in the last weeks that companies like Telerik, Syncfusion and Infragistics already support or will support until the end of the year **Xamarin.Forms**, which gives developers a cross-platform version of more than 140 controls including graphs, gauges, and grids.

7

Each time we need to develop a cross-platform app we are faced with two possibilities:
the customer already has apps or he doesn't
(it can also happen that those that exist are not relevant).

When applicable we evaluate how much of the code can be ported to other platforms
– we use **Xamarin's .Net Mobility Scanner**, a tool that analyzes DLL's or EXE files and gives a full report, method by method, of what's used and where.

In the end we will get a percentage for the code's portability and we will do our estimate based on that.

We have had cases where we had 94% portability of a C# codebase for iOS and Android!

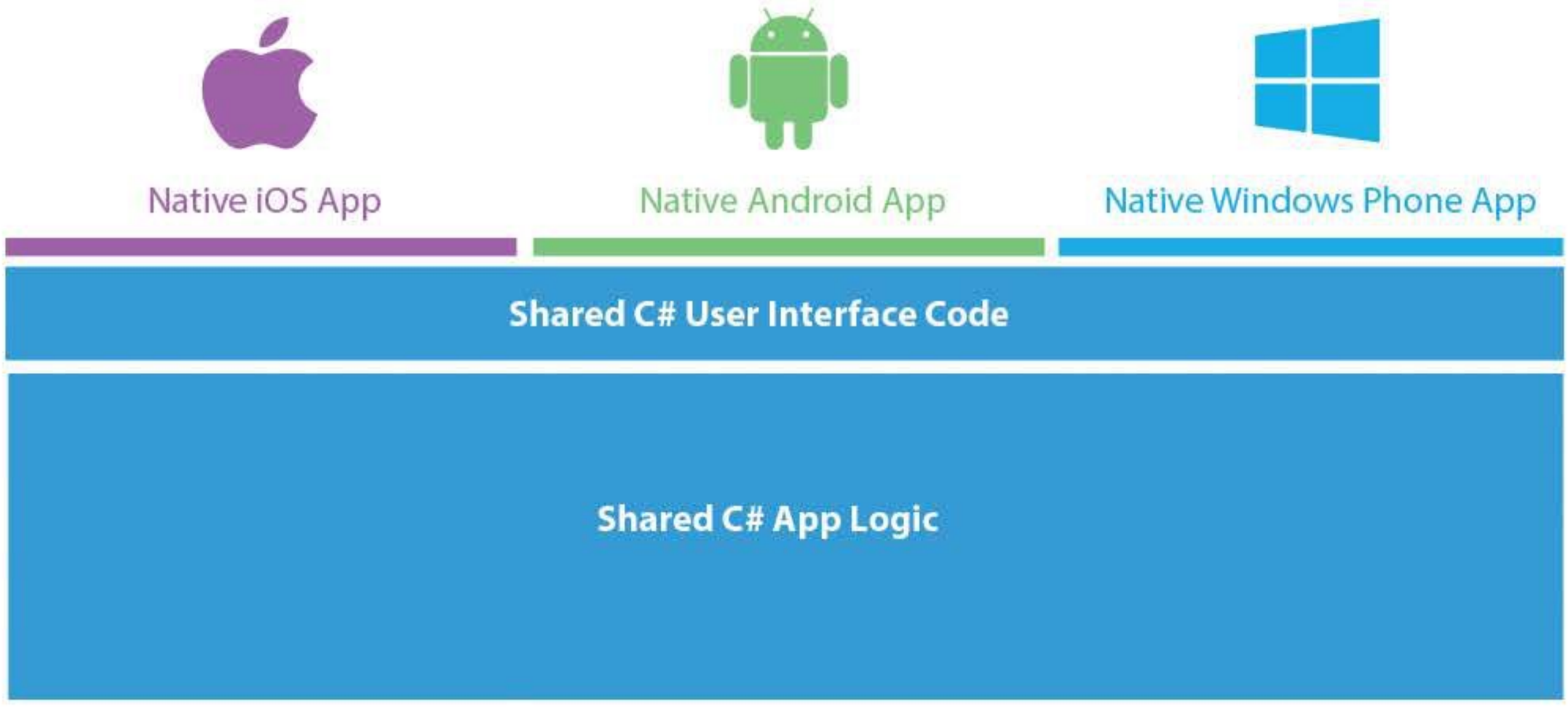

Image courtesy of Xamarin Inc.

7

When we need to do it our background in iOS & Android native development is a huge ***competitive advantage***

After this phase, there are some customers where a Proof-of-Concept is made in order to test specific functionalities, but most of the times we plan the development based on the information we gather from work sessions.

Our estimates are done separately for core business logic and for interface development – it should be straightforward at this time that we do this division because it is what makes sense in a Xamarin approach.

Our customers work with us on these estimates. They are always aware of what our assumptions are and this ensures an alignment between what we propose and what they want to see implemented.

When the project goes forward there are several things that need to be taken in consideration.

One of the most important is the app's architecture. We have to ensure that we take advantage of the core business logic to implement as much as possible, so that the development of platform specific code is minimized.

7

We have started using ***Test Cloud***

Xamarin's Device Cloud

Regarding testing we have started using **Test Cloud – Xamarin's Device Cloud** that allows to write tests in C# and then execute them against hundreds of devices.

First, we upload the app and the tests.

After the test run finishes we have access to screenshots of the different steps on every device we chose, and if there was an error in any of them we can check the console log and stack trace.

Pretty powerful stuff!

We are creating a specific offer for Test Cloud, because it focus one of the biggest pains in mobile development: Testing.

Xamarin also announced in Evolve 2014, their developers' conference, a preview of something called **Xamarin Insights.**

This will allow the implementation of app monitoring strategies focused on cross-platform apps, simplifying the whole process.
Even though this is a very recent solution, I have no doubt it will be a part of the offer we are building with Xamarin's products.

7

In sum, **our experience with Xamarin has been VERY GOOD**,
technologically and in terms of partnership relation.

We have been developing rich and complex apps using the technology
and it has been living up to the expectations.

Please feel free to reach out if you have any questions
or if you have a scenario where we could work together using **Xamarin**!

To find out more visit us at
www.xpand-it.com

xpandit

LISBON ADDRESS

PARQUE DAS NAÇÕES
EDIFÍCIO MAR DO ORIENTE
ALAMEDA DOS OCEANOS
LOTE 1.07.1Y - FRAÇÃO 2.3
1990-203 LISBOA

VIANA DO CASTELO ADDRESS

RUA DE FORNELOS 77
4900-709 VIANA DO CASTELO

EMAIL: **marketing@xpand-it.com**
PHONE: **+351 21 896 71 50**
FAX: **+351 21 896 71 51**
SKYPE: **www.xpand-it.com**

MEET > CODE > BUILD > TEST > DOCUMENT > REVIEW > RETROSPECT