

April 2014, HAPPIEST MINDS TECHNOLOGIES

# An Analysis of Mobile Application Development Approaches

Author

Umesh Narayan Gondhali



SHARING. MINDFUL. INTEGRITY. LEARNING. EXCELLENCE. SOCIAL RESPONSIBILITY.

## Copyright Information

This document is exclusive property of Happiest Minds Technologies Ltd. It is intended for limited circulation.

## Table of Contents

1. Abstract
2. Introduction
3. Evaluation Criteria
4. Analytical Overview of Cross-Platforms
5. Selection Criteria
6. Happiest Minds Role in Leveraging Cross-Platform Development

## Abstract

With the surge of mobile devices in the market and its wide-spreading applicability in almost all sectors of the world, it has become very important to assess the development approaches in building information systems on the mobile. Mobile devices being backed by various operating systems and development platforms have posed a challenge in building applications in terms of various aspects like development cost, development technology, skilled people, learning curve of the developers etc. In order to analyse such various modes/practices of development, we, in Happiest Minds, have taken up a systematic study of Xamarin, Phone-gap and Appcelerator platforms based on well-defined criteria as enlisted in this paper.

## Introduction

Mobile devices, i.e. hand held devices in general, are gaining ever increasing popularity due to its hardware and software capabilities like sensors, camera, touchscreen, information communication windows/gateways. As per Gartner report, 2016 will witness the widespread applicability of mobile/handheld devices in majority spheres of businesses, industries and human life. Though there are various operating system players, it is noteworthy to consider Android, iOS and Windows as the most established and steadily evolving operating systems in mobile devices space. As all the mobile OS platforms differ in their technology, programming language, implementation and approach; the possession of different technology skillset, has become very challenging for the developer community and enterprises in building applications on mobile to reach wider & diverse audience of target users.

Cross-platform development approaches are seen as a solution to the above mentioned challenges by conferring the developer to use unified cross-platform technology that is compatible across the mobile operating systems. These cross-platform development methodologies avoid repetition in app development and foster the existing skillset of developers and greatly influence to reduce the learning curve.

This paper outlines our studies, analysis and critical examination of cross-platform development methodologies/platforms/tools based on HTML5, JavaScript, CSS3 and Xamarin. As these differ in their architecture, implementation approach and vivid technology skills, it expects the need for establishing criteria to assess the cross-platform technologies like Xamarin, Appcelerator and Phone-Gap, which we have tried to outline in this paper.

The paper first introduces the criteria on which the platforms are assessed then each cross-technology platform is examined with respect to the established criteria, and then the selection criteria is discussed, followed by Happiest Minds role in adopting & catering to these disruptive technologies to finally conclude the paper.

## Evaluation criteria

There can be various factors which qualify to be a benchmark in evaluating cross-platform development approaches but we have limited to some of the potential criteria listed below:

- Look and feel
- Mobile OS platform support
- Leveraging device capabilities
- Learning curve
- Development accelerators
- Impact on regular development time
- Performance
- Packaging & distribution
- Long term feasibility & developer community
- Platform cost, licencing model and maintenance

## Analytical overview of the cross-platforms

### Xamarin

Xamarin is a studio built to develop cross-platform native mobile applications. It consists of a suite of products viz Xamarin.iOS, Xamarin.Android, Xamarin.Mac and Xamarin.Studio that caters to a cross-platform mobile app with separate implementation for respective native UI with a common sharable application code. Mobile application builds comes with packaged assemblies that act in invoking the native APIs of a mobile OS via reflection.

- **Look and feel**  
Xamarin leverages native look and feel which is the main differentiator in the league of cross-platform competitors. It has dedicated flavours of IDEs and interfaces in iOS, Android, Windows and Mac which communicates to the respective native UI APIs via reflection.
- **Mobile OS platforms support**  
Xamarin supports 3 major operating systems: android, iOS and windows. In addition to these it also supports application to be ported on Mac.
- **Leveraging device capabilities**  
As Xamarin is a native cross-platform technology, it supports all the device capabilities of an operating system. If a new version of a particular mobile operating system is released, the respective API changes are incorporated by Xamarin and latest version is released for the Xamarin developers in order to incorporate the new features.
- **Learning curve**  
Xamarin uses C# as a programming language. So In order to develop a cross- platform native application, a developer will have significant learning curve, as he/she needs to learn the C# programming language and acquire some basic knowledge of native application development.

- **Development accelerators**

Xamarin comes with a components store. These components are either free/paid in nature & proves to be a great mean for acceleration of application development using these reusable components.

- **Impact on regular development time**

Primary objective of cross-platform application development is reduction in code and/or avoid repetition of app development in various native development mode. Majority of application logic code can be reused for the respective native platforms. However the GUI code will be platform specific, and cannot be reused.

- **Performance**

Applications built using Xamarin will demonstrate the same performance as that of native applications. However there will be some reflection overhead incurred as .net runtime assemblies are packaged with the application bundle. This overhead is quite negligible and doesn't pose any significant hindrance on the performance of the application.

- **Packaging & distribution**

Xamarin studio churns out the respective native builds which can be uploaded onto the native platform stores. Along with the application specific artefacts Xamarin bundles a light weight mono runtime into the build that interprets and invokes the platform specific APIs at the runtime of the application.

- **Long term feasibility & developer community**

Xamarin has an ever increasing huge developer base across the globe, presently it is close to half a million. It has around 100+ partners comprising of small, medium and large scale IT service providers. Xamarin proactively engages in conferences, rendering technical support and substantive learning material online along with varied SLA support.

- **Platform cost, licencing model and maintenance**

Xamarin studio or plugin to the Visual studio comes under paid licence and a free trial option. Licences are of annual subscription in nature which varies from \$299/- to \$1899/- per OS platform for a year. Trial period is for 30 days and applications built using trial period (free) licence will have a functional/deployment validity of 24 hours. The Xamarin developer studio licence needs to be renewed every year. The applications built using Xamarin studio will bear no license fees/costs.

## Phone-Gap

Phone-Gap is a free and open source framework that allows you to create mobile apps using standardized web APIs for the 7 major mobile operating systems.

- **Look and feel**

Look and feel of the applications will be non-native as the GUI is built from HTML and CSS and other JavaScript frameworks like Sencha, jQuery Mobile etc.

- **Mobile OS platforms support**

Phone-Gap supports all major OS players. It supports Android, iOS, Windows Mobile, Blackberry, Samsung Bada and web OS.

- **Leveraging device capabilities**

Phone-Gap supports majority of device capabilities like camera, sensor, accelerometer, GPS etc. It also supports OS platform specific hardware features like NFC for android.

- **Learning curve**

It is important to have a novice competency skillset in at least one mobile native OS programming. In addition to this, developer should have a profound skillset in HTML & JavaScript. Developers with web programming background will quickly establish command on Phone-Gap development.

- **Development accelerators**

There are various paid/open source JavaScript and/or HTML5 frameworks which will foster the development. Some of them are JQuery Mobile, Sencha etc.

- **Impact on regular development time**

Phone-Gap development leads to building mobile application that can be packaged into platform specific builds and deployed across 7 major mobile operating systems. This avoids the repetition of application development by many folds, bringing great reduction in development team size and various skillset of developers and by reducing the development cost.

- **Performance**

Performance of the hybrid application built using Phone-Gap will be comparably lower than the applications built natively. But with the latest releases in Phone-Gap software, the gap of difference is gradually shrinking, indicating a good sign and traction towards mobile hybrid applications development.

- **Packaging & distribution**

There are two ways by which a developer can churn out the specific builds for the different mobile platforms. First is using the eclipse plug-in in the development mode.

The second way is with the cloud based tool from Adobe called “Adobe Phone-Gap Build”. Using this method developers can upload their corresponding application artefacts and cloud to hold to handle the packaging of application for different platforms. Once they get the packages, they can host them in the respective OS stores.

- **Long term feasibility & developer community**

Phone-Gap being is an open source ever growing developer community, and its long term survival can be foreseen. Most of the service industry leaders are showing keen interest in Phone-Gap. It is majorly sponsored by Adobe and Nitobi. There are millions of applications already developed using Phone-Gap and a huge user base is using those applications.

- **Platform cost, licencing model and maintenance**

The Phone-Gap code has been contributed to the Apache Software Foundation (ASF) under the name ApacheCordova. It is a free and open source under the Apache License, Version 2.0

## Appcelerator

Appcelerator tries to address the new mobile reality of diverse set of devices by delivering native cross-platform apps at the speed of web, mobilizing any data source, and driving success with real-time analytics – all from an open, cloud-based platform:

- **Look and feel**

Titanium uses rich set of API libraries that are used to build GUI using native UI widget elements and also HTML5 UI elements. So developer has the degree of freedom to choose to build either native or web UI elements in building GUI.

- **Mobile OS platforms support**

Currently Appcelerator supports building applications for five major platforms that include Android, iOS, Windows, Blackberry and web hosted (HTML5).

- **Leveraging device capabilities**

The titanium library has rich set of APIs which can be used to access device specific features. These APIs can be classified into generic device access APIs which can be commonly used once and at run time they act as per the device & operating systems. These APIs have fair degree of control over using device capabilities. The second category of APIs are platform specific APIs which demonstrate more degree of control in maximising utilization of the device capabilities.

- **Learning curve**

The technology stack mainly requires deep knowledge in Java Scripting and some of the open source JavaScript frameworks like JQueryMobile. So a developer from web technology background will find it easy to learn and build applications using Titanium. It is also important to possess some rudimentary knowledge with respect to mobile operating systems in order to choose the platform specific APIs. Platform website provides rich set of learning material which will foster easy learning process.

- **Development accelerators**

Appcelerator has rich set of accelerators built and published in its market place by third party developers/organizations; they can be either paid or free of cost Platform also provides proven MVC framework library which eases the cost of development by reducing numbers of code lines.

- **Impact on regular development time**

Development time is greatly influenced on the nature of the Titanium APIs being used. The platform specific APIs usage requirement depends on the development time for platform specific implementation using Titanium APIs. If more generic APIs are being used then it greatly reduces the repetition of platform specific implementation and therefore fosters reduction in development cycle time.

- **Performance**

During the initial versions of Studio, applications with simple views used to demonstrate considerable performance but for the complex views, the user could notice a little hiccup in rendering the view and in views transition. But with the subsequent releases of platform, this latency has greatly shirked, and the gap of performance between an ideal native app and app built using titanium studio has become thinner.



- **Packaging & distribution**

Titanium studio facilitates packaging and bundling platform specific builds and the builds can be hosted in the OS specific stores.

- **Long term feasibility & developer community**

Currently there are more than half a million developers across 185 countries involved in building mobile applications using the Appcelerator platform and close to 200 million devices are running the applications built using Appcelerator platform. This has a rich set of partners and training materials.

- **Platform cost, licencing model and maintenance**

Appcelerator comes under two plans; one is free of cost for independent developers. This mode of license has limitation on the resource utilization on daily and monthly bases.

The other license mode is for enterprises. This comes in two varieties public cloud and virtual private cloud. Both require licence purchase.

## Selection criteria

When it comes to choosing cross-platform technologies for mobile application development, many critical factors contribute to the decision making process, such as

- Target audience (geography, language, etc.)
- Type of business (B2E, B2C)
- Feature list and its impact on devices (device hardware dependant/independent)
- Code partition (device and/or server/cloud code)
- Time to market etc.

So, by funnelling all such decisive factors into architectural mould, we can view the mobile applications in one or many of following perspectives;

- UI rich applications ( e.g. photo shop, gallery apps)
- Applications with intensive computation implementation on device (e.g. games)
- Information rendering applications (e.g. e-Book reader, sales dashboard apps)
- Applications with intensive hardware utilization (e.g. torch, map, compass apps)
- Applications productively operable in offline mode

Xamarin Mono touch is designed in such a way that code share or reusability of code happens at the application layer. This means GUI implementation will be repeated for platform specific UI implementation. So in order to reap maximum benefits from Xamarin it is advisable to build the mobile applications which require intensive computation (deep application logic) on the device. It is suitable for applications that can benefit from offline mode operations and also the applications that rely heavily on the hardware capabilities of the device.

Phone-Gap is more suited for application which acts as information renderer by pulling the information from the cloud, and also for the applications where native look and feel of the application is of least priority. Dashboard intensive applications in sales force sector or applications that monitor an enterprise KPI get an inherent technological benefit from HTML5 and some of the open source of JavaScript frameworks. By deciding on Phone-Gap for such applications

development life cycle is greatly reduced and hence favours go-to-market timeline of the application.

When we have to choose an intermediate path by giving significant consideration to look and feel, where UI is not a top priority, then the enterprise business applications that requires inherent analytics and features like MAM and MDM can be built using the Titanium development platform. A rich set of specific APIs of Titanium development platform provides close to native look experience.

## Happiest Minds Role in Leveraging Cross-Platform Development

### About Happiest Minds Technologies:

Happiest Minds, the Mindful IT Company, applies agile methodologies to enable digital transformation for enterprises and technology providers by delivering seamless customer experience, business efficiency and actionable insights. We leverage a spectrum of disruptive technologies such as: Big Data Analytics, AI & Cognitive Computing, Internet of Things, Cloud, Security, SDN-NFV, RPA, Blockchain, etc. Positioned as “Born Digital . Born Agile”, our capabilities spans across product engineering, digital business solutions, infrastructure management and security services. We deliver these services across industry sectors such as retail, consumer packaged goods, edutech, e-commerce, banking, insurance, hi-tech, engineering R&D, manufacturing, automotive and travel/transportation/hospitality.

Headquartered in Bangalore, India; Happiest Minds has operations in USA, UK, The Netherlands, Australia and Middle East.

**Business Contact:** [business@happiestminds.com](mailto:business@happiestminds.com)

**Media Contact:** [media@happiestminds.com](mailto:media@happiestminds.com)