**Presented by: Arul Kumaran | Author of Restler API Server Framework, CTO of @Luracast | @\_Arul**

Xamarin is an enterprise-grade, multi-platform development platform, with increased traction, off late.

Single code base, publish interfaces to all applications.

In Xamarin, code can be shared across platforms; provides full API coverage.

File size of an application increases due to SDKs of all platforms.

iOS does not allow JIT compilation (no RAW code running allowed). Android, however allows the same.

All iOS C# libraries are readily available to Xamarin toolkit.

Development tools available for Mac: Xamarin studio | Visual Studio with Xamarin plugin. While present Visual Studio is at Alpha versioning, it will soon be replacing Xamarin studio as an enhanced suite. For now, Xamarin studio is mature and stable and recommended for use.

Hackintosh can run on a Windows PC hardware; specific set of branded hardware (HP, Dell, etc.) can run the code, as the underlying processing platform is Intel for both Mac and Windows PCs.

Mac PC is not needed for development but for compilation. Need SSH and client software, run the simulator on the Mac to provide iPhone experience.

Also, Windows can be run as a native OS in Mac PC hardware, but could be terribly slow.

Xamarin brings the advantage of shared code with independent UI layer for the three common mobile platforms.

Xamarin forms: Develop once, with a common library. While compilation, compile along the independent devices’ platform specific native counterparts to have the UI compatible. By this, most of the code will be shared, with a bit of performance snag because of the shared and independent codes’ presence. However, the code hence developed, is not reusable, unlike in a native development platform.

Exposing and consuming a web service (especially a server backend) for a mobile application front end (client software), for the former is a heavy XML chunk and as a general practice the data layer should be thin/shrunk as much as possible. This challenge is especially possible with multi platform applications, such as an ASP.NETbackend and iOS front end developed using Swift.

Native development is more valuable than unified platform development, which is a Xamarin form. The former is always valuable and helps debug problems effectively. Learning both helps.

**Learning Xamarin**

Xamarin university offers $999, a very good course. Alternatively, there is a 30 day trial which can be signed up for. Xamarin certified developer has high-demand.

Evolve sessions contain videos that can help learning Xamarin.

BRAX provides videos on specific problem statements instead of lengthy videos covering all aspects of a specific technology domain.

Xamarin alliance provides a set of challenges to learn building applications using Xamarin forms.

**Learning C#**

For those interested to learn C#, Microsoft Virtual Academy offers the online video course, titled “Programming in C# Jump Start”, accessible at the URI https://mva.microsoft.com/en-US/training-courses/programming-in-c-jump-start-14254?l=j0iuozSfB\_6900115888

Monotouch dialog ready to use libraries, available for ease of development.

Xamarin workbooks: REPL, Read - Evolve - Print - Look, a strategy to lookup for a specific problem in a code easily, instead of compiling the entire code and wading through pages to lookup to the problem code. Also, a simulator is available to notice UI specific problems using the REPL tool for Xamarin.

Xamarin can work as a shared platform development tool as well as a native development tool.

Swift language is not Objective C as the latter is a more difficult to leverage for coding. In comparison, C# is a much more cleaner and effective language.

Xamarin has many sample applications, which can be leveraged, customised and repurposed for our purpose.

Difference between simulator and emulator: iOS native development platform is called simulator but Android one is called the emulator. Simulator, because the processor architecture is different hence requiring recompilation for iOS processor architecture, before porting to the actual iPhone device. This recompilation effort has to be paid for. Simulation does not guarantee performance, some features dysfunctional, uses computer memory and seem very fast but slow down when run from the actual IPhone device. Nevertheless, simulator is a quick way to test the code. Do not run everything on the UI thread, for example, a server call; instead running on a separate thread to avoid wait times of the main program.

Emulators solve the above issues, but code compilation is slower. Also, use a single emulator and not too many, for that could cause compilation issues as well. Genie motion is the best simulator for Xamarin development. Android emulators based compilation could fail due to fragmentation issues; hence test with more possible test cases, especially on the device itself.

Xamarin test cloud offers a paid service, to enable volume testing before releasing the application code for user experience.

Syncfusion offers plenty ebooks related to development, offering brief but complete coverage of the discussed concepts. “Xamarin forms succinctly” is a free ebook available from Syncfusion

Evolution of Xamarin: Originally planned as a bridge project to port .NET code to Linux OS platform, known as “Mono”, then evolved into Ximian. After Ximian’s initial failure to achieve the purpose, it was shelved. It was repurposed again with a new conception, named as “Xamarin” and present today, with multiple enhancements.

Xamarin studio installation on a Mac is relatively easier compared to that on a Windows PC, for the latter needs a high-speed internet connectivity. Offline installer shall come handy.

Programming with a proper framework consumes time, initially, but helps . When a framework is added, it of course adds another layer to the programming paradigm and hence causes performance loss; Nevertheless, if the value outweighs the performance loss, then it is worth considering. Framework helps structure an application, enabling easier collaboration, including team development and bug tracing - fixing. MVC, evolved into MVVM, is a good framework. MVVM Lite can be leveraged for effective development.

Version control: GitHub, Bitbucket are the best, but others such as TFS, SVN, can also be used. TFS is free for 5 developers. Visual Studio online can be leveraged, from a Mac system.

splitviewcontroller works with navigationcontroller to help with data transfer/projection between the Master and Details pages, when a Master-Details App template choice is made for development within Xamarin studio. Storyboard is an XML file that describes a UI based code flow, in this case, between the Master and Details pages, with the former producing “List” view and the latter “Details” view. Tools from the Toolbox shall be dragged onto the storyboard alongside springs, to lock them in a certain position, appropriate to the various devices on which it shall be displayed. Enter the device in debug mode, to avoid/workaround default navigation, processing, and security controls, which may otherwise work differently or based on a specific trigger.

One can create custom search controls if the native is ineffective for user experience.

Applications can be executed on a Mac using “Command+Enter” key combination.

Testing the various theories that we have conceived, including that of the multi-dimensional renderings of overlapping objects in a form, is possible also.

There is an aspect of programming known as “Markdown”, which is an attempt to reduce overheads in code pages, making them to be better readable lest unnecessary tags. Xamarin workbooks internally leverage markdown, which shall help understanding and tweaking code in an easier manner, collaboratively. More information at:

<https://en.wikipedia.org/wiki/Markdown>

<https://www.codeproject.com/KB/android/>

<http://web4.codeproject.com/Articles/799446/Using-Markdown-for-Effective-Logging?msg=4868294#xx4868294xx>

We have another community pertaining to or involving Azure platforms and services, named “Microsoft Azure User Group - Chennai”, evangelizing effective and efficient Azure usability for development and operational purposes. Interested professionals may join us at <https://www.meetup.com/Chennai-Microsoft-Azure-User-Group/>.