

Multi-Criteria Analysis and Advanced Comparative Study between M-learning Development Approaches

<https://doi.org/10.3991/ijim.v12i3.8083>

Youness Zidoun^(✉), Rachid Dehbi, Mohamed Talea
Hassan II University of Casablanca, Casablanca, Morocco
youness.zidoun@gmail.com

Abstract—With the great advances in the world of mobile learning, development approaches are a very important challenge in designing an M-learning platform for instructors. The development can quickly become a handicap to the integration of certain functionalities and features if the approach is not very well thought and adapted to the pedagogical model designed for the platform. The purpose of this article is to clarify certain comparative aspects between the various methods of development of mobile learning platforms, their ordering and processing and the advantages and disadvantages of each approach. The authors present a multi-criteria analysis which can help instructors in decision-making by assisting their choice of the mobile development approach according to the most important criteria. This paper constitutes an overview of the m-learning platforms from design implementation point of view and will help researchers implement the pedagogical design model described in previous work.

Keywords—M-learning, mobile processing, development approach, MCDA, comparative study

1 Introduction

M-learning has emerged as a new paradigm in the world of digital learning; it offers multiple opportunities for learners and instructors. Among the features that distinguishes the most this mode of learning, the flexibility. The possibility to learn anywhere and at anytime can be both helpful and frightening at the same time, especially for instructors who are not yet adapted to the age of digital learning.

However, learner's perceptions are very enthusiastic. The authors already covered this matter with a study about student's perception about mobile learning in a previous article showing the increasing interest and readiness regarding m-learning in Morocco [1].

The great attention paid to mobile learning by all the conferences and publications since early 2000, still has not allowed a clear common definition and an understanding of it. There are some interchangeable terms such as wireless, ubiquitous, seamless, nomadic or pervasive learning/education, as well as mobile computer-supported collaborative learning, and mobile e-learning.

Frameworks are considerable tools in m-learning projects, since they allow a clear modeling of all factors that orbit around the platforms, their involvement level and even their impact.

Plenty of frameworks are available when it comes to TIC integration in education, for example: TCI (Theme Centered Interaction) by Cohn and Matzdorf, Conversational Framework by Laurillard, TPACK by Koehler and Mishra, SAMR model by Puentedura or even the Task model for mobile learners by Taylor et al and Sharples et al. These frameworks can be used either for designing mobile learning projects or evaluating them [2].

Beside the framework, several choices have to be taken designing an m-learning project between authors of educational content, IT specialist or instructors and learners different considerations. The most important choice to make in mobile designs in general and more precisely in digital learning projects is the development platform. This choice can be imposed following an already existing need as it can be defined in adaptation to the target of the project.

However, there are many development tracks. The major ones are by default native applications and web based applications development or hybrid solutions.

A comparative study between leading approaches adopted in the design of m-learning project is presented in this article. The authors will describe each approach, its methods and standards [22]. Then, they will detail a comparative study specifying in the advantages and disadvantages covered by each approach.

2 Background and related works

The pedagogical framework helps a lot guiding a digital project, but when it comes to give the design a shape, the language is the first choice to make which can be double-edged; it can put users in difficulties sometimes or even disqualify them.

The importance of communication and data storage or even the user interface in mobile learning project, makes the choice of the development platform harder. The biggest platforms nowadays are completely different and clearly don't work the same way (APIs, IDEs...).

In a previous work, researchers found as expected that the most used OS by students are android, iOS and windows phone [1]. Each one use a different development platform specific to their operating system and their applications are distributed only through platforms that control its nature and content.

Many distinctions are more visible on deeper levels [3]:

1. Applications architectures
2. Access to features and resources
3. Security
4. Data storage
5. Graphics and user interface
6. Communication between applications

Given the large and different characteristics, and to remain in the aims of this article. Authors choose to shed light on the global approaches, clarify them and study the advantages and drawbacks of each one.

3 Mobile development approaches

3.1 Native applications

Native learning applications are mobile applications designed specifically for an operating system in order for it to run on it by Smartphone or tablets (iOS, Android, Windows Phone, etc.). These apps are developed in the core technologies of the mobile devices that are downloadable from the "store" of the main manufacturers namely Apple, Google and Windows. Table 1 shows the differences between the three biggest platforms in the mobile market.

Table 1. Difference between each mobile OS

	Android	iOS	Windows Phone
Owner	Google	Apple	Microsoft
Language	Java	Objective C	C#
Executable files	.apk	.app	.xap
Distribution	GooglePlay	AppStore	MarketPlace

The native application development environment provides access, tools, and widgets that allow the design and creation of some standard learning interfaces with a native user interaction experience. Learners will be able to access and use their devices as they usually do for social networking or texting, in a liberal and user-friendly manner. However, the instructor will have to make more effort in the design and if necessary he will need an expert intervention in order to avoid the bad surprises of bugs or maintenance complications thereafter.

Native technologies are often exploited to bring a high-end user experience with recurrent usage that justifies a download with an application presence on the user's desktop. They allow users to integrate functionalities related to the targeted operating system such as geographical position, camera, synchronization with the personal calendar, etc. Their environment provides several tools and design widgets for creating standard or custom interfaces with a native user interaction experience.

From a more technical point of view, Android, IOS and WP7 are similar on several points. In terms of development standard and basic structure, the three platforms have a configuration file, XML / XAML / XIB views with code behind, with their events and lifecycle pages and resources of all types . They still are very different when it comes to development languages as shown in table 1 above or even data storage. For example: IOS and Android do use the same database which is SQLite while Windows phone uses SQL CE [3].

For communication windows phone is very classic using the typical inter-process communication mechanisms. On his side android propose a new one namely binder whereas IOS uses URL schemes [4].

Phone features are important to consider too, for example for the GPS iOS uses the core location framework [5], android on his side relies on the android location service using the location API [6] while WP7 takes advantage of the GeoCoordinateWatcher class of the location object [7].

For the motion sensor, android uses multiple classes provided by the sensor framework which can allow the performances related to the sensor [8] while iOS utilize the CoreMotion framework [9]. WP7 operates with the reference of sensors object namely Microsoft.Devices.Sensors [10].

3.2 Web applications

Web apps are basically dynamic web pages or multiple web pages that are designed to work on smaller screens. The app is combined with a server programming side which offers the functionalities namely user's interactions, database connection and which generates the results to the browser used.

In the case of web apps development, there is no installation to perform and the updates are automatic, for each page change or consultation the Web app is reloaded. So the updates are transparent to users. Web apps are developed with the classic web technologies, like HTML5, CSS3 or javascript and to access these applications, the user will have to go through web browsers which are nowadays more advanced than ever and still on progress. More features are integrated with new browsers versions that can encourage the choice to develop a mobile learning web app [11].

The program libraries, components, and tools are organized in an architecture that allows developers to manage and create complex web application projects using a fast and efficient approach. Furthermore, Web application frameworks are designed to optimize programming and promote code reuse by defining folder structure and organization, documentation, guidelines [12].

The technology bar is lower for instructors who would like to develop their own m-learning applications; generally it is easier to start in mobile web development than in native or hybrid development. Each mobile device holds a very specific idea about what constitutes the size and resolution of the screen usable, which generates the additional task of testing on different devices. Another important point to consider is the incompatibility of browsers and especially on Android devices.

Distribution and support are much easier in web development than native applications. HTML5 has become a very popular technology for creating mobile web applications; Several UI frameworks are available to facilitate development that may be complex. IScroll does an extraordinary job of emulating momentum-style scrolling. JQuery Mobile and Sencha Touch offer elegant mobile components, with many plugins that offer everything from carousels to super-advanced controls [13].

So, if web apps development makes it easier to design, maintain and reach the widest range of devices, users can't access native features on the device and unfortunately will not have the familiarity of the native appearance and sensitivity. They will not be

able to use compound gestures that they know too, even that the latest browsers support hardware accelerated CSS3 animation properties providing smooth motion for sliding panels as well as transitions between screens. But progress is continually being made and more features are supported by newer versions of browsers nowadays.

3.3 Hybrid applications

Hybrid development incorporates the best of both the native and Web. Authors identify hybrid apps as a web app, mainly built using HTML and JavaScript, which is then enveloped inside a native container that gives access to native platform features. The entire package is downloadable on distribution platforms.

PhoneGap, Cordova and Appcelerator titanium are an example of the most popular containers for creating hybrid mobile applications.

Presentation, Content and performance are the three major goals of any mobile app design. The hybrid approach is centered on these goals assigning the best techniques possible to respond accurately and effectively to each goal individually. Developers can now with the hybrid way create advanced mobile applications that don't sacrifice the cool native capabilities and to implement such an app, there are two ways possible. The first one is to use the same structure as native applications and package HTML and JavaScript code inside the mobile application binary, in this case REST APIs will be used to communicate between the device and the cloud. The second consist on implementing the full web app from the server and using the container as a small shell over the UIWebView, in this case caching is highly recommended for better performances. Hybrid apps are more maintainable as long as the right framework is chosen, namely: KendoUI, jQuery mobile, Ionic, etc. Also, it is important to note that if the tools and the framework used are not up-to-date to provide a certain new features; it will be disadvantageous and instructors will not be able to implement features that are not provided by the hybrid app development framework and Tools [14].

Instructors can choose between using SPA (single-page app) or MPA (multi-page app), although the more advisable to use is the MPA considering the ease it provides in the short and long term. All views and controllers are carefully categorized into independent HTML and JS documents and this provides a clear MVC-based file structure, better memory management, easier error detection and much more. Even the global variables problem can be addressed in many ways and unlike what was the case a few years ago, the hybrid development has progressed a lot nowadays making it possible to quickly create complex applications with simple cross-view for example. By default, all hybrid mobile app frameworks depend on standard implementations of WebViews for the app presentation layer. Once the framework is chosen, many types of WebViews are available and the architecture of the application will be set at the moment the development approach is chosen. Done appropriately, hybrid apps can ensure a good performance and the idea of using WebViews turns to be much simpler and fun to work with [15].

Many combinations are possible with these kinds of applications and this is what makes it very useful and especially affordable compared to other approaches, but at

the same time every side of the application's design have to be considered before embarking on a development path.

4 Process distinctions

As authors analyze it in the previous chapter, the three major development approaches are technically very different and so is the processing of the mobile applications.

By creating a mobile app, several levels are designed appropriately to meet all user specifications and provide the best user experience possible. However, having access to the mobile services of any device while keeping the learner fully transparent to the software chain involved is totally handled differently by each of the three approaches cited.

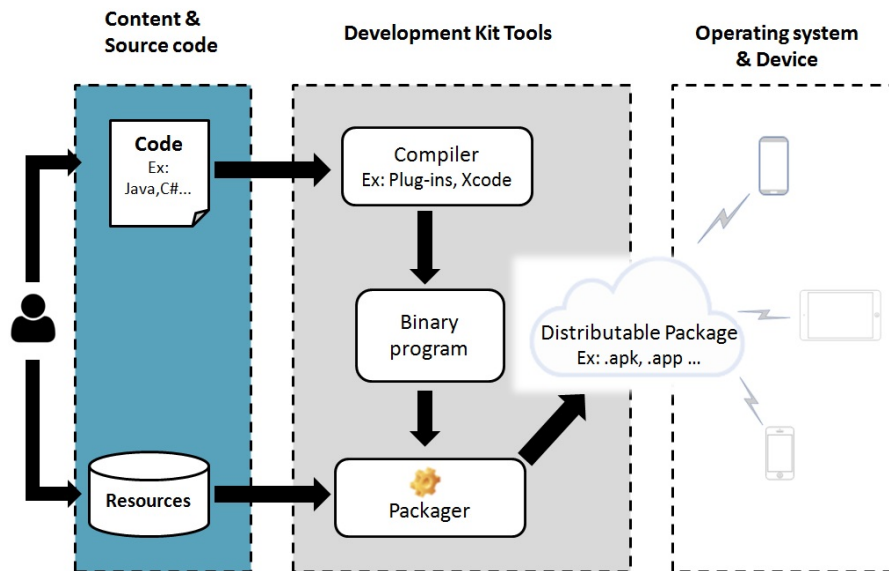


Fig. 1. Native applications development

Instructors have to keep in mind that mobile learning application has to get hall classroom students to be comfortable using it, and that what can explain some of the native choices, given that this is the old common idea in universities IT departments.

As illustrated in figure 1 the process of obtaining a binary executable program and preparing the app for distribution is essentially the same for all the leading mobile os. The source code of the app is written by developers in human readable form which is java or C# for example, additionally to further resources like videos, images or podcasts. They then compile the source code by using the tools supplied by mobile os vendors in order to generate a binary program and then package it together with the app's resources so as to obtain a package that can be distributed to mobile devices

through distribution platforms like Appstore and Google PlayStore. The native application is launched and executed directly by the operating system and will make explicitly use of the operating system APIs.

Each mobile operating system comes with its unique SDK but the process is the same for the majority of the leading OS in the market nowadays.

Even though the overall process is similar for all platforms as mentioned, the specifications are very different for each one. When the code is written on a particular platform it cannot be used on others, which means that the process of designing and maintaining a mobile app for multiple operating systems is very expensive and especially if the designer wants to offer a similar user experience through all platforms, he must have good knowledge of GUI toolkit specific to each one of them which can be even more expensive. But this choice of development comes with many benefits too, the interaction with the operating system is very fluid using practically all APIs available to access low level hardware services or high level ones giving the user a friendly interaction with the app and making it more practical for use.

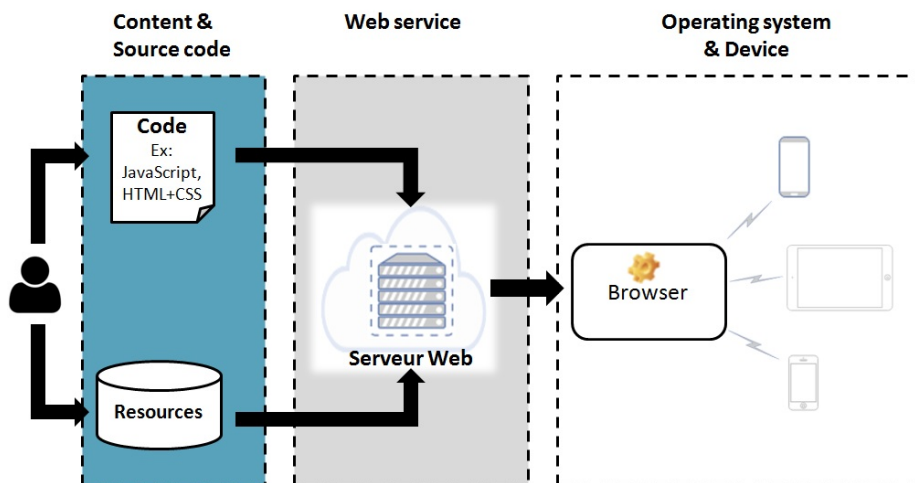


Fig. 2. Web applications development

Next category of apps is more recent than the first one explained, but it's important to note the difference between mobile web apps and mobile browsing first.

Mobile browsing is basically a mobile-optimized website accessible by a web browser which can recognize when they are being contacted by a mobile device, so they serve HTML pages designed to provide comfortable user experience on a small screen.

Nowadays, most of the mobile devices come with powerful browsers with support of HTML5 features, CSS3 and JavaScript. Mobile web apps are written entirely in these forms, served from a web server and they are practically indistinguishable to users from native applications. They are installed and launched like native apps, in

addition to that, they have a friendly interactive user interface (UI) optimized for touch screens and while data is retrieved from the web server as shown in figure 2 the logic for rendering the UI remains entirely on the mobile device which allow a high responsiveness and doesn't require any waiting time getting a response from the web server. Furthermore, the app can run in offline mode since the rendering is on the client side. Although this type of application has many advantages and is promising, there are some limitations that must be clarified. As mentioned before web apps run within the browser which is a native app itself, so the web app inherits the same limitations as the browser on a mobile device. They can access some hardware services in a limited fashion and even not accessing others at all. But still these technologies are evolving rapidly and a specific work is being conducted to allow more access to additional services. Without a doubt HTML5 will be the most widely supported cross-platform technology for mobile.

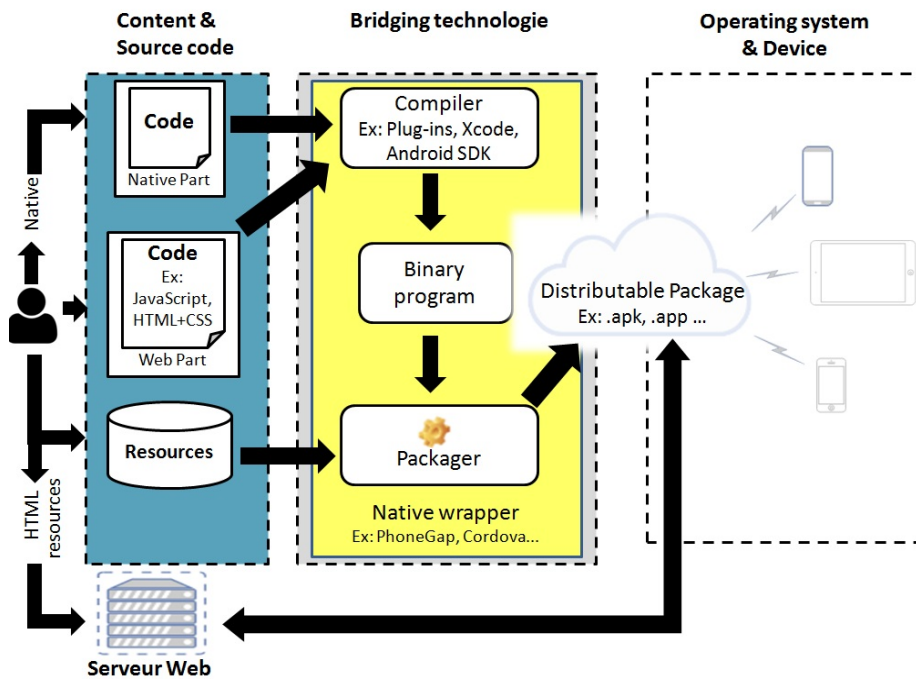


Fig. 3. Hybrid applications development

The third major approach consisting of hybrid applications can get a bit confusing when it comes to explain the processing of it. As shown in the figure 3 and as it can be understood from the nomination of it, hybrid apps take advantage of both the two previous approaches combining the strength of native apps and the flexibility of web apps. Hybrid apps can be defined as native ones with embedded HTML part either by downloading it directly from a web server or by packaging it within the app "see figure 3".

Two separate components can be identified, namely the native part and the web part. Native part can interact with all the operating system components and services (camera, microphones, calendar...). The second part which is the web one is rendered by the heart of the browser of the device and by default can only access the services available for the browser itself. This time the processing is different from a simple mobile web app, the native part has control over the rendering engine of web part and can create what is called a native wrapper or a bridge that allows the web part to access all the operating system APIs, it can either be created by the developer himself or by using an existing one such as PhoneGap. The user is still in complete transparency and does not have an idea about this software mix engaged. Virtual reality features can also be added given the flexibility of the services access.

There is many ways and scenarios for designing a hybrid app but the processing of it is much similar since the native part is practically the same for most of the operating systems on the market nowadays, the web part is free to suit every app designer depending on his preferences and conveniences. Authors chose to illustrate a full scale scenario where the HTML resources are downloadable directly from a web server imagining a small updates "see figure 3". .

5 SWOT analysis

In this chapter researchers introduce a SWOT analysis to summarize the strengths and weaknesses of the approaches presented:

Table 2. SWOT analysis

Native approach	
Strengths	<ul style="list-style-type: none"> • Multi-touch and UI gestures • Speed processing with high reliability • Better and responsive user experience • Push-notifications • Tools and debugging
Weaknesses	<ul style="list-style-type: none"> • Codebase for each platform • APIs differences • Updates and distribution approval
Opportunities	<ul style="list-style-type: none"> • Fluent access to native device services • Easy distribution
Threats	<ul style="list-style-type: none"> • High cost development and maintenance • Development Time
Web approach	
Strengths	<ul style="list-style-type: none"> • Various launch mechanisms • Interactive user interface • Touch optimized • Client side rendering • Offline support and data URLs • Continuous communication with the server
Weaknesses	<ul style="list-style-type: none"> • Code is executed by the browser not by the operating system

	<ul style="list-style-type: none"> • Limited device services access • Execution speed is still slower than other approaches • Animation may not behave seamlessly
Opportunities	<ul style="list-style-type: none"> • App optimization and cross-platform execution
Threats	<ul style="list-style-type: none"> • Less security with data being cloud exposed
Hybrid approach	
Strengths	<ul style="list-style-type: none"> • Full access to native APIs • Smooth user experience and responsive web-design • Native distribution • Applications are more easier to maintain • Security features can be implemented • Build it once and run it everywhere • Less expensive development
Weaknesses	<ul style="list-style-type: none"> • Increasing number of code paths • As good as the WebView • Not fast as native yet • Lack of performance on large-scale apps • Debugging can get more complicated in some cases
Opportunities	<ul style="list-style-type: none"> • Compatibility with changes in the mobile market
Threats	<ul style="list-style-type: none"> • Possible delays for adaptation to OS changes • Some mobile OS develop themselves in competitive way

6 Multi-criteria analysis

After presenting the different strengths and weaknesses of each approach, a multi-criteria decision analysis (MCDA) is presented which is a very efficient tool helping instructors make such a complex decision that a development path represents for a mobile learning application design [24] [25].

The score of each approach will be calculated based on a number of criteria [23]. From the study above, authors have identified eight criteria based on the approaches specifications and the SWOT analysis conducted. The eight criteria are: Development time, Interoperability, Reusability, Scalability, Maintainability, Flexibility, Customization, Interactivity and UI/UX.

6.1 Multi-criteria analysis method

Many possible mathematical methods are identified and can help for decision making, authors summarized them to three major families [16] [17] [18] [19]:

- Complete aggregation (top-down approach): Aggregating the X criteria to reduce the to a single criterion
- Partial aggregation (bottom-up approach): Comparing rankings or actions to each other and establish an outranking relations between them
- Local and iterative aggregation: It suits primarily starting solution; we conduct an iterative search to get a better solution thereafter.

Very commonly used, the Weight Sum Method (WSM) will be chosen for this study. This method offers the possibility to find the best approach by allocating a weight to each comparison criterion without a criterion penalizing another one [16] [17] [18] [19]. If there are m alternatives and n criteria, the best choice is the one that satisfies (in the case of maximization) the fellow expression [20]:

$$A^*_{\text{WSM-score}} = \max_i \sum_{j=1}^n a_{ij} w_j, \text{ for } i=1, 2, 3, \dots, m. \quad (*)$$

The additive utility assumption is the hypothesis that governs this model. The total value of each alternative is equal to the sum of the products given as (*).

$A^*_{\text{WSM-score}}$ is the WSM score of the best alternative, n is the number of decision criteria, a_{ij} is the actual value of the i -th alternative in terms of the j -th criterion, and w_j is the weight of importance of the j -th criterion [21].

6.2 Criteria and weight comparison

Eight criteria were identified from this study above which the comparative study will be based on. Researchers summed up all the approaches specifications and characteristics in these criteria bellow:

- C1- Development time : The efficiency of each approach is equivalent to development time won
- C2- Interoperability : Ease of integration, build it once and run it everywhere
- C3-Reusability: The reusability of the content should be important for learning materials
- C4-Scalability : The approach should offer a scalable environment where administration is easy and transparent
- C5- Maintainability : The key to a long term use and learner satisfaction
- C6- Flexibility : The mobility of users should stay untouchable
- C7- Customization : The administrator must have multiple windows to adjust content and configuration
- C8- Interactivity and UI/UX : The development approach have to guarantee the best user experience possible

Authors believe that the order of importance of these criteria is as follows:
 Interactivity and UI/UX > Development time = Maintainability = Interoperability = Scalability = Flexibility > Reusability = Customization

Table 3 present the WSM weights granted:

Table 3. Weight accorded to criteria

Criteria	Weight
Interactivity and UI/UX	3
Development time = Maintainability = Interoperability = Scalability = Flexibility	2
Reusability = Customization	1

6.3 Multi-criteria matrix choice

Multi-criteria choice matrix has to be achieved in order to apply the WSM method. With that being said, authors chose to score approaches criterion based on three values: 3 for good, 2 for medium and 1 for low [16] [17] [18] [19]. The matrix is presented as shown in figure 4.

Criteria	Approach	Native Development	Web Development	Hybrid Development
		Approach	Approach	Approach
Interactivity and UI/UX	3	3*3	1*3	3*3
Development time	2	1*2	3*2	3*2
Maintainability	2	3*2	3*2	2*2
Interoperability	2	1*2	3*2	3*2
Scalability	2	2*2	3*2	2*2
Flexibility	2	1*2	2*2	3*2
Reusability	1	1*1	3*1	2*1
Customization	1	3*1	3*1	2*1
WSM		1,9	2,4	2,6

Fig. 4. Multi-criteria matrix choice

6.4 Summary of the analysis

To summarize better the results a histogram was made showing the final scores of each development approach “see figure 5”.

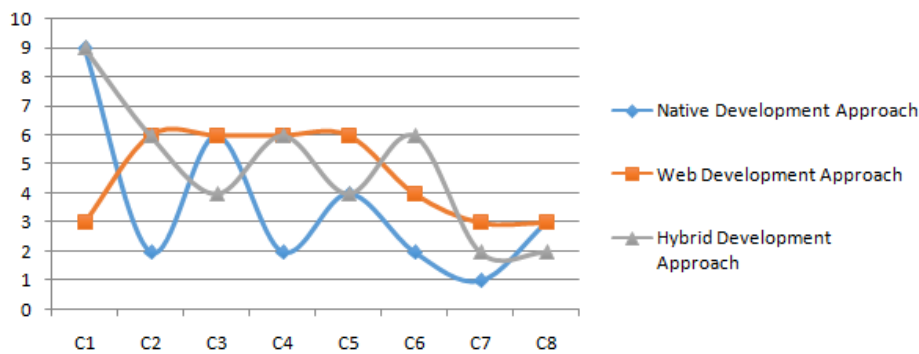


Fig. 5. Rating against criteria

According to this study, none of the development approaches reached the perfect score which can tell about the level of doubt instructors and professionals get when it's time to choose the development approach of the m-learning apps. It is clear that the native approach is far behind the two others, but still very present in nowadays platforms. Some IT departments continue to choose to go with the old comfortable and reliable solution for them even if they are passing by benefits of the others.

7 Conclusion

The choice of the mobile development approach is very important in each mobile learning project. In this paper, authors describe the processing of every approach, the differences between them, their strengths and weaknesses. The researchers presented a detailed comparative study using the WSM method of a multi-criteria analysis in order to rank these development approaches and help future instructors and learning mobile application designers to decision making. Authors have concluded that none of the development approaches have reached the perfect score and most of them are still evolving to be more efficient. The raising challenge and still in the same concept of helping instructors for decision making is to facilitate the integration of m-learning from the pedagogical model to the architecture and the development coding. For future work, the researchers aim to simplify m-learning integration by presenting solutions to guide pedagogical models through the process of realization according to instructor's conveniences and taking into account this comparative study. An implementation of our five axes integration m-learning framework will be proposed.

8 References

- [1] Zidoun, Y., Elarroum, F., Talea, M., & Dehbi, R. (2016). Students' Perception About Mobile Learning in Morocco: Survey Analysis. *International Journal of Interactive Mobile Technologies*, 10(4). <https://doi.org/10.3991/ijim.v10i4.5947>
- [2] Frohberg, D., Göth, C., & Schwabe, G. (2009). Mobile learning projects—a critical analysis of the state of the art. *Journal of computer assisted learning*, 25(4), 307-331. <https://doi.org/10.1111/j.1365-2729.2009.00315.x>
- [3] Charkaoui, S., Lahmar, E. B., Marzak, A., & Abdelbaki, I. (2016). Cross-platform Mobile Development based on MDA Approach. *International Journal of Interactive Mobile Technologies*, 10(4). <https://doi.org/10.3991/ijim.v10i4.5570>
- [4] Xing, L., Bai, X., Li, T., Wang, X., Chen, K., Liao, X., ... & Han, X. (2015, October). Cracking app isolation on apple: Unauthorized cross-app resource access on mac os. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (pp. 31-43). ACM. <https://doi.org/10.1145/2810103.2813609>
- [5] Apple, « Getting the User's Location », retrieved from <https://developer.apple.com>.
- [6] Android « Location Strategies », retrieved from <http://developer.android.com>
- [7] Misrosoft, « GeoCoordinateWatcher, classe », retrieved from <https://msdn.microsoft.com>
- [8] Android « Sensors Overview », retrieved from <http://developer.android.com>.
- [9] Apple, « Core Motion Framework Reference », retrieved from <https://developer.apple.com/documentation>
- [10] Misrosoft, « How to get data from the accelerometer sensor for Windows Phone 8 », retrieved from <https://msdn.microsoft.com>.
- [11] Zhao, K. (2016). U.S. Patent No. 9,420,031. Washington, DC: U.S. Patent and Trademark Office.
- [12] Web Application Frameworks, Comparison and Benchmarks, retrieved from <http://www.comentum.com/web-application-framework-comparison.html>
- [13] Kessin, Z. (2011). *Programming HTML5 applications: building powerful cross-platform environments in JavaScript*. " O'Reilly Media, Inc."
- [14] Mahato, R. (2016). *Hybrid Mobile Application Development*.
- [15] Pooryousef, S., & Amini, M. (2016, September). Fine-grained access control for hybrid mobile applications in Android using restricted paths. In *Information Security and Cryptography*

- tology (ISCISC), 2016 13th International Iranian Society of Cryptology Conference on (pp. 85-90). IEEE. <https://doi.org/10.1109/ISCISC.2016.7736456>
- [16] Zeleny, M. (1998). Multiple criteria decision making: Eight concepts of optimality. *Human Systems Management*, 17(2), 97-107.
- [17] Olson, D. L. (2001). Comparison of three multicriteria methods to predict known outcomes. *European Journal of operational research*, 130(3), 576-587. [https://doi.org/10.1016/S0377-2217\(99\)00416-6](https://doi.org/10.1016/S0377-2217(99)00416-6)
- [18] Lumanjibunga, L., & Mukonkimayekela, P. (2017). Utilisation D'une Methode Multicritere D'aide A La Decision Pour Le Choix D'une Structure De Donnees Dans Un Probleme De Gestion. *International Journal of Innovation and Applied Studies*, 20(2), 711.
- [19] Lehoux, N., & Vallée, P. (2004). *Analyse multicritères*. Ecole Polytechnique de Montréal: Montréal.
- [20] Fishburn, P. C. (1967). Interdependence and additivity in multivariate, unidimensional expected utility theory. *International Economic Review*, 8(3), 335-342. <https://doi.org/10.2307/2525541>
- [21] Triantaphyllou E. (2000) Multi-Criteria Decision Making Methods. In: *Multi-criteria Decision Making Methods: A Comparative Study*. Applied Optimization, vol 44. Springer, Boston, MA https://doi.org/10.1007/978-1-4757-3157-6_2
- [22] Bakhoui, A., Dehbi, R., & Talea, M. Selection of commercial and open source LMS: Multi-criteria analysis and advanced comparative study, *International Journal of Applied Engineering Research*, 11(7), pp. 4980-4989
- [23] Batouta, Z.I., Dehbi, R., Talea, M., Hajoui, O. , Multi-criteria analysis and advanced comparative study between automatic generation approaches in software engineering, *Journal of Theoretical and Applied Information Technology*, 81(3), pp. 609-620
- [24] Bakhoui, A., Dehbi, R., & Talea, M. (2016, December). Multiple Criteria Comparative Evaluation on the Interoperability of LMS by Applying COPRAS Method . *FTC 2016 - Future Technologies Conference 2016*. IEEE
- [25] Dehbi, R. (2017). Model driven engineering applied in e-learning development process: advanced comparative study with ROC multi-criteria analysis. *International Journal of Online Pedagogy and Course Design (IJOPCD)*, 7(1), 15-32. <https://doi.org/10.4018/IJOPCD.2017010102>

9 Authors

Youness Zidoun is a PhD candidate at the Faculty of Sciences Ben M'Sik, Information Processing Laboratory Hassan II University, Morocco. His research fields: m-learning frameworks, educational modeling, systems engineering.

Rachid Dehbi is an associate professor at the Faculty of Science Ain Chock, and member of LR2I laboratory (Research laboratory in Computer Innovation). He is responsible of research strategies and doctoral education training process at Hassan II University. His work focuses on E-learning, M-learning, E-health, M-health, Model Driven Engineering, Database Security, Big Data, and Internet of Thing.

Mohamed Talea is a professor of higher education at the Faculty of Sciences Ben M'Sik and Research Director of Information Processing Laboratory, Hassan II University, Morocco. His major research interest is on Systems engineering and security of information systems.

Article submitted 08 December 2017. Resubmitted 19 February 2018. Final acceptance 08 May 2018. Final version published as submitted by the authors.