**Cross Platform Mobile Development**

**Abstract**

The purpose of this Literature Review is to consider the existing research, approaches, techniques and methodologies available for the development of mobile applications that have been written using a single framework, yet are able to run on platforms with different hardware and operating systems. These types of apps are known as cross platform.

Some of the more well known frameworks for cross platform development will be examined in detail. Relevant scientific literature to exploring and solving the issue that has arisen out of having many different consumer hardware and software devices and implementations while trying to obtain a single, unified experience across these different devices.

Also, this paper will exemplify the selection and criteria necessary for updates to the app to take place.

*I certify that all material in this dissertation which is not my own work has been identified*.

Matthew Shaw

**Introduction**

The ubiquity of smartphones and tablets in daily life has lead to a vast array of hardware, and a smaller number of Operating Systems. As businesses and developers attempt to make their products readily available to as wide as an audience as possible, the different strategies for development can become vitally important in terms of time management, efficient use of resources and user experience. [13-19]

The two main platforms [1] which dominate the majority of smartphones and tablets are iOS and Android, and each platform has a different legacy of software libraries which have a major influence on how software can be run and developed. The main issue faced by developers and businesses is how best to develop for different platforms, while providing a consistent experience both technically and in terms of User Experience. [9]

Cross platform development is a term used to describe a single programing framework or language that can be run on different operating systems. This is in contrast to native development which requires that the code must be written in a language specific to the operating system. [1-12]

A well known example of cross-platform compatibility (or lack thereof) amongst computer systems can be found in video games. Video games are sometimes found on one console system and not another, PC games are often designed to be run on specific architectures and occasionally some game developers will decide to translate or port the game over to other systems or have separate teams working on individual games.

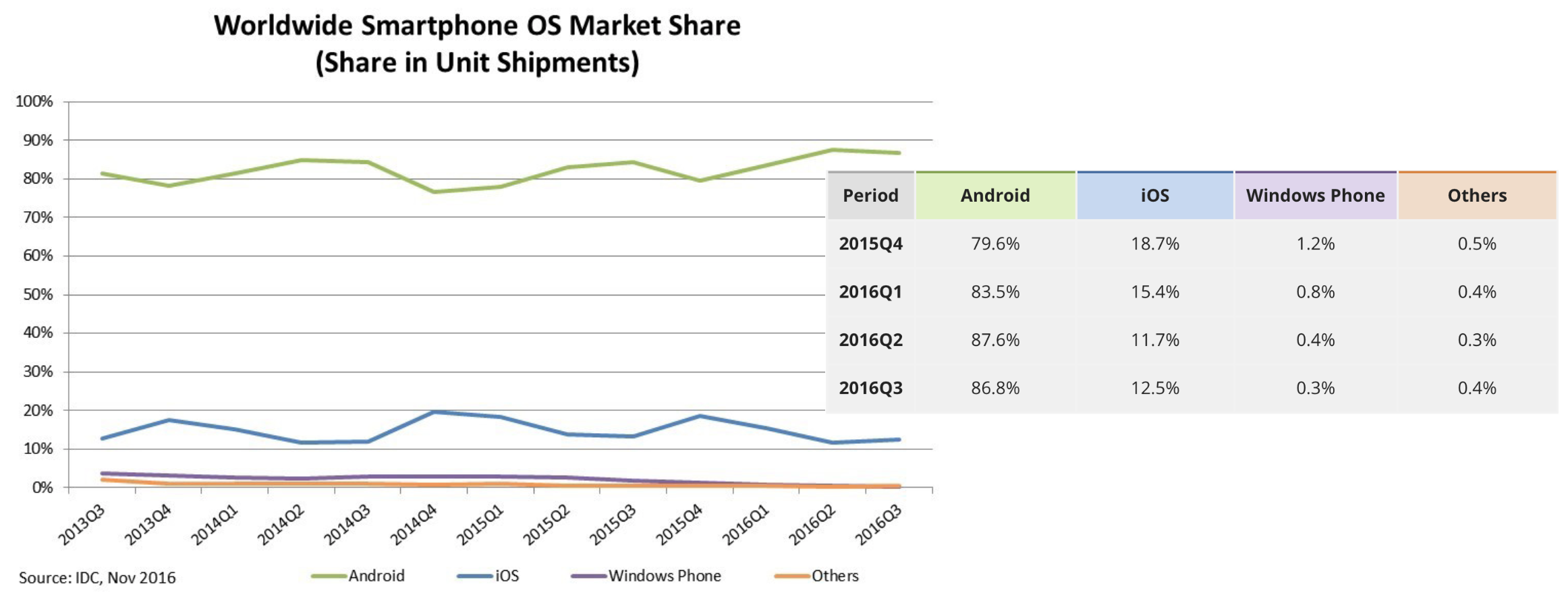


fig 1. showing mobiles os breakdown [1]

The aim of this literature review is to examine and explain the different approaches to developing for these platforms from a technical viewpoint. This will be done by comparing and contrasting some of the more popular cross platform systems in place, considering the pros and cons of cross platform development and even the different ways programs are created in a cross platform context.

To aid with the specification of the program to be created, this paper will briefly explore some of the popular development methodologies found in Software Engineering.

**Literature Review**

The following section will look at some of the prominent literature available on the subject of cross platform development as well as the different software packages available. This section of the paper will also include a comparison of the pros and cons of cross platform design with native platform development before further examining some of the existing literature on the matter in more detail.

**Cross Platform or Native Development?**

One advantage of Cross Platform development is that it can be simpler to have one code base. This is because developers can make changes to the program without having to do the same work more than once. This saves time, money and effort for the development teams.

Conversely, having a single code base could be considered a drawback, as developers are not able to take advantage of the individual benefits to be found on platform such as different the hardware component and software libraries. [13,16,19]

Another advantage of cross platform development is a consideration for the developers that are creating/improving the functionality of the app. A common industry practise is to expect or hire developers who are able to program effectively in a number of programming languages.

As most developers currently working in industry are able to code in a number of programming languages [20], Cross Platform frameworks can leverage developers understanding of the inherent differences of each platform in a way that is not restricted to a single operating system. The benefit of this is that the developers already have a familiarity with each platform and can optimise the code accordingly to run effectively using the different software and hardware libraries found in each system.

Having developers who are primarily skilled in Native mobile development languages not working in with the platforms they are skilled in should be considered a drawback as this leads to a loss of time learning a new language.

An advantage of having Native development over cross platform is that the separate code bases can be independently maintained. This should be considered a benefit of Native development as when one platforms Operating System is updated, changes can be made without an effect on the other.

**Analysis of Scientific Literature**

Comparison of Top Four Cross Platform App Development Frameworks [3]

This body of work has been considered for this Literature Review as it useful for showing where and when some of development technologies would be useful for a given project or lacking. This paper details a brief explanation of popular frameworks as well as tables clearly showing the advantages and disadvantages of each. This gives a good starting point to delving further into the specifics of how each platform could best be utilised as well as the unique strengths and weaknesses of each.

As this is an article from a website that specialises in providing technical assistance for development, there is a sense of interaction and directness as questions pertaining to the article can be discussed with the author in near real time. This source is useful as either a starting point to look into cross platform development, and as a reference once more background reading has been completed to quickly check ones understanding of the subject.

Taxonomy of Cross-Platform Mobile Applications Development Approaches [10]

This paper covers in great detail many aspects of cross platform development in extensive detail. This is a well put together paper that examines many of the challenges and solutions to creating code that is able to be compiled and run for different mobile systems. There are many useful charts and several references that make this paper an invaluable tool for exploring the work of Scientists, researches and programmers who have tackled this problem before. Interestingly yet somewhat outside the scope of this Literature Review and Project Specification, this paper looks at several different techniques of cross platform development besides software packages such as cloud based, merged and component based cross platform development.

This paper is an excellent reference as it shows how high the standard of quality can be for an academic paper of this type. The authors have even taken the time to compare different papers in sufficient depth so that other researches can glean insight into how to discern other good literature on the topic.

Comparison of Cross-Platform Mobile Development Tools [2]

While this paper does not extensively cover as much ground as the one previously described, one of the most useful aspects of this paper is how it considers the advantages of cross platform development and the specific challenges related to it. Some of the technologies specified in this paper have fallen out of favour since it was published, however the structure of the paper is clear and useful to other researches for finding specific information, be it architecture or compatibility with different API’s.

Xamarin Mobile Application Development [12]

This resource will be an invaluable asset throughout the remainder of this project. It contains a wealth of information related to utilising Xamarin, the related technologies and cross platform architecture. It focuses on the use of Xamarin.Forms and C# and clearly explains many useful generic and design patterns that can be used specifically for iOS and Android. Detail drills down to the class level and comparisons between the systems are explained in a way that allows developers and researches to get to work as quickly as possible on the task at hand.

**Specification**

This section is for the consideration of the technical and non technical functions to be delivered. The functional specification lists the requested updates to the existing system in terms of the app’s functionality. The non functional deliverables include less technical and more holistic features to be delivered. Comparison and analysis of the different technologies available to develop cross platform applications will also be placed here.

**Analysis of Software Tools and Packages**

Xamarin

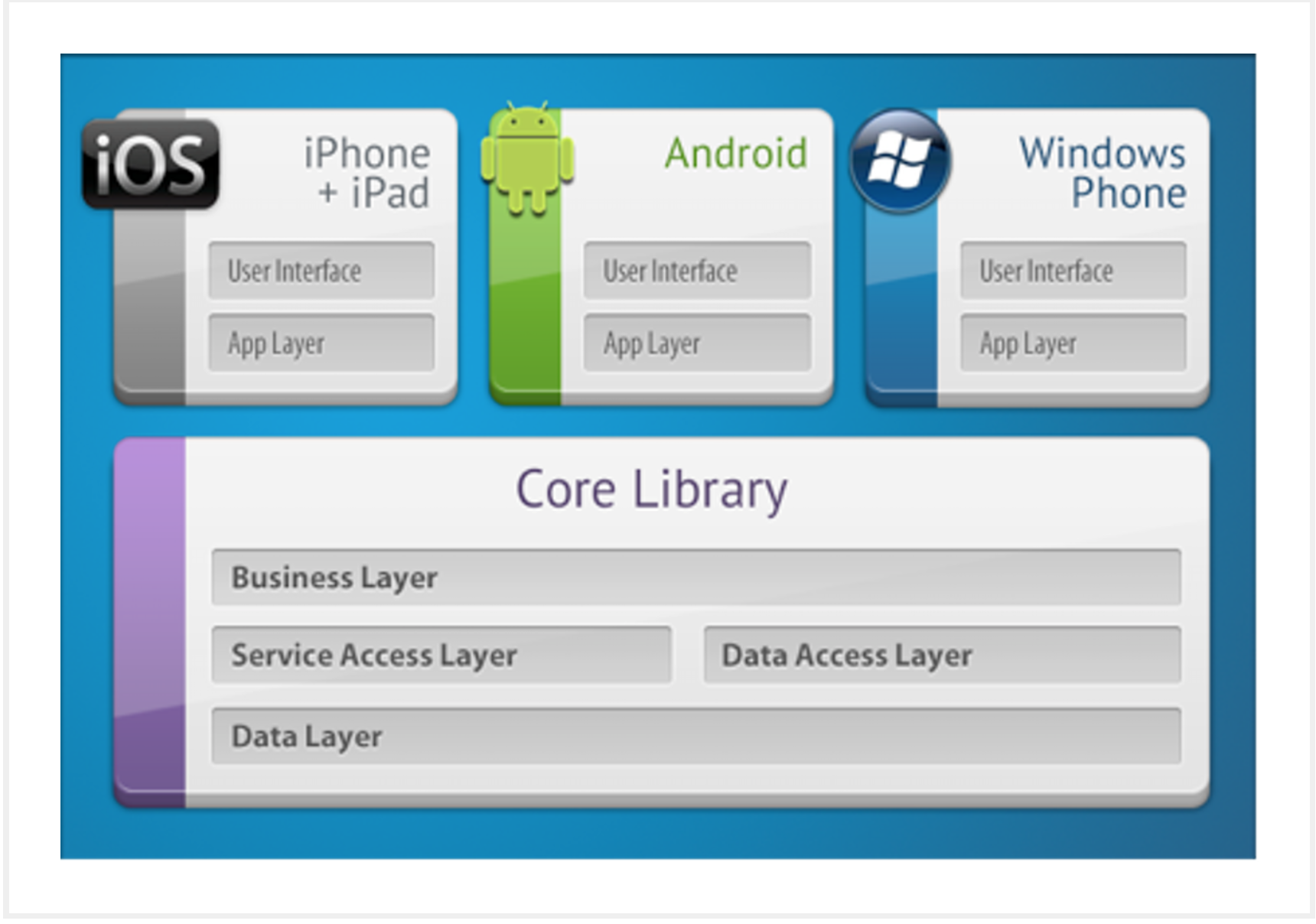


fig. 2 Xamarin Framework overivew[3]

Since 2011, Xamarin has been a widely used tool for communicating between the Application Protocol Interfaces or API’s for three of the most widely used mobile operating systems: iOS, Android and Windows. Native libraries can be utilised which will take advantage of pre-written code that handle, for example, the opening of a camera, running the GPS module or even system functions such as how an application is paused in the background when a phone call is received.

Xamarin makes use of wrapping native libraries with .NET and C# which allow third party libraries for Android and iOS to operate on the platform. Xamarin gives cross platform programs the operability of native applications and User Interfaces or UI’s can be built which are aesthetically pleasing on both Android and iOS.[citation, apps with Xamarin + four cross] With consideration to User Interfaces, there are over forty different UI controls which are not all supported on android. This may require Native UI development which is an issue for organisations attempting to have a constant look and feel for users across different devices.

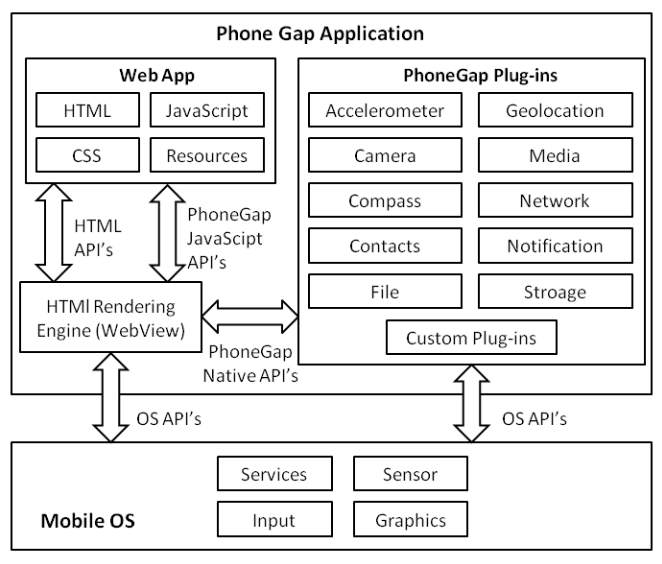
As the .Net framework is not typically a development language commonly used by mobile developers, this gap in learning must be offset by developers by native platform understanding in either Objective-C or Java. Apps built with Xamarin can be much larger than native apps or other platform independent options which can be very slow to download or emulated. [3,6]

Appcelerator Titanium

Approximately 10% of all smartphone apps are built using Titanium. This is partly due with the fact that it is Open-Source and as such benefits from a large community of experienced developers. Titanium is based on the JavaScript language which is known for its portability[citation] and combines the use of platform independent languages such as HTML5, CSS, JQuery and Ajax. These languages are the basis of Web Development and as such, developers looking to produce websites into mobile apps will find Titanium a powerful toll at their disposal as the API’s allow them to integrate the hardware components found in mobile devices.

A large benefit of using Titanium for cross platform development is thanks to the community of developers and business acquisitions which include the implementation of cloud services. Reasons why Titanium is not more often used include that it is open source; drawbacks of having a system where the source code is available to everyone in the public sphere include security issues with high profile projects as organisations may wish to develop more proprietary or closed source systems. There have been documented cases of issues with specific SDK versions and incompatibility issues and problems with in app purchases. [2-8]

Phonegap

A popular choice from the opinion of the developers who create cross platform applications as this framework has a less steep learning curve than some other options which allow the Software Engineers to focus on making apps rather than trying to work with a specific implementation. Similar to Titanium, Phonegap makes use of some of the most popular Web Development languages: JavaScript, HTML5 and CSS. Apps developed using Phonegap can take advantage of a simple ‘plug-in’ style component architecture which essentially extends the native API’s. 

Complete schema of PhoneGap architecture and interfacing among components

There is no licencing fee as Phonegap has ben open sourced. However, open source software is only as useful as the community which supports it and there are still many useful open source API’s which are not supported by phonegap. This issue becomes more problematic as without proper control to interface between the devices hardware and the app, updates to either can render the app inoperable. There are also security issues in addition to having the source code publically available in that there is little to no cache or memory management which has worrying consequences for applications which handle financial transactions or even personal data such as location.

From weighing up the benefits and drawbacks from each of these technologies, the most appropriate course of action will be to continue to develop the project using the Xamarin framework. This is because of the strong compatibility with the existing code base owned by Mapyx as mentioned in the Project Statement and plan, and mainly due to the use of C# and Xamarin’s focus on mobile apps where Phonegap and Appcelerator Titanium are more concerned with technologies that are Internet based, instead of focusing on mobile devices.[2,5]

Functional deliverables

• Add ability to centre map on location when click on device in trackers

• Files removed through the sync process are not removed from device

• iOS – Routes view does not scroll

• Activation / Deactivation using the same device name as unique id is not consistent

• Ability to record POI with note and photo?

• Add beacon ability to share location to server when tracking self

These are the desired improvements to the existing software app to be delivered to Mapyx in partnership with the University of Exeter and further detailed in the application proposal.[9]

Non-functional deliverables

The main non functional criteria for this project is to provide a seamless experience across both iOS devices and different Android mobile devices. This will be achieved by ensuring a consistent look and feel across the devices with particular attention paid to ensuring the User Experience is as similar as possible across the different handsets available. Different methodologies for completing this project have been considered and upon reflection the Agile methodology for software development is best suited to complete this project.

Additionally, a consideration for the data security and integrity will be performed as any improvements to the existing system would ideally be made before the replacement of the Data Protection Act. As mentioned in the Ethical Considerations sections of the proposal it is important that data is being collected in a way that balances rights to individual privacy as well as the needs of the business providing the service.[20-22]

**Evaluation and Criteria**

The final product will be assessed based on how well the functional and non functional deliverables have been completed. This will be specified through regular feedback and communication with Mapyx as to how they feel the product has been improved and updated throughout the project. Testing will be conducted by a focus group of users who will be asked to rate differences in the user experience, usability and functionality of versions of the app before and after the improvements have been carried out.

**Summary and Conclusion**

To summarise, this project will aim to improve the existing app owned by Mapyx through the use of Xamarin and will be produced using the Agile methodology. As the Agile methodology is particularly adept at considering new additions to the workload, the possibility exists for extra updates to be carried out depending on client and user feedback. towards completion of the project and depending on how satisfied all parties are with the state of the app, there is an opportunity to further develop some additional features a new, stand alone product.

References

[1] Y. Wu, Mobile and Ubiquitous Computing, University of Exeter, 2017

[2] M. Palmieri, Comparison of Cross-Platform Mobile Development Tools, 16th International Conference on Intelligence in Next Generation Networks, 2012

[3] A. Mahajan, Comparison of four cross platform app development frameworks, 2016

[4] J. Perchata, Component Based Framework to Create Mobile Cross-platform Applications,The 3rd International Symposium on Frontiers in Ambient and Mobile Systems, 2013

[5] G. Hartmann, Cross-platform mobile development, Office of Naval Research Global, 2011

[6] V Bezuglaya, Comparing five popular frameworks for mobile development in 2017, Graph Digital Limited, 2017

[7] A. Holzinger, Making Apps Useable on Multiple Different Mobile Platforms: On Interoperability for Business Application Development on Smartphones, Medical University Graz, 2012

[8] anon, Mobile app development tools a detailed comparison, build fire, 2017

[9] S. Jeffs, MX Map Mobile Consolidation Project, Mapyx, 2017

[10] W. El-Kassas, Taxonomy of Cross-Platform Mobile Applications Development Approaches, Ain Shams University, 2015

[11] J. Mello, 10 cross-platform mobile development tools for enterprises, TechBeacon, 2017

[12] D. Hermes, Xamarin Mobile Application Development, Apres

[13] P Beynon-Davies, Rapid application development (RAD): an empirical review, European Journal of Information Systems, 1999

[14] D. Walker,Software Development Lifecycle, University of Exeter, 2016

[15] A. Spiral Model of Software Development and Enhancement, TRW Defense Systems Group

[16] F. Giunchiglia, The Tropos Software Development Methodology: Processes, Models and Diagrams

[17] P. Bresciani, Tropos: An Agent-Oriented Software Development Methodology,2004

[18] J. Highsmith, Agile software development: the business of innovation, Computer Volume: 34, Issue: 9, Sep 2001

[19] A. Cockburn, Agile software development, the people factor, Computer Volume: 34, Issue: 11, Nov 2001

[20] C Petzold, Creating Mobile Apps with Xamarin, Forms Preview Edition 2, 2015

[21] D. Gollmann, Computer Secu- rity (Third Edition), John Wi- ley and Sons, 2011

[22] L. Brown, Computer Security: Principles and Practice, Pearson, 2014