ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 1, No.7 , January 2014
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

Developing mobile educational apps: development strategies, tools and business models

Serena Pastore

[1] **Astronomical Observatory of Padova, INAF**
Padova, 35122, ITALY
serena.pastore@oapd.inaf.it

Abstract

The mobile world is a growing and evolving market in all its aspects from hardware, networks, operating systems and applications. Mobile applications or apps are becoming the new frontier of software development, since actual digital users use mobile devices as an everyday object. Therefore also in an educational environment, apps could be adopted in order to take advantage of mobile devices diffusions. Developing an app should not be a decision linked to the trends, but must follow specific strategies in choosing target devices, mobile platforms, and distribution channels that in these contexts usually go through e-commerce sites called stores (e.g., Apple Store, Google Play, Windows Phone store). Furthermore the design of an educational mobile app requires a careful analysis on the methodologies to be adopted and the available tools especially if the aim is to build a successful and useful app. Very often there is a tradeoff between the costs associated with the development that requires a lot of technical and programming skills. The economic return is neither high nor guaranteed considering the business models offered by the major stores. The paper deals with the approaches to apps development analyzing methodologies, distribution channels and business models in order to define the key points and strategies to be adopted in an educational context.

Keywords: *Mobile apps development, cross-platforms mobile tools, native apps, e-commerce, hybrid apps, Sencha Touch, Apache Cordova*

1. Introduction

The mobile world is a growing and evolving market in every its facets. Mobile devices [1] intended as electronic devices created for specific functions (e.g., voice like cell phones and smartphones, reading books as e-book reader, and more general activity such as laptops or tablets) have undergone exponential growth in recent years, especially with the introduction in the market of mobile devices connected to the Internet network as smartphones and tablets. Even if such market includes the different categories of mobile devices, tablets and smartphones seem to be the top categories in accordance with the several market analysis companies [2]. Mobile actors involved in the mobile ecosystem (Fig. 1) includes devices manufacturers (e.g., Nokia, Samsung, RIM Blackberry), software platforms providers (e.g. Apple; Google, Microsoft), applications developers and mobile operators that in each country (e.g., in Italy Vodafone, TIM, 3G) provide Internet connectivity. A single mobile device is composed by specific hardware and software platform that deals with the peculiarities of these systems. Mobile devices should follow the user's mobility (therefore mobile devices are generally of reduced size and weight, such that they can be transported easily). Their hardware components manifest limited processing power and memory, little screen dimension, but a variable numbers of other hardware such as cameras, network adaptors (Wi-Fi, Bluetooth), sensors like GPS (Global Positioning systems) receivers, accelerometer, compass, proximity sensor, ambient light sensor (ALS), and gyros.

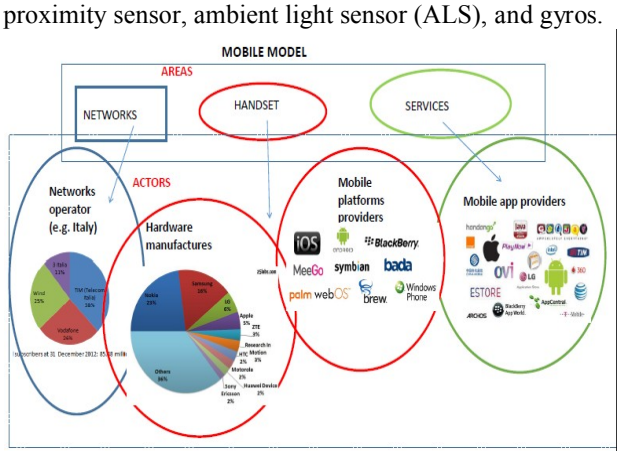

Fig. 1 Actors of the mobile ecosystem

Software applications on such devices or mobile apps as they are universally called, are the new frontier of software considering the mobile devices diffusion and should follow the rules of such ecosystem. Developing an app of every category, even in an educational environment, should be the result of a careful analysis of mobile platforms, tools and methodologies to be adopted. There

are a lot of hardware manufactures (e.g., Apple, Samsung, Nokia, LG, HTG, Asus) and this fragmentation also breads in mobile software platforms (e.g., iOS, Android, Symbian, and Windows Phone) that directly deal with hardware and help to take advantage of the device's performance. An app is usually targeted to few software platforms, since developing for all existing platforms is really expensive and requires high technical skills. The paper deals with the approaches to apps development focusing on methodologies, tools and distribution channels to analyze peculiarities and issues of the different methods. From methodologies point of view, apps are divided into native and web apps [3] meaning the use of different technologies. Moreover a compromise between the pro and cons of the two methods are the hybrid apps that use special frameworks to convert web apps into native apps. There are different aspects that should be taken into consideration when choosing a specific platform or methodology, and the paper would like to analyze them in order to figure out which is the best approach for educational apps. The first section analyzes apps' features focusing on types and the structure of their market. The second section distinguishes the choices in the development methodologies and tools, while the third section examines the tools needed for the development of each kind of app. It will finally give an evaluation of the various types of development taking into account also the distribution channels of the store normally used by users to search for an app.

2. Mobile apps features and the business models

App design (Fig. 2), starting from the concept idea and requirement analysis, mainly differs in the stages concerning the development and the final delivery.

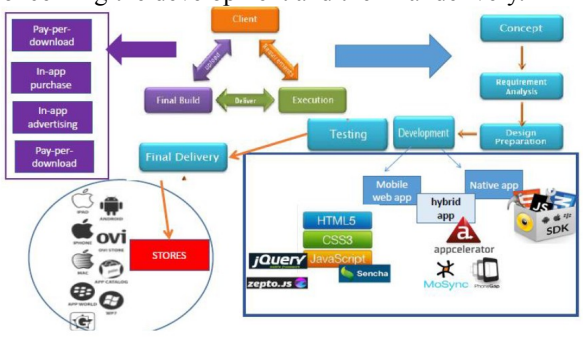

Fig. 2 Apps' development and deployment steps

As regards development, apps can be divided into native or web apps meaning the use of different programming technologies and thus different tools and frameworks. Native apps use the languages of the specific mobile platform and are therefore capable of interfacing completely with the hardware's device. But since they are dedicated to mobile devices equipped with specific operating system, they are targeted to a single platform. Web apps use web technologies (HTML5, JavaScript, CSS3 [4]) and are multi-platform, since they can be executed from any mobile device equipped with a web browser. However such app could not take advantage of all hardware capabilities, since interacts with the hardware by means of an intermediate software layer and specific Application Programming Interfaces (APIs). A compromise between the pro and cons of the two methods are the hybrid apps that use special frameworks mainly based on JavaScript technology to convert web apps into native apps [5].

Usually apps could be preinstalled in the device and thus imposed by the mobile software reference platform or could be installed by the user or, as in the case of web app, imply a request to an Internet server machine. Convince a user to download and use an app depends a lot on both its utility and its quality and how it is distributed.

As regards apps distribution, even if some app can be delivered through specific websites or e-mail, the main distribution channel is an e-commerce site called store [6] owned by major hardware and software mobile providers. Each store imposes specific policies for apps distribution and different business models. In this context, a developer should decide if developing for one or more mobile platforms, since this decision affects the choices on methodologies, technologies, tools and distribution channels.

2.1 Mobile devices and platforms

Each mobile device has different technical features in terms of available memory, computing power, screen resolution, operating system (OS) and the hardware/software platform that determine the quality and performance. Even if actually the market seems to be occupied by few dominant players (Apple and Samsung), there are many other manufacturers of hardware and software that are trying to earn their slice market. Many companies producing mainly software such as Microsoft and Google have made agreements with hardware manufacturers to provide their own systems. Other companies like Apple or RIM Blackberry provide both the hardware and the software. Each manufacturer tries to keep pace with new technologies, thereby spreading out ever newer models of the two main categories of devices (i.e., smartphones and tablets) trying to anticipate competitors to increase their market value.

Among the leading manufacturers of mobile hardware, Nokia has been at the top of the sale of mobile

phones for many years, offering devices running the SymbianOS operating system. But with the arrival of smartphones, Nokia has lagged behind its competitors Apple and Samsung. It currently offers mobile phones that still have the Symbian OS operating system, but also the smartphones Lumia series that for now, thanks to an agreement with the supplier of Microsoft software are equipped with the Microsoft operating system Windows Phone. Samsung has been for many years very successful in selling low-end phones and smartphones. Samsung uses as software platforms the BadaOS of his property (even if it will be probably replaced with the TizenOS [5], an open operating system developed in collaboration with Intel) in most of mobile devices. It uses the Android operating system in the smartphone Galaxy series which are those that have decreed its success.

Sony-Ericsson has equipped its devices with Windows Phone and Android in the latest Xperia products, while Research in Motion (RIM) is linked to its successful BlackBerry product, now equipped with the operating BlackBerry OS system version 10 as other mobile devices provided by such company. LG instead equips its mobile devices with Android.

iOS and Android actually are predominant in the market of mobile OS, but other OS (e.g., Windows Phone, Blackberry [5]) could, however, have their market and could be the target platform for a developer. An app provider very often has to optimize the software for a specific platform or device. In order to obtain the best performance and thus to maximize the mobile experience, he/she should know the devices' various mobile platforms. An app depends strongly on the mobile context, then on the type of hardware device, the software platform and the network used for data transmission. Then apps final delivery is usually done by means of stores. The user searches on a store the app and, when selected, it is automatically installed on the device. There is no way to test the app before downloading and this makes it difficult for the user to select the application on the basis of the available information. Considering the spread of such devices, Fig. 3 displays mobile users' main activities.

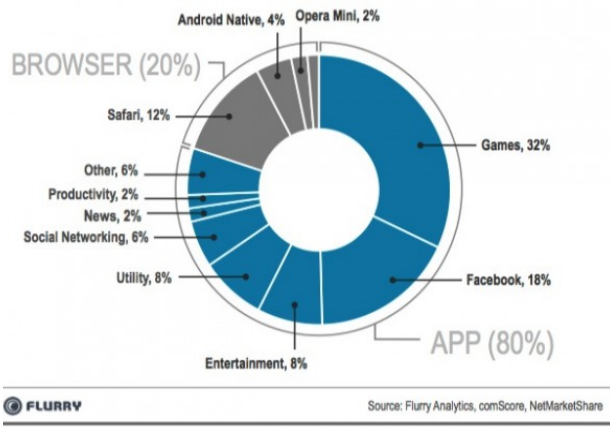

Fig. 3 Time spent on iOS and Android device (April 2013)

You see as the percentage of time spent on iOS and Android devices in the use of apps is greater than the time spent in web browsing. This clearly indicates the importance of an app.

Very often the factors that influence a user in downloading an app, apart from the choice of the category, are the price, a clear description of the purpose of the sample images, and the vote given by the users. Rarely a developer can affect the price in an important way, considering the fact that users prefer to download free apps, but it can affect the other elements. The description and pictures showing the app in action are very important, since they are the way in which a user can understand how an app behaves. Once the user has downloaded the application, he/she will expect that the app responds to his/her needs. A simple and intuitive application is the best method to be appreciated by a user. If the apps is delivered by the reference store, there are other factors that many influence the choice. Each store in fact implements an evaluation method based on a vote that the user needs to do. A developer must keep in mind that app' future updates should be able to improve the user experience and remedy the possible negative criticism.

Focusing on store distribution channel, each store offers different business models to take a revenue from apps, and studies demonstrate that this is the preferred channel to search for an app. Recent study by Canalys analyst firm on the main store (Apple's AppStore, Google Play, Windows Phone and Blackberry Store World) [7] shows a constant growth in apps sales during the first quarter of 2013. Between the stores, Apple store records the highest turnover of three-quarters of the total value of the market (74%), while Google Play is the first framework for number of app downloads (51%). The other stores are more detached, but still remain the point of reference for users and developers of related ecosystems.

2.2 Apps business models on mobile stores

On a store, app are usually downloaded according a license that is set by the developers. However each store, as the Table 1 shows, offers a specific program to sell the app ranging from free/paid app for download to the others business models (e.g., in-app purchase and in-app advertising). The first distinction is about free apps and paid apps. Then as regards the ways to monetize an app, the two main models are pay per download or premium offering and in app purchases (IAP). Pay for download method was the first model adopted especially in the App Store with different values of fees required (e.g., most apps requires a little fee of $0.99 or $1.99). Each user pays the app before downloading and installing into the device. The in-app purchases (IAPs) is a model that lets a user to download an app both for free (in this case the model is known as freemium) or paid, and requires a payment if the user want to enable some added features or functionalities, make some upgrades or have game advancement (e.g., enabling levels in a game). Another source of revenue could be the in-app advertising or in-app ads.

Table 1: Business models for apps selling in the different stores

	Fixed tires (e.g. Pay for download)	***Additional models***
App Store	Fixed tires: free, $0.99, $1.99, $2.99, … Exchange rate fixed and controlled by Apple. Price can be changed anytime Fremium model generates 72% of total revenues of paid apps	- In app purchases (offer additional digital content, functionality, services or subscription within the paid or free apps); - iAd rich media ads (exploiting the Apple's digital advertising platform and adds an ass to receive 70% of the net ad revenue generated) - volume purchase program (a specific program for business and education institution to purchase apps in volume)
Google Play	Free (no charge to download) and priced (user charged before download)	-in app products and subscriptions (e.g., one-time purchases and auto-renewing subscription from inside the app, freemium); - ad-supported model (ads apps with the use of AdMob integration or distribution for free and selling in-app products or advertising)
Window Phone Store	Free and paid apps whose base price can be selected by country	- in-app product (e.g., trial apps and apps divided into consumable or durable); - ad-funded apps (free apps with ads made by inserted code thanks to Microsoft Advertising service
Blackberry World	Free and paid (purchase before downloading)	-7-day, 30-day subscription (trial period after than users pay a fee to renew their subscription)

Other analysis however have shown (e.g., by Distimo [8]) that the in-app purchases in the fremium model generate the majority of revenue in the app stores. Finally the in-app advertising model uses the store advertising platform or service and consists in inserting specific advertising lines to the app code in order to take a revenue from such publicity. According to recent statistics ads through mobile or mobile advertising is increasing, even if this model should necessary be different from the traditional model of ads that dominates the Web sites. In many cases, getting an advertising message when download an app for free, seems to be a price that users are willing to pay for having no costs. As the Table 1 describes, there are different models of advertising such as affiliate marketing that promotes other mobile apps in order to earn commissions when the app is purchased and downloaded.

In any case the best strategy for marketing and app success (Fig. 4) depends on several factors, including the buying habits of the target audience and the type of application. For example according several analysts, Apple users are more used to pay for their apps and so paid apps are normally. In other contexts, the fremium model seems to prevail, since the app is offered for free and you only pay for enhanced functionalities. However, considering the number of free apps and the low revenue from advertising network, most developer use other channels to make money from creating mobile software.

Fig. 4 Good practice for app design, marketing vs. milestones for app success

A recent report [9] shows as the major revenue comes from other sources that app store sales and advertising such as e-commerce licensing and commissioned app-making. Probably the top revenue-generating apps on the

store belong to the game category or in some cases to some mobile messaging app.

2.3 Apps category

There are categories of apps that seems to be more appealing in a mobile context. For example the research company Nielsen describes as in Brazil and in Italy, users prefer to surf the Internet with the mobile device rather than downloading apps, while in countries like Australia, India, South Korea and Turkey the use of apps and web browsing is comparable. Mobile apps and websites for mobile devices should be different things referring to two different activities, but sometimes many apps are developed as versions of mobile optimized websites or just links to websites. In any case, mobile websites are designed to work on any mobile browser, while generally the apps are designed for one or a few mobile platforms. Focusing on apps, statistics on major stores shows as the games represent a great percentage of used apps, even if, as the Fig. 5 shows, maps (Google Maps) and social network apps are in the top positions (Facebook, YouTube, Google+, WeChat, Twitter) followed by messaging apps (e.g. Skype, Whatsapp).

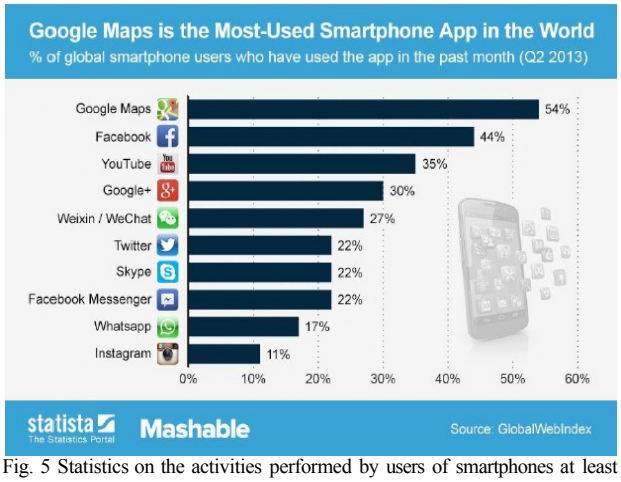

Fig. 5 Statistics on the activities performed by users of smartphones at least once a month in various countries (source: Nielsen survey, February 2013)

In some way software platforms discriminates also the apps categories considering that mobile apps are preinstalled on the device. If a developer needs to mediate between the actual usages of the app and its cost, these analysis are very important.
In the Italian context, for example, a report about mobile users behavior [10] shows as mobile users have on average 27 apps installed on the device, but about half use every month and every day. Of these most of them are free apps, and only 6 % of users have mainly paid app.

3. Strategies on apps development

App development project is not a simple action and requires critical decisions on strategies to be adopted considering the scope and the target. Usually apps manifest some peculiarities: even if developed with the same techniques used by other type of application software, they should be light, essential and easy to use considering the features of mobile devices. Even if a mobile app could be distributed via a website or e-mail, the main channel is the reference store of the mobile software platform. Since each store is proprietary, it imposes rigid policies on content and quality of the app influencing its development. Furthermore there is a cost variable to each store related to the store's developer program registration.
The focus is so on the mobile players and on the different aspects of app's development that range from the scope, the software tools needed and the business model.

3.1 Tools needed for mobile apps development

Considering the development methods, the choice of developing a native, web or hybrid apps includes the choice of the tools and the frameworks. Table 2 summarizes the different tools available for native, web and hybrid apps using several mobile platforms.

Table 2: Tools and frameworks for mobile development

Mobile software platform	***Development tools for native, web and hybrid app***
iOS	***App lifetime cycle:*** Develop, test, distribute maintain; ***Native app develop tools:*** Xcode (Knowledge of Objective-C). Native .IPA file to deploy on the device or on the store; ***Web app develop tools:*** Web frameworks Javascript libraries (e.g., jQuery mobile, Zepto) Knowledge of HTML5, CCS3, Javascript. ***Hybrid app tools:*** Phonegap, Titanium AppCelerator, RhoMobile suite by Motorola and Rhodes open source Ruby-based framework to build native apps, eMobc
Android	***App lifetime cycle:*** Setup, develop, debug&test, distribute ***Native app develop tools:*** Android development kit (ADT) or Android Studio (knowledge of Java) – packaged as apk file ***Web app develop tools:*** Web frameworks and Javascript libraries. ***Hybrid app tools:*** Phonegap, Titanium Appc, RhoMobile suite, eMobc
Windows	***App lifetime cycle:*** design, develop, distribute

Phone	***Native app develop tools:*** Windows Phone SDK (along with Visual Studio) Knowledge of C#, C++, Visual Basic. Packaged as XAP file. ***Web app develop tools:*** Web frameworks and Javascript libraries. ***Hybrid app tools:*** Phonegap, Titanium AppCelerator, RhoMobile suite
Blackberry OS	***App lifetime cycle:*** design, develop, distribute ***Native app develop tools:*** Momentic IDE (Blackberry 10 Native SDK) ***Web app develop tools:*** Web frameworks and Javascript libraries. ***Hybrid app tools:*** Phonegap, Titanium AppCelerator, RhoMobile suite

For native apps development, each mobile software platform provides a custom software development kit (SDK) that consists in a set of tools allowing you to design, develop, build and manage the process of app development. Most of them enable you to create a package version of the app suited to be published on the target store. Generally in each SDK there are compilers to translate the source code in the mobile platform reference language (e.g., Objective-C for iOS, Java for Android, C# or C++ for Windows Phone, C/C++/Qt for Blackberry 10) into an executable, the standard libraries with public interfaces and other tools such as the visual editors and the simulators in order to help the development. The development environment takes advantages of the mobile operating system and the software component related to runtime environments (e.g., the Android Dalvik Virtual machine or the Windows WinRT runtime environment) and the applications frameworks (e.g., Windows Silverlight, Apple Cocoa Touch). Mobile native apps are designed for a target mobile platform by using native programming languages and because of that they have access to all features of the device ensuring optimum performance.

Normally they require the installation into the device, a local execution and not necessary they need an Internet connection (except for social network or messaging apps). The main distribution channel is the store.

Mobile web app are developed in accordance with the HTML5 framework that understand the language of style CSS3 and Javascript programming language implemented using different dialects. There are different HTML5, CSS3 and Javascript frameworks that simplify the development process and speed up the coding.

Table 3 shows an example list of frameworks that could be used to develop web apps. We tried to made a distinction between tools that are used to create web applications for desktop (i.e., 52 framework) from those optimized for a mobile environment (e.g., Zoey), from the Javascript libraries built-in for mobile web apps (e.g., jQuery mobile, Zepto.js). Focusing on mobile framework, some of this framework are simple JavaScript libraries offering the developer a set of useful built-in functions.

Table 3: Tools and frameworks for mobile development

Framework	***Software available***
HTML5 framework	**Iio Engine** -Open source lightweight framework for creating HTML5 applications with JS and canvas. SDK + debugging system + cross-platform deployment engine **LimeJS** -HTML5 game framework for building games that work on touch screens and desktop browser. Created with the Closure library by Google. **52 framework** -HTML5-CSS3 based framework **Kendo UI** – inclusive HTML5/JS framework for modern web and mobile app development
JS mobile libraries	**Zepto.js** – jS framework for mobile webkit browser compatible iQuery **jQuery mobile** – touch optimized web framework built on top of jQuery **M-project** – mobile HTML5 JS framework to build mobile apps easy and fast **Xui** – a super micro tiny DOM library for authoring html5 mobile web apps **Sencha Touch** – HTML5 mobile app to develop app that look and feel native on iOS and Android
HTML5-CSS3 mobile framework	**Zoey** -Lightweight framework for developing mobile apps Built on Zepto.js, support modern browsers (iOS 4.x and Android 4.x) **Jo** -Open source mobile application framework working on iOS, Android, Symbian, Safari, Chrome and dashboard widgets. Compatible with PhoneGap **Lungo.js** -Mobile framework that includes features of HTML5, CSS3 and Javascript (iOS, Android, Blackberyy, WebOs). Distribution also on stores **Junior** Front-end framework for building HTML5 mobile apps that look and behave native (uses Zepto.js and integration with backbone.js views + routers) **eMobc** – open source framework for generation of web, mobile and native iOS and Android apps using XML

The inclusion in the source code of the app is simple and consists, as the Fig. 6 shows for the case of the Zoey framework, in putting in the scripts and the built-in style sheets inside the specific tags of the HTML5 code (e.g., the *script* and the *link* tags).

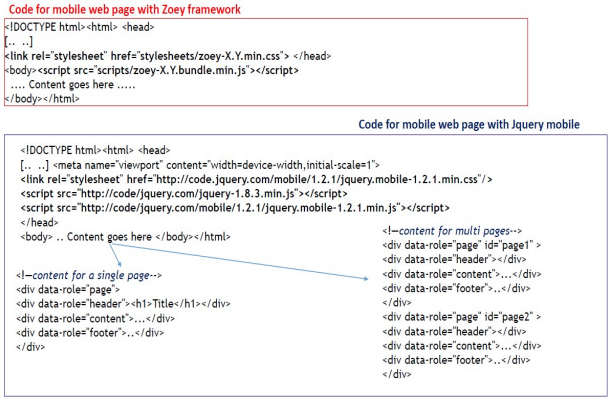

Fig. 6 Tools and frameworks for mobile development

The same figure shows an example of a web page coded as single o multi-web page that uses the JQuery mobile framework by including the Javascript library (*jquery-mobile.js)* and the style sheet (*jquery-mobile.css*) in the header section of the web page. Once included, the various functions coded in the library and the instructions inside the style sheet could be called directly in the body of the page. Other frameworks are more complex and also include a developing environment and the command line tools. Moreover they could be targeted to some platforms or development languages as the Fig. 7 shows.

Fig. 7 A comparison between mobile frameworks and functionalities.

For example Sencha company offers a set of products that allow to design (Sencha Architect), develop (Sencha Ext JS, Sencha GXT, and Sencha Touch) and deploy (Sencha Space) apps for mobile and desktop. Of these products, Sencha releases some of these software with an open source license as the case of Sencha Touch that is the framework dedicated to develop HTML5 Mobile app for iOS, Android and Blackberry. Since we focus on an open source framework that could allow to develop a mobile app, we examine, from the technical skill required, three products that are Sencha Touch, Phonegap/Apache Cordova and Appcelerator Titanium.

3.1.1 Sencha Touch

Sencha Touch is a Model View Controller (MVC) JS framework that appears to be more complex than a single library as seen before. Its use requires to download the Sencha Touch SDK and Sencha Cmd that install other software like Ant, Ruby, Sass and Compass. Sencha cmd is necessary to generate the app, since it gives a set of command line tools that allows to create the skeleton for the application available through a web address. The skeleton is composed by some directories (e.g., *app* that contains the models, *views*, *controllers* and *stores*) and files useful for the app. The *app.js* file is the entry point of the app. It contains the *launch* function where the application logic is inserted. The app is so made by an *index.html* file and the Javascript file that contains the source code of the app. Sencha touch comes with a set of components, objects, functions and code ready to use. A Sencha touch app is web-based, but the framework offers a way to include JS commands (by using native API with Ext.device function) in order to be translated into native functions (e.g., orientation, notification) when the app is compiled. The options to be set in order to transform the app into native are configured in the *packager.json* file.
An example of the structure of a Sencha touch app is shown in Fig. 8.

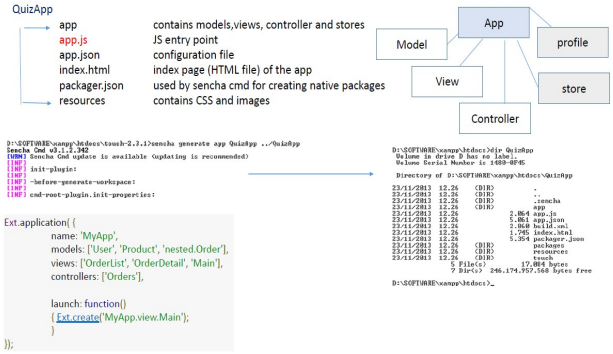

Fig. 8 Tools and frameworks for mobile development

After the writing of the code, following main components of an app (views, controllers and data), an app could be packaged for a native provision. The app packaging process is similar for iOS or Android devices. Each platforms differs by the configuration file. iOS requires the registration to the Apple iOS provisioning profiles portal in order to obtain a certificate, identity devices and an App ID. Also Android requires a certificate to sign the application by using the Android SDK Manager. For Blackberry and Windows Phone App, Sencha touch in the last version (2.3) introduces tools and an API for working with Cordova and Phonegap in order that these frameworks could package the app for the use on these devices. In any case, developing with Sencha Touch

requires a set of actions that are not as simple as it seems. Probably using the entire Sencha ecosystem allows an easily coding by means of visual tools, but the other frameworks are quite expensive. If we focus on the open source version, technical skills are required.

3.1.2 Phonegap vs. Cordova

Abobe Phonegap is built on top of Apache Cordova that is a platform providing API for building native mobile applications using the HTML5 framework. The difference is on how their packaging tools are implemented. Phonegap provides a remote building interface at Adobe Phonegap Build that emulates an app for a single platform in the cloud.
Phonegap requires the installation of the Nodejs framework in order to install all the packages needed for the app creation. Then it is required the presence on the local system of the different SDKs.
As the Fig. 9 shows, the creation and the execution of an app by means of the Phonegap command line tool, requires the sign up to the Phonegap Build portal and thus of an Adobe ID account.

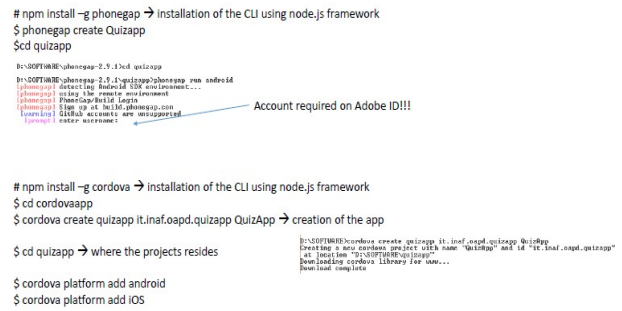

Fig. 9 Steps required for Phonegap/Cordova installation

Cordova packaging tools allow to build apps on a local computer. In any case it is necessary to have a certificate and AppID for each platform for the market in which a developer what to distribute the app. Apache Cordova requires the installation of Node.js (the JS platform used to build network applications), that is used to install locally the Cordova software. When installation is completed, the Cordova tools allow to create the main structure of the app inside a specific directory. When it is necessary to add platforms that are needed to build the project for the target mobile platform (e.g., iOS, android, wp8, blackberry). The ability to run the commands depends on whether the local machine support each SDK (e.g., a Windows-based machine lacks of the iOS SDK).
The cordova command allows to build the app for the target platform (i.e., using the build option), and to test the app on an emulator or a device (i.e., using the emulate option). Also in this case, the development of an app requires technical skills and the knowledge of the target mobile platforms and their development kits.

3.1.3 Appcelerator Titanium

Appcelerator company provides two main products: the Appcelerator platform designed to enterprise that provides a single open, cloud-based platform to deliver native cross-platform apps and Titanium designed to developers to create cloud-connected native apps.
Titanium is a development environment that includes an open source SDK to support various devices and mobile operating systems APIs, an integrated development environment (Titanium Studio), an MVC framework (Alloy) and cloud services to deliver apps.
Titanium products are provided as free, but it is required the registration to the site. Titanium Studio (Fig. 10) is a complete product that requires technical skills in order to be used. The software is installed on the local machine, but require the signing to the platform in order to work.

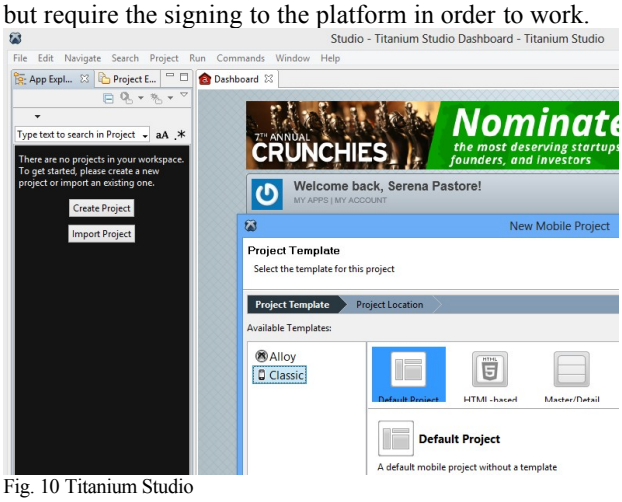

Fig. 10 Titanium Studio

3.2 A comparison between some strategies and tools

A comparison between native, web and hybrid apps based on some criteria rated on a scale of 0 to 5 is shown in Fig. 11. Native apps, HTML5 apps, Phonegap apps and Titanium apps are compared according the following criteria: the features of the app, the user experience and the performance, the monetization, the distribution, the updates, the cost, the platforms fragmentation and security.

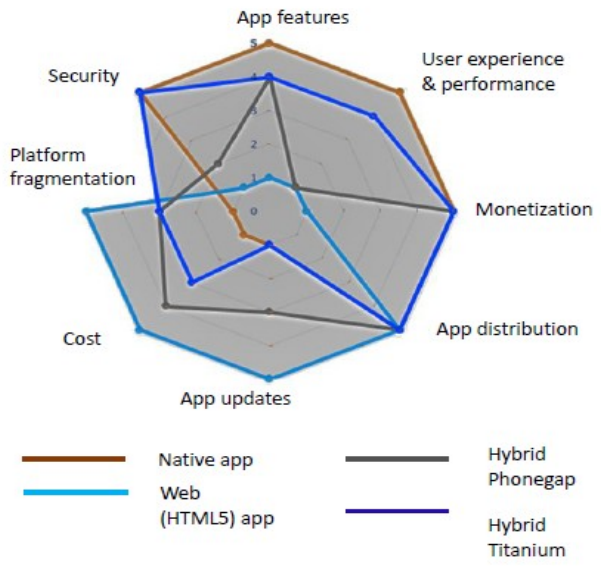

Fig. 11 Evaluation of methods of analysis for application development features.

As far as apps features, native (score 5) and hybrid (score 4) apps offer the possibility to access to the various resources of the device directly or via plugins or modules provided by cross-platform tools. HTML5-based web app instead (score 1), despite the rapid evolution of the technology, have not yet reached the same level as the other and do not allow to exploit the full capabilities of the device on which they operate.

Focusing on the user experience and the performance, since the performance evaluation is done according to the graphics and the load time of an app, native apps have the highest score (5), followed by hybrid app Titanium (4), while the others methods have the minimum score. Note that a poor performance lead to app failure, since user wants an app that runs fast and with high performance. Such features till now are possible only on native apps. The user experience of the app Titanium is very close to that of a native app, while a Phonegap app or a web app may have performance issues related, for example, to the time of loading a page, maybe because of the number of files relating to external style sheets or scripts to load.

The gain is undoubtedly linked to the method of distribution in the store and that is why hybrid native apps and have maximum score than HTML5 -based app. The app store distribution lead to a greater visibility with a big chance of being purchased and create a revenue for developers, although the competition is very high. The web app is distributed via the website and therefore is less visible since it is related to its availability through the search engines.

Some research (e.g., Canalys) reveals how Google Play is the store that has achieved the highest number of download (51% of total downloads carried out on the App Store, Google Play, Windows Phone and BlackBerry Store), but the App Store is the one that receives the most revenue getting 74% of the total profits of the four analyzed store. Even if the download of apps is expanding, the gain does not increase in proportion, since the percentage of free download is approximately 90% of all installed apps. Native app has thus a low score compared to other solutions whereas in the first case the strategy is imposed by the store.

Different scores are related to the updates that solve bugs or made apps' improvements. The web app has the highest score given that the update is automatic, a Phonegap app has an average score considered that is encoded with web technologies. Conversely a native app and a Titanium app have a low score considering that the updates imply the user having to re-download the app or must wait since also updates are subjected to the store's approval process.

The costs are related to the number of platforms to be achieved and the type of target device. The costs associated with the web app are lower (high score) considered the large number of developers with knowledge of web technologies and thus their competition. The other type of app have a cost that depends on the knowledge of the tools and the cross-platform software. The creation of an app supported by more than two platforms involves a high cost when developing a native app. But also in the web-based approach, there is a requirement for the app depending the different versions of the mobile browsers installed in the mobile devices.

The platform's fragmentation means that different versions of the mobile operating system may or may not enable a functionality. The fragmentation has few disadvantages in web-based app, app average in hybrid app and many issues in native apps. Since there may be several versions of the same operating system (e.g. Android) active on different devices, development must take into account the different hardware configurations which every manufacturer adopts. A native app created for the latest version of the operating system may not work on systems that are equipped with the previous versions.

Also a web app manifests a fragmentation problem that concerns the mobile web browser each of which has various versions and different engines that may or may not implement the language-specific content. In this case, the developers are obliged to test a web-based app on different devices and browsers. A Phonegap app is able to manage this fragmentation since it supports multiple

software platforms at the same time. Conversely, the extension of a product created with the Titanium framework to all these operating systems would be extremely expensive.

Finally, the security aspect presents a high score for a native and a Titanium app and a low score for a Phonegap app or a web-based app. The source code of a web-based app is easily recoverable and there are several ways to attack the security of a web app. On the other hand there are techniques to secure a web app. HTML5 provides the ability to cache data within the browser, and considering that the app manage data on the device, these data should be adequately protected and encrypted. The app can use the native API of its operating systems to encrypt the stored data, but this is not possible to implement such technology with web languages.

As widely stated, the programmer is placed in front of a series of choices. The final decisions about the tools to be adopted are a compromise between the features and the skills needed.

4. Conclusions

Developing a mobile app is not simple as it appears and requires a carefully analysis on the mobile ecosystem and some decisions about the strategies in the development. The choice of a cross-platform tool that allows to distribute an app on more than one mobile operation system is a choice that requires technical knowledge and skills comparable to those required in the development of a native app. This is particularly true when we focus on open source tools. Solutions like Sencha Touch or Cordova could really help to develop various version of the same app targeted to different mobile platforms, but require time and cost. Moreover the mobile world is an evolving and interesting market, but the revenue for an app developer is not so sure. Even in a research environment where we work, it is necessary to make an analysis between the costs related to the development and distribution of an app and the return in term of user satisfaction and knowledge of our activities.

References

[1] F. Sheares, Power management in mobile devices, newness, Elsevier, 2008.
[2] L. Columbus, 2013 Roundup of Samrtphone and Tablet forecasts and market estimates, Forbes, Jan. 2013. At url: http://www.forbes.com/sites/louiscolumbus/2013/01/17/2013-roundup-of-mobility-forecasts-and-market-estimates/
[3] S. Pastore, The role of open web standards for website development adhering to the one web vision. International Journal of Engineering and Technology (IJET), Vol.2, No.11. November 2012. ISSN 2049-3444.
[4] M. Pilgrim, HTML5: Up and running, O'Reilly, 2010
[5] S. Pastore, Mobile platforms and apps cross-platforms development tools, Int. Journal of Engineering Research and Application www. ijera.com. ISSN : 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp.521-531.
[6] T. Pierce, Appreneur: secrets to success in the App store, Apress, 2013
[7] T. Ferguson, App store revenue, downloads continue upward trajectory – Canalys, Apr. 2013, at url: Mobilewordllive.com
[8] H. Koekkoek, Publication: how the most successful apps monetize their user bade, Distimo.com publication, March. 2013.
[9] C. O'Connor, Evolution of the app business model, Boston.com Aug, 2013
[10] L. Lazzarin, Il Mobile Internet italiano vola a 1,25 miliardi, Content e app a 620 milion, wireless4innovation magazine, Jun 2013.

Serena Pastore Researcher/technologist from 2007 in ICT at the Italian National Institute of Astrophysics (INAF) and University teaching professor in some Italian University. Current research interest are on Internet and web standards focusing on the mobile environment, distributed network paradigms (grid and cloud), open source products.