International Journal of Mobile Computing and Multimedia Communications
Volume 13 • Issue 1

Decision Framework for Cross-Platform Mobile Development Frameworks Using an Integrated Multi-Criteria Decision-Making Methodology

Mohamed Lachgar, LTI Laboratory, ENSA, Chouaib Doukkali University, El Jadida, Morocco*
https://orcid.org/0000-0002-6155-3309

Mohamed Hanine, LTI Laboratory, ENSA, Chouaib Doukkali University, El Jadida, Morocco
https://orcid.org/0000-0001-5981-2511

Hanane Benouda, EST Khenifra, Sultan Moulay Slimane University, Beni Mellal, Morocco

Younes Ommane, MSDA, Mohamed VI Polytechnic University (UM6P), Ben Guerir, Morocco

ABSTRACT

Because of the growing popularity of smartphones and tablets, the use of mobile applications has exploded recently. However, the variety of mobile platforms compels developers to create an app for each, making the process more complex and expensive. Thus, open-source cross-platform mobile frameworks have been developed to address this problem, allowing the same code to be imported across various operating systems. This paper comes to propose a new framework for the selection of the appropriate platform for the implementation of a cross-platform mobile application. This framework is based on the most used multi-criteria decision-making (MCDM) methods, namely analytic hierarchy process (AHP) and technique for order preference by similarity to ideal solution (TOPSIS) methods are used. A demonstrative example is proposed to illustrate the suggested methodology.

KEYWORDS

AHP, Cross-Platform Mobile Development, Decision Framework, Multi-Criteria Decision Making, TOPSIS

INTRODUCTION

The mobile application market lives a continuous growth over the last decade, in 2020 the number of application downloads has exceeded the 218 billion globally, this growth is due to features and amazing benefits these systems offer to users, in order to save time and effort when searching.

Nonetheless, the development of mobile applications has become a difficult and an expensive task, due to the diversity of operating systems and multiple devices. Therefore, the industry is

DOI: 10.4018/IJMCMC.297928

*Corresponding Author

oriented towards the use of cross-platform (CP) development tools, which have enabled developers to target multiple mobile operating systems, using a single source code to deploy the application across multiple platforms, reducing development costs and implementation time, while providing a feel native to the end user.

Besides, companies face several cross-platform mobile development tools, which makes it very difficult to choose the right tool for the achievement of their applications. Therefore, architects must make this decision based on many criteria, to limit the costs and risks associated with projects. Furthermore, decision-making in selecting cross-platform mobile development tools has become more complex, due to the large number of platforms and approaches available in the market.

Choosing a framework for mobile app development is a crucial step that requires consideration of a wide number of elements. The most important variables in mobile app development are the app's end goal, as well as its requirements and major challenges, the most notable benefits of the chosen mobile app framework, and how those benefits align with the app's core characteristics, such as speed, security, performance, scalability, and so on.

The selection process is an important part of mobile app development, since it influences how the app is built and performs. Therefore, to ensure that an effective framework is put in place to help decision-makers (managers, developers, etc.) to choose the most appropriate tool with the least risk, this work made use of MCDM methods which have shown their effectiveness in several areas.

MCDM has proved to be a powerful and effective approach for tackling this type of selection problem due to the multi-criteria character of the tool selection. The analytical model integrated with the AHP and TOPSIS method will help to determine the most accurate judgment for tool selection, according to the specific requirements of the decision makers. This article suggests an integrated AHP-TOPSIS model taking account both qualitative and quantitative factors. From this point of view, AHP are often very useful in involving multiple decision makers with multiple conflicting criteria to succeed in consensus with the decision-making process. On the other side, the TOPSIS technique is employed to calculate the evaluations of the alternatives.

In this work, an in-depth study is carried out in order to determine the key criteria that intervene in the decision of choosing CP frameworks, based on the most relevant research works in the field. Then an evaluation of the importance of the criteria is conducted in collaboration with experts in mobile development with different tools. So as to provide a framework based on the AHP and TOPSIS methods to rank CP mobile development tools.

Without forgetting that the landscape of CP frameworks is very dynamic, so new frameworks are expected to emerge in the future. Thus, this work focused on illustrating the application and feasibility of AHP and TOPSIS methods in the area of CP tools evaluation and selection.

The remainder of this paper is structured as follows: Section 2 shed light on MCDM methods and mobile development approaches. A summary of existing methods and studies for selecting of cross-platform mobile development tools is presented in Section 3. In Section 4, AHP and TOPSIS methods are explained, respectively. In section 5, the proposed methodology is explained concisely, followed by an empirical study and a sensitivity analysis. Finally, the results of the article and suggestions for future studies are illustrated in the last section.

BACKGROUND

Multi-Criteria Decision Making

Since the 1970s, the MCDM methods are a rapidly developing research area. There are several organizations linked to the MCDM, including the International Society for Multi-Criteria Decision Making, the INFORMS section and the Working Group on Animal Resources Analysis in Europe.

Multi-Criteria Decision Analysis (MCDA) is a branch of operational research allowing the evaluation of several contradictory criteria in decision-making, whether in daily life and even in commercial, governmental or medical contexts. In our daily life, multiple criteria are implicitly evaluated to make decisions that are based on intuition. The MCDM is a set of methods for assisting in decision-making in order to reduce the ultimate decision-responsibilities maker's and provide a solution that meets the criteria in question. When faced with these types of problems, there is no single optimal solution, and it is necessary to use the preferences of the decision maker to differentiate the solutions. Nevertheless, the correct structure of complex problems and the considering of several criteria lead to more enlightened and more effective decisions.

This is the reason why different approaches and methods, often implemented by specialized decision-making software, have been developed and applied in a range of disciplines, ranging from politics and business to environment and business energy.

The MCDM methods are based on the knowledge in many areas including: mathematics, decision analysis, economy, computer technology, software engineering and information system and different areas of science.

These methods can be used in several fields such as: energy and power, automotive, agriculture, chemistry, maritime industry, health, construction and manufacturing. As well, several research works have been devoted to do use of these methods, authors in (Emovon & Oghenenyerovwho, 2020) reviewed a total of 55 scientific articles, published in high-ranking journals between 1994 and 2019 and were culled from different databases including Web of Science (WoS), Scopus and Google scholar, in which they applied the MCDM technique to analyzing material selection problems. The popular MCDM tools that have been applied in this study are AHP, TOPSIS, VIKOR, ELECTRE, and MAUT. Finally, the literature review revealed that the AHP method is the most often used strategy in construction, infrastructure, logistics, transportation, energy, and other disciplines, while TOPSIS and AHP are the most commonly used approaches in supply chain management.

Mobile Development Strategy

There are four main approaches (Zaragoza et al., 2016) (Nunkesser, 2018) (Lamhaddab et al., 2019) for developing mobile applications: Native, Web, Hybrid, and Cross-platform (CP):

- **Native**: native development involves the use of platform specific programming languages, SDKs, development environment, and other tools provided by operating system vendors. Therefore, developing native applications for multiple platforms requires the use of distinct technology stacks.
- **Mobile web**: the skills and methodology used in development are the same as those used in traditional "desktop" web development. Developers create webpages with HTML, JavaScript, and CSS, which are then viewed by mobile browsers. While some local caching is possible, most mobile web apps rely on a continual Internet connection and a web server to supply views and content as the user navigates around the app.
- **Hybrid**: this approach is based on WebView - which are a platform-specific components used to display web content directly in an app instead of a standard browser (Firefox or Chrome). In this way, each operating system displays the application in the same way, thus, the applications will work in a comparable way on all the devices.
- **Cross-platform (CP)**: In this approach, the tools used (Xamarin, Flutter, React Native, Native Script, etc.) make it possible to tailor every element of an application to each specific platform, ensuring not only maximum customisation, but also enhanced performance compared to the hybrid approach.

Each of the main mobile app development approach has its own set of defining and distinguishing qualities. As some of the approaches are constantly evolving, their definitions and classifications are hotly debated. It can be demanding for non-developers and persons, without prior expertise in this sector, to appreciate the definitions, differences and classifications of the diverse approaches and their possible implications. In this article we will focus on CP development tools. According to the work presented in, the technologies used for the implementation of native and web approaches are unique, for the native case, we use Java or Kotlin for Android, Swift or objective C for iOs, for the web case we use the web technologies (HTML, CSS, JavaScript, etc.). On the other hand, for the case of CP applications, a variety of tools are relied on by developers (e.g. React Native, Ionic, Xamarin, etc.). Thus, the goal is to propose a Framework allowing choosing the best tools based on MCDM.

RELATED WORKS

Firstly, this section gives typical scenarios of the use of particular multi-criteria decision methods in various fields of application, in order to figure out what methods would be employed in this project, secondly, an examination of the works that proposed assisting approaches in the evaluation and the selection of a cross-platform mobile development tools are provided.

The authors in (de FSM Russo & Camanho, 2015) propose a systematic literature review, which assesses how the AHP method is used in real-world situations, and how the criteria are being defined and measured. In this paper, a descriptive approach is utilized to increase knowledge about the AHP method and to clarify how criteria are managed and defined through this method to make a good decision.

In (Zlaugotne et al., 2020), authors compared five MCDA methods using the same data set, so as to see either ranking alternatives would be different or similar for each method, which turned out to be the best renewable energy technology in the Latvian case. The MCDA method is widely used in every field to figure out diverse decision problems by alternative evaluations and the choice of the suitable method would bring about a considerable impact on the results. MCDA methods chosen in the present paper are TOPSIS, COPRAS, VIKOR, PROMETHEE, GAIA and MULTIMOORA, since they have various methodologies on the best way to register elective values and there it was feasible to utilize measures, that can be characterized least or greatest as best value. On the contrary, the primary outcomes showed that TOPSIS, PROMETHEE-GAIA and VIKOR have similar priority selection and the most elevated positioning was chosen for hydropower plant. In contrast COPRAS and MULTIMOORA results were advantageous to Solar PV. Authors deduced that the best sustainable power innovation for Latvia is hydroelectric force plant (HPP) and the wind power plants (WPP) alternative took the lowest position, in light of the fact that in 3 out of 5 strategies are in the last rank.

The approach suggested in (Kabir et al., 2012), showed the impacting elements on the success in online retail service, at that point assessed and evaluated these elements through examining components using the AHP endowed with its fuzzy extension, specifically Fuzzy Analytic Hierarchy Process (FAHP). In this study, FAHP was effectively applied to put more emphasis the basic success factors, as well as to the TOPSIS method for consequent positioning. For this end, the benchmarking of the performance and the present company's position has been carried out, based on basic success factors among its contenders employing FAHP to attain the targeted standard of performance.

MCDM approaches have been utilized by several researchers to aid decision-makers in the evaluation and selection of software packages for the creation of mobile applications and the choosing of frameworks. In (Jadhav & Sonar, 2011) the authors describe a generic methodology for software selection, software evaluation criteria, and a hybrid knowledge-based system (HKBS) approach to help decision makers in the evaluation and selection of software packages such as Data Mining, CRM (Customer Relationship Management), ERP (Enterprise Resource Planning), and so on. This study provided a comprehensive list of software evaluation criteria that are common and could also be used to appraise any software package, comprising functionality, technicality, quality, cost and benefits, opinion, and output criteria. The hybrid knowledge-based system (HKBS) technique reported in this paper utilizes an integrated rule-based (RBR) approach to detect user needs for software packages, along with problem case-based reasoning (CBR) techniques to retrieve and compare prospective software packages to the package's user needs.

The authors also investigated and assessed the HKBS approach to other frequently used software evaluation techniques, such as the Weighted Scoring Method (WSM) and AHP, which implemented ActiveSMS, SMSDemon, GSMActive, and SMSZyneo as software components. Further to that, they conclude that HKBS outperforms AHP and WSM in terms of knowledge/experience reuse, problem-solving flexibility, and consistency and presentation of evaluation outcomes.

The paper (Khachouch et al., 2020) compared the merits and disadvantages of native, web, hybrid (Ionic Framework), cross-platform (Xamarin), modeling, cloud-based, and combined mobile app development approaches, as well as presented a decisional framework to pick among them. The proposed frameworks are based on answering multiple questions, with the weight assigned to each of these questions influencing the framework's ultimate conclusion and returning a final judgment for each mobile project setting (1: Not so important, 2: Important, and 3: Very important). The decision schema was presented using a UML activity diagram, in which each question is represented by an activity, and transitions between activities reflect the responses to those questions. In addition, the authors discuss the importance of the native approach in the absence of financial or human resource constraints. Without ignoring performance, which is a critical factor in the creation of mobile applications as discussed in (Biørn-Hansen et al., 2019). In the context of animations and transitions in mobile user interfaces, the article assesses the performance of mobile applications produced utilizing cross-platform frameworks. The authors deploy Hybrid, Interpreted, and Cross-compiled techniques, as well as a range of performance profiling tools. Frames per second (FPS), device memory consumption, Graphics Processing Unit (GPU) memory usage, and Central Processing Unit (CPU) usage were all examined in this study in order to determine the effect of transitions and animations in mobile user interfaces.

In (Hanine et al., 2016), the authors introduced an approach based on a combined multi-criteria decision-making process that would allow developers to select the appropriate ETL (Extract, Transform, and Load) software to create a viable investment market. As a result, the success or failure of every Business Intelligence (BI) project hinges on the choice of ETL software. The software prototype produced for this methodology is based on AHP for studying the structure of the ETL software selection problem and deriving weights for the given criteria, as well as the TOPSIS technique for generating alternative ratings.

These studies provided different applications based on their evaluation criteria using different MCDM methods as TOPSIS, AHP, ELECTRE, VIKOR and MAUT. Nonetheless, a substantial number of studies revealed that the AHP and TOPSIS methods are the most often used strategy for making decisions in infrastructure (Anastasiadou et al., 2021), energy (Sedghiyan et al., 2021), logistics (James et al., 2021), construction (Marzouk & Sabbah, 2021), transportation (Broniewicz & Ogrodnik, 2020), education (Mohammed et al., 2018), and other disciplines.

In this competitive market, CP mobile application development tools help to design high-performance mobile applications that fully match the customers' requirements. These tools are currently gaining popularity around the world due to their capacity to compile application source code for multiple supported operating systems. As a consequence, numerous research articles have addressed the subject of assisting developers in making the best decision possible given their limits and requirements.

The survey (Dalmasso et al., 2013) provides several decision criteria beyond portability issues for choosing an appropriate multiplatform tool for the development of mobile applications. To examine performance in terms of power consumption, CPU and memory usage, Android test apps are developed using PhoneGap, PhoneGap & JQuery mobile, PhoneGap & Sencha Touch 2.0 and Titanium. The authors found that PhoneGap consumes less power, CPU, and memory than other tools, but it doesn't have a very attractive user interface.

In (Mohamed & Abdelmounaïm, 2017), the authors introduced a methodology for selecting the best technique and tool for developing a mobile application based on a simple survey with binary questions and a set of criteria. The proposed framework is divided into two phases: the first allows for the deduction of the mobile development method (native, web, or hybrid) based on a completion percentage renowned as "precision," and the second determines the appropriate tool for each method whose precision exceeds 50% based on a set of relevant criteria.

Another research (Heitkötter et al., 2013) outlined a detailed set of criteria for assessing cross-platform development methodologies for mobile apps. The criteria have been organized into infrastructure perspectives, encompassing criteria pertaining to an app's life cycle, usage, operation, functionality, and development perspectives, which include all criteria that are directly relevant to the application's development process.

In (Rieger & Majchrzak, 2019), the authors also issued a detailed set of criteria for evaluating cross-platform development methodologies for mobile apps. Web apps, apps produced with Titanium or PhoneGap, and natively developed apps are all evaluated using the same set of criteria. The authors concluded that if a near resemblance to a native interface can be ignored and the maturity of cross platform techniques demonstrates that native development is not required when creating mobile applications, PhoneGap should be considered.

The authors in (Nawrocki et al., 2021) presented a comparative study of the most recognized mobile development frameworks namely Flutter, React native and Xamarin, of all the solutions tested in this study, Flutter appears to be the best overall.

All existing approaches are based on few performance criteria for mobile development tools. Since these approaches are based on precision calculation of the arbitrarily assigned weights to the used criteria, without any call for decision support methods, which is not enough for making a relevant and accurate decision. As a result, this paper tries to fill that gap by suggesting an improved version regarding the existing literature, i.e. MCDM methods that have shown their effectiveness in several related research areas.

Thus, this article suggests a decision framework for CP frameworks, based on an integrated AHP-TOPSIS model.

MULTI-CRITERIA DECISION-MAKING METHODS

AHP Method

The AHP method is among the most commonly used multi-criteria decision-making methods, which was initially introduced by Myers and Alpert in 1968 (Myers & Alpert, 1968) and subsequently refined by Thomas L. Saaty with his research in 1990 (Saaty, 1990), and the mathematical stages of the technique were therefore created. It's commonly used for analyzing and structuring of complicated decision-making issues. In this method, the decision problem is first broken down into different criteria (Dağdeviren et al., 2009). The AHP technique can be used to help decision-makers in calculating the weight of each criterion using pairwise comparison judgements, see (Liberatore & Nydick, 1997), (Yoo & Choi, 2006) and (Panda et al., 2014). In detail, the AHP stages are describes below (Saaty, 2008):

Stage 1: Establishes the decision context and organizes the criteria by grouping them hierarchically under high-level and lower-level goals.

Stage 2: Sets up a set of all judgements in the comparison matrix in which the set of items are compared to itself using the fundamental pairwise comparison scale provided in Table 1.

Stage 3: Finds the relative importance of the criteria by computing the eigenvectors according to the maximum eigenvalues.

Stage 4: Determines the consistency index (CI). The CI is calculated using the formula:

Table 1. Numerical scale of relative importance.

Scale	Definition
1	Element E1 has equal importance compared to E2
3	Element E1 has moderate importance compared to E2
5	Element E1 has strong importance compared E2
7	Element E1 has very strong or demonstrated importance compared to E2
9	Element E1 has extreme importance compared to E2

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{1}$$

where:

$\lambda_{\max}$: maximal eigenvalue
n: dimension of the matrix

Stage 5: Checks the matrixes' consistency factor. The CR is calculated using the formula:

$$CR = \frac{CI}{RCI} \tag{2}$$

with, the random consistency index (RCI) values are defined in Table 2.

Table 2. The random consistency index values.

Number of criteria (n)	1	2	3	4	5	6	7	8	9	10
RCI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

The matrix can be deemed to have sufficient consistency if the CR value < 0.10; otherwise, pairwise comparisons should be changed to decrease inconsistency.

TOPSIS Method

The TOPSIS method was pioneered by Hwang and Yoon and Wang (Hwang & Yoon, 1981), to solve multiple criteria decision making (MCDM) problems founded on the principal that the alternative chosen should have the shortest distance from the ideal solution (A*) and the farthest from the negative ideal solution (A−). In the TOPSIS process, performance ratings and criteria weights are given as exact values (Lengacher & Cammarata, 2012). Lately, various important studies have focused on the TOPSIS technique and applied it in several areas like tourist destination evaluation, supplier selection, financial performance evaluation, evaluation of companies, location selection and ranking of carrier alternatives. Some case of these studies can be found in literature such as ERP platform selection

(Huiqun & Guang, 2012), customer-centric product design process (Lin et al., 2008), open-source EMR (Electronic Medical Record) software packages (Zaidan et al., 2015). The TOPSIS method consists of the following stages (Tsaur, 2011) (Ding, 2012):

Stage 1: Constructs a decision matrix for ranking:

$$\begin{matrix} & & w_1 & w_2 & \cdots & w_n \\ & & C_1 & C_2 & \cdots & C_n \end{matrix}$$

$$M = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{pmatrix} \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{pmatrix}$$

where $A_1, A_2, \ \ , A_m$ are possible alternatives, $C_1, C_2, \ \ , C_n$ are evaluation criteria, w_{ij} is the performance value of alternative A_i under criterion C_j, and w_j is the weight of criterion C_j.

Stage 2: Normalizes the decision matrix using the following equation:

$$n_{ij} = w_{ij} \Big/ \left(\sum_{j=1}^{m} \left(w_{ij}\right)^2 \right)^{\frac{1}{2}}, j = 1, \ldots, n, i = 1, \ldots, m \tag{3}$$

As a result of this normalization, every attribute has the same unit scale.

Stage 3: Multiplies the normalized decision matrix with its associated weightings to get the weighted normalized decision matrix:

$$e_{ij} = w_j \otimes n_{ij}, j = 1, \ldots, n, i = 1, \ldots, m \tag{4}$$

Stage 4: Identifies the ideal $\left(A^*\right)$ and the negative-ideal solutions (A^-) as follows:

- Ideal:

$$A^* = \left\{ e_1^+, \ldots, e_n^+ \right\}$$

$$A^* = \left\{ \left(\max_i e_{ij}, j \in J \right) \left(\min_i e_{ij}, j \in J' \right) \right\}, i = 1, 2, \ldots, m \tag{5}$$

- Negative-Ideal:

$$A^- = \left\{ e_1^-, \ldots, e_n^- \right\}$$

$$A^{-} = \left\{ \left(\underset{i}{min}\ e_{ij}, j \in J \right) \left(\underset{i}{max}\ e_{ij}, j \in J' \right) \right\},\ i = 1,2,\ldots,m \tag{6}$$

where J refers to the benefit criteria, and J' refers to the cost criteria.

Stage 5: Calculates the separation measure as follows:

- Ideal separation:

$$D_i^{*} = \left(\sum_{j=1}^{n} \left(e_{ij} - e_j^{+} \right)^2 \right)^{\frac{1}{2}}, i = 1,\ldots,m \tag{7}$$

- Negative-Ideal separation:

$$D_i^{-} = \left(\sum_{j=1}^{n} \left(e_{ij} - e_j^{-} \right)^2 \right)^{\frac{1}{2}}, i = 1,\ldots,m \tag{8}$$

Stage 6: Calculates the relative closeness to the optimal solution using Eq. (9):

$$R_i = \frac{D_i^{-}}{D_i^{*} + D_i^{-}}; i = 1,2,\ldots,m;\ 0 \le R_i^{*} \le 1 \tag{9}$$

Stage 7: Ranks the preference order.

Rank the best alternatives pursuant to R_i in descending order.

THE PROPOSED METHODOLOGY

In recent decades, various researchers have deployed their efforts to design the best decision-making approaches. In this work, the suggested methodology is intended to make the use of MCDM methods as efficient as possible. Thus, both different methods, namely AHP and TOPSIS, are coupled to rank alternative CP mobile development tools according to the defined criteria. The purpose of using the well-known AHP method is to organize the problem's decision hierarchy. Lastly, one of the most successful MCDM approaches, such as TOPSIS, may be used to classify the alternatives.

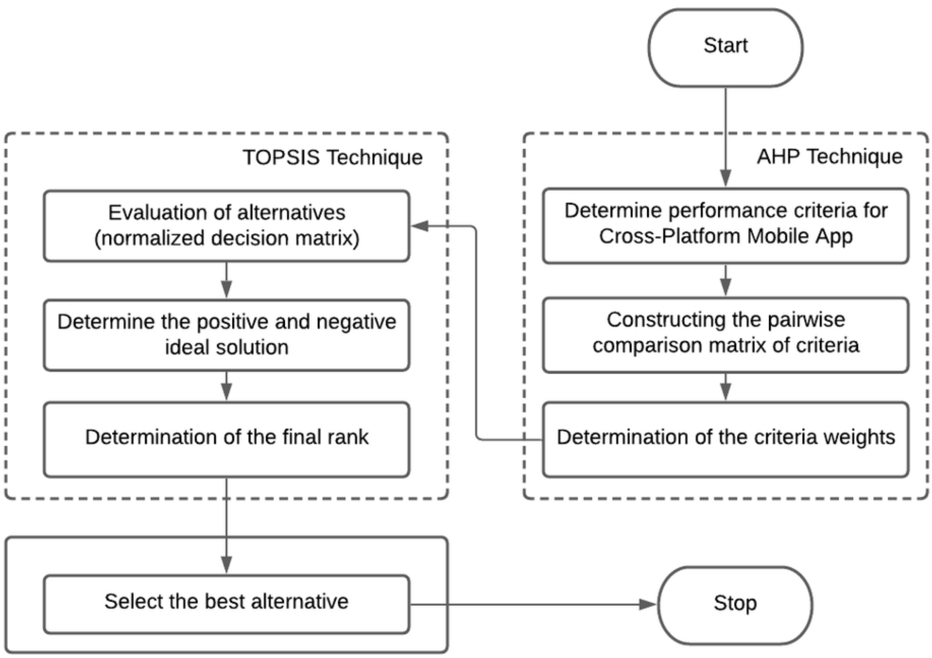
Below, the main stages of the integrated methodology proposed to be developed by decision-makers for the problem of selecting tools for the development of CP mobile applications are as follows:

- **Stage 1:** Defines the main and sub-criteria that most affect the problem of selecting a multiplatform mobile development tools.
- **Stage 2:** Builds a hierarchical decision model for the problem.
- **Stage 3:** Determines the comparison matrix for each level (level of criteria and sub-criteria) by means of the AHP method to compute the local weight of each criterion and sub-criterion.
- **Stage 4:** Calculates the overall weight by normalizing the local weight.

- **Stage 5:** Uses the TOPSIS technique to assess alternatives, in order to determine the optimal solution easily.
- **Stage 6:** Chooses the highest alternative.

The process of the proposed and adapted integrated methodology to evaluate and select the CP mobile development tools is depicted in Figure 1.

Figure 1. Proposed integrated methodology for CP mobile app development tools selection problem



PROBLEM DEFINITION

Deciding on a CP mobile development framework is not an easy decision to make. There are many variables to consider and there is also the matter of technological philosophy. Thus, it is very important to use clearly defined criteria to make the decision, which best meets the needs of the end-customer, but also which corresponds to the commercial and technological conditions that the developing company can afford. The aim is to provide a framework for making this decision, based on a theoretical and objective perspective rather than subjective opinions. Also, making the contribution of the practice as an area where developers and managers are constantly making up their decisions to improve their business, develop new strategies and gain competitive advantage.

IDENTIFICATION OF CRITERIA AND HIERARCHY

In order to set up a decisional framework for multi-platform mobile development frameworks, we studied the various existing works which are the pioneers in the field namely (Aurelius, 2020) (Rieger & Majchrzak, 2019) (Heitkötter et al., 2013), targeting the choice of criteria that have the most impact on decision-making. According to an in-depth study of this research work, four main criteria have been identified:

- **Business environment perspective**: The Business environment perspective covers aspects about the company and the market the application is targeting (Aurelius, 2020).
- **Development perspective:** The development perspective included all criteria directly related to the development process of the app, e.g. topics such as testing, debugging and development tools (Heitkötter et al., 2013).
- **Application perspective:** The application perspective comprises all criteria that are directly related to the developed application, e.g. Security, Hardware access, System integration, Performance, Robustness, Usage patterns.
- **Infrastructure perspective:** The infrastructure perspective included criteria related to the life cycle of an app, its usage, operation and functionality/functional range (Heitkötter et al., 2013).

Then, the catalogue of criteria was refined by experts in the field. Thus, we have opted for the criteria presented in Table 3 below. In contrast, Table 4 provides a compilation of terminology similarities. For each criterion, we cite related works that have proposed a similar criterion or of the same term. The hierarchical structure of the criteria along with alternative solutions is given in Figure 2.

In the next, decision-makers follow the computational process of weights for chosen criteria by utilizing AHP technique, and then rank the alternatives (frameworks) with the TOPSIS technique. In the first stage of AHP method, we developed a hierarchy model of CP mobile selection based on the criteria, sub-criteria, and alternatives (Stage 2). Such as presented in Figure 2, the top and the lowest levels of the hierarchy signify the global objective (selecting the most CP mobile) and the framework proposed (F1, F2, F3, F4 and F5) respectively. The four main criteria are positioned in the 2nd level and are moreover divided into sub-criteria in the 3rd level.

When the hierarchy has been developed via the AHP method, the next stage is devoted to calculate the weights of elements at each level of the hierarchy. A various comparison matrix of all elements for each level of the hierarchy according to elements of the higher level are performed. The judgments of decision-makers are specified using Saaty scale (Saaty, 2008) as depicted in Table 1.

Table 5 shows the first pairwise comparison matrix for the major criteria proposed by decision-makers. In addition, Table 6 displays the matrix of sub-criteria for the first primary criterion B (i.e. Business environment perspective) as shown by decision-makers (Stage 3).

Other sub-criteria comparison matrices are not included here. However, like criterion B in Table 6 and Table 7 presents all weights vectors derived by pairwise comparisons (Stage 4). Finally, the consistency ratios (CR) of each pairwise comparison judgment matrices are also showed below in each matrix. This latter has a CR of less than 0.1.

In the final phase, the TOPSIS technique is used to rank the alternative frameworks. The global score of each sub-criterion presented in level two (Table 7) can be employed as an input in the TOPSIS technique. Then, the decision-makers are requested to assess the frameworks with respect to each sub-criterion (Table 8) by using the scale in Table 1. The next, in the TOPSIS method, the normalizing of the aggregate ratings matrix, using the Equations (3 and 4), as presented in Table 9 and afterward, by using the Equations (5 and 6), we can compute the ideal and negative-ideal solutions (i.e., A^* and $A-$) for the five alternatives frameworks.

The ranking of the list of frameworks are computed by applying the Equations (7, 8 and 9). The Table 10 presents the calculation outcomes and final classification of frameworks.

Table 3. The main and sub-criteria to set up a decisional framework for multi-platform mobile development

Main Criteria	Sub Criteria		Description
Business environment perspective	B1	Agility	• The Agility in human resources management and respect of the budget maintain the chosen framework: • Different aspects of employees; • Development cost of different platforms; • Developers recruitment cost to get the required speed of development; • Platform changing cost if necessary.
	B2	Competence	• Deployment of existing skills already in the company; • Deployment of available skills on the market and recruit developers.
	B3	Maintainability	• A strong technical expertise is required to modify basic code of the application; • Different performed customizations which also have an impact on the maintenance of the application must be managed.
Infrastructure perspective	I1	License	• The license according to the framework published is extremely important for the sort of application to be developed. It allows evaluating if developer can build commercial applications without forgetting the pricing model which also be considered.
	I2	Target platforms	• The selecting of CP framework is based on the versions of the mobile OS, the number and the importance of supported mobile platforms.
	I3	Long-term feasibility	• Technical support to provide. • New features in the mobile applications to offer. • The Support of the development tools used in selected approach allows handling different risks.
Development perspective	D1	Development environment and testing	• Fragmented platforms supported by the application to test; • Testing tools used must be supported.
	D2	User interface design	• Support advanced Graphical User Interface (GUI) features such as Augmented Reality (AR), Virtual Reality (VR) and 3D rending components; • The platform-specific usage principales such as the location of navigation menus, gestures and scrolling must be supported by the applications. The UI should have a native appearance that is consistent with the platform.
	D3	Continuous delivery	• Continuous integration, automated building, testing and deployment should be supported; • Continuous application store integration to be provided.
	D4	Pace of development	• Manipulate the core code modifications without affecting speed of development.
Application perspective	A1	Security	• Develop secure applications supporting access permissions, data encryption mechanisms, secure data transfer protocols and input validation preventing cross-site forgery and code injection.
	A2	Hardware access	• Supply an essential access to device API (Application Programming Interface) and device specific hardware.
	A3	System integration	• Support the required system integration (cloud and back-end) needed in the application; • The back-end system is used by different application to feed it with different data options as protocols for data sharing, serialization, and a variety of data types.
	A4	Performance	• The performance of the application must be accepted by the user in particular the loading time of the application, the speed of views modification, the calculations resulting from the interactions of the user, the exhaustion of battery while running, the load of the CPU and download size.
	A5	Robustness	• Applications should be fault tolerant and incorporate intelligent methods in case specific functionalities are limited or unsupported.
	A6	Usage patterns	• The experience desired by the user must be instantaneous, available after closing the application if it is not saved, locally saved data retrieved from the Internet and integrated into common applications, such as messaging, email and social media.

Consequently, the appropriate alternative is the one with the longest distance to the negative ideal solution and with the shortest distance to the positive ideal solution. The findings of the suggested approach show that the framework 2 is the appropriate alternative with R_i value of 0,782.

Table 4. Related work referring to key criteria or subordinate terms

Criterion		Literature referencing the criterion or related/subordinate terms
B1	Agility	Company profile (Aurelius, 2020)
B2	Competence	(Aurelius, 2020)
B3	Maintainability	When in time (Aurelius, 2020)
I1	License	(Aurelius, 2020) (Rieger & Majchrzak, 2019) (Heitkötter et al., 2013) (Palmieri et al., 2012) (Dhillon & Mahmoud, 2015), (direct) costs (Dhillon & Mahmoud, 2015) (Hudli et al., 2015) (Heitkötter et al., 2013) (Sommer & Krusche, 2013), open-source (Hudli et al., 2015) (Vitols et al., 2013), (Palmieri et al., 2012), availability (El-Kassas et al., 2017)
I2	Target platforms	(Aurelius, 2020) (Rieger & Majchrzak, 2019), Supported platforms (El-Kassas et al., 2017) (Dhillon & Mahmoud, 2015) (Dalmasso et al., 2013) (Sommer & Krusche, 2013) (Heitkötter et al., 2013), mobile platforms (Vilček & Jakopec, 2017), versions (Botella et al., 2016), portability (Sommer & Krusche, 2013), mobile operating systems (Palmieri et al., 2012)
I3	Long-term feasibility	(Aurelius, 2020) (Rieger & Majchrzak, 2019) (Heitkötter et al., 2013), popularity (Mohamed & Abdelmounaïm, 2017), count of updates (Vitols et al., 2013), community (Vitols et al., 2013)
D1	Development environment and testing	(Aurelius, 2020), Development environment (Rieger & Majchrzak, 2019) (Latif et al., 2016) (Heitkötter et al., 2013) (Ribeiro & da Silva, 2012), Testing (Rieger & Majchrzak, 2019), IDE (Que et al., 2016) (Dhillon & Mahmoud, 2015) (Hudli et al., 2015) (Palmieri et al., 2012), tool restrictions (Sommer & Krusche, 2013), dependencies (Sommer & Krusche, 2013), testing (Umuhoza & Brambilla, 2016) (Sommer & Krusche, 2013), debugging (Que et al., 2016) (Botella et al., 2016) (Dhillon & Mahmoud, 2015) (Sommer & Krusche, 2013) (Ohrt & Turau, 2012), simulator (Latif et al., 2016), emulator (Hudli et al., 2015) (Ohrt & Turau, 2012), test framework (Hudli et al., 2015)
D2	User interface design	(Aurelius, 2020) (Rieger & Majchrzak, 2019), GUI design(er) (Heitkötter et al., 2013) (Ohrt & Turau, 2012), graphical tool for GUI (Mohamed & Abdelmounaïm, 2017), UI design assistant (Botella et al., 2016), no-code/low-code support (Hudli et al., 2015), customizability (Sommer & Krusche, 2013)
D3	Continuous delivery	(Aurelius, 2020) (Rieger & Majchrzak, 2019), Building time (Ebone et al., 2018), build service availability (Dhillon & Mahmoud, 2015), build support (Hudli et al., 2015), simplified/automatic builds (Sommer & Krusche, 2013), compile without SDK (Ohrt & Turau, 2012), instant update (Charkaoui et al., 2014), upgrade (Que et al., 2016), updates (Hudli et al., 2015)
D4	Pace of development	(Aurelius, 2020), (Rieger & Majchrzak, 2019), Development rate (Mohamed & Abdelmounaïm, 2017), speed of development (Heitkötter et al., 2013), developing time (Botella et al., 2016), time to market (Mohamed & Abdelmounaïm, 2017), budget (Mohamed & Abdelmounaïm, 2017), complexity of development (Vilček & Jakopec, 2017), easiness of development (Ahti et al., 2016)
A1	Security	(Rieger & Majchrzak, 2019) (Mohamed & Abdelmounaïm, 2017) (Latif et al., 2016) (Dalmasso et al., 2013), secure storage access, code obfuscation (Dhillon & Mahmoud, 2015), security vulnerabilities, encrypted local storage (Hudli et al., 2015)
A2	Hardware access	(Rieger & Majchrzak, 2019), (Xanthopoulos & Xinogalos, 2013), device features (Latif et al., 2016), device API (Charkaoui et al., 2014), device resource support (Hudli et al., 2015), sensor data capture (Dhillon & Mahmoud, 2015), built-in features (Dalmasso et al., 2013), hardware sensors (Sommer & Krusche, 2013), mobile device functions (Vitols et al., 2013), platform-specific features (Heitkötter et al., 2013), APIs (Palmieri et al., 2012), accelerometer (Ciman & Gaggi, 2017) (Dhillon & Mahmoud, 2015) (Vitols et al., 2013), (Palmieri et al., 2012), (Ribeiro & da Silva, 2012), compass (Ciman & Gaggi, 2017) (Dhillon & Mahmoud, 2015), (Palmieri et al., 2012), proximity (Ciman & Gaggi, 2017), (Dhillon & Mahmoud, 2015), GPS (Ciman & Gaggi, 2017) (Que et al., 2016) (Dhillon & Mahmoud, 2015), (Ribeiro & da Silva, 2012), geolocation (Sommer & Krusche, 2013) (Vitols et al., 2013) (Palmieri et al., 2012), camera (Ciman & Gaggi, 2017) (Que et al., 2016) (Dhillon & Mahmoud, 2015) (Vitols et al., 2013), (Palmieri et al., 2012), (Ribeiro & da Silva, 2012), audio record (Ciman & Gaggi, 2017), microphone (Dhillon & Mahmoud, 2015), Bluetooth (Dhillon & Mahmoud, 2015) (Ohrt & Turau, 2012) (Palmieri et al., 2012), accelerator (Que et al., 2016), GPU acceleration (Dhillon & Mahmoud, 2015), light (Ciman & Gaggi, 2017), notification light activation (Dhillon & Mahmoud, 2015), noise cancelation microphone (Dhillon & Mahmoud, 2015), NFC (Dhillon & Mahmoud, 2015) (Palmieri et al., 2012), gyroscope (Dhillon & Mahmoud, 2015), barometer (Dhillon & Mahmoud, 2015), Wi-Fi positioning (Dhillon & Mahmoud, 2015), cellular positioning (Dhillon & Mahmoud, 2015), network (Sommer & Krusche, 2013) (Vitols et al., 2013), low-level networking (Dhillon & Mahmoud, 2015), connection (Palmieri et al., 2012), (hardware) buttons (Sommer & Krusche, 2013), device (information) (Palmieri et al., 2012)
A3	System integration	(Rieger & Majchrzak, 2019), Social APIs, Cloud APIs (Dhillon & Mahmoud, 2015), backend communication (Dalmasso et al., 2013), corporate identity (Sommer & Krusche, 2013)

continued on following page

Table 4. Continued

Criterion		Literature referencing the criterion or related/subordinate terms
A4	Performance	(Rieger & Majchrzak, 2019) (Sommer & Krusche, 2013), execution time (Biørn-Hansen et al., 2019), duration (Corbalan et al., 2018) (Delia et al., 2017), energy/power consumption (Corbalan et al., 2018) (Ciman & Gaggi, 2017), (Latif et al., 2016), (Dalmasso et al., 2013), app size (Jia et al., 2018) (Ahti et al., 2016), (Ohrt & Turau, 2012), size of installation (Biørn-Hansen et al., 2019), (Sommer & Krusche, 2013), CPU (load) (Corbalan et al., 2018), (Latif et al., 2016), CPU occupancy ratio (Que et al., 2016), RAM/memory usage (Jia et al., 2018) (Ohrt & Turau, 2012) (Ahti et al., 2016), memory occupancy (Que et al., 2016), application/activity launch time (Biørn-Hansen et al., 2020) (Ohrt & Turau, 2012), rendering time (Jia et al., 2018) (Biørn-Hansen et al., 2020), start-up consuming time (Que et al., 2016), app starting time (Ahti et al., 2016), installation consuming time (Que et al., 2016), battery temperature (Que et al., 2016), network flow (Que et al., 2016), resources consumption (Latif et al., 2016), application speed (Heitkötter et al., 2013)
A5	Robustness	(Biørn-Hansen et al., 2020), stability, reliability (Sommer & Krusche, 2013)
A6	Usage patterns	(Biørn-Hansen et al., 2020), User experience conventions (Ohrt & Turau, 2012), screen rotation (Dhillon & Mahmoud, 2015) (Palmieri et al., 2012), device orientation (Ciman & Gaggi, 2017), accessibility features (Ohrt & Turau, 2012), frequency of use (Mohamed & Abdelmounaïm, 2017), offline mode (Mohamed & Abdelmounaïm, 2017) (Charkaoui et al., 2014)

Figure 2. Hierarchical decomposition of criteria in Cross-Platform mobile app development

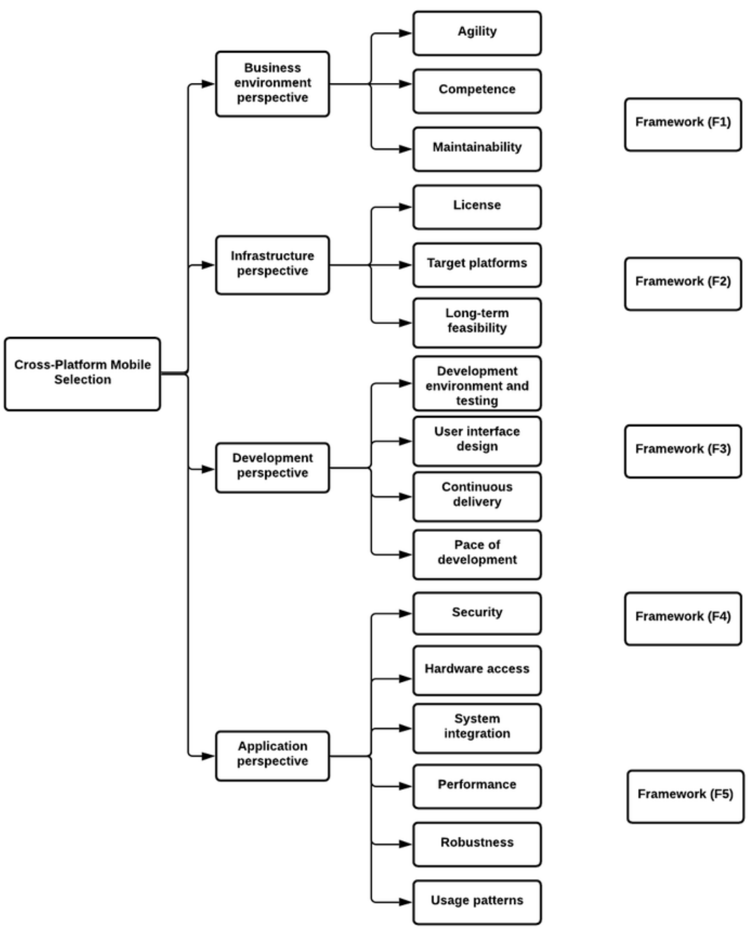


Table 5. The comparison matrix of main criteria

	B	I	D	A	Weight
B	1,00	0,20	0,33	5,00	0,145
I	5,00	1,00	3,00	7,00	0,548
D	3,00	0,33	1,00	5,00	0,255
A	0,20	0,14	0,20	1,00	0,051
CR:					0,09

Table 6. The comparison matrix of sub-criteria according to criteria B

	B1	B2	B3	Weight
B1	1,00	3,00	5,00	0,634
B2	0,33	1,00	3,00	0,260
B3	0,20	0,33	1,00	0,106
CR:				0,029

Table 7. Weightings of normalized sub-criteria

Criteria	Level One	Sub-criteria	Level Two
B	0,145	B1	0,092
		B2	0,038
		B3	0,015
I	0,548	I1	0,045
		I2	0,106
		I3	0,397
D	0,255	D1	0,067
		D2	0,014
		D3	0,143
		D4	0,031
A	0,051	A1	0,005
		A2	0,008
		A3	0,012
		A4	0,021
		A5	0,003
		A6	0,002

SENSITIVITY ANALYSIS

The both-stage AHP and TOPSIS methodologies suggested in this article are subjected to a sensitivity analysis. For this purpose, the criteria weights calculated by AHP technique are swapped among two

Table 8 Input values of the TOPSIS analysis

Criteria	F1	F2	F3	F4	F5		A*	A-
B1	0,03025	0,0605	0,03025	0,04537	0,03025	-	0,03025	0,0605
B2	0,01596	0,02661	0,01064	0,01596	0,01064	-	0,01064	0,02661
B3	0,0075	0,0075	0,006	0,0075	0,0045	-	0,0045	0,0075
I1	0,006	0,031	0,019	0,025	0,006	-	0,006	0,031
I2	0,03	0,07	0,04	0,04	0,04	+	0,07	0,03
I3	0,2	0,3	0,1	0,1	0,1	+	0,3	0,1
D1	0,034	0,027	0,027	0,027	0,034	+	0,034	0,027
D2	0,006	0,007	0,007	0,006	0,006	+	0,007	0,006
D3	0,087	0,069	0,052	0,052	0,052	+	0,087	0,052
D4	0,013	0,02	0,01	0,013	0,01	-	0,01	0,02
A1	0,003	0,002	0,002	0,002	0,002	+	0,003	0,002
A2	0,0039	0,0039	0,003	0,0039	0,003	+	0,0039	0,003
A3	0,007	0,007	0,005	0,005	0,003	+	0,007	0,003
A4	0,012	0,01	0,008	0,008	0,008	+	0,012	0,008
A5	0,002	0,001	0,001	0,001	0,00099	+	0,00166	0,00099
A6	0,001	0,001	0,0008	0,001	0,0008	+	0,001	0,0008

Table 9. The weighted normalized decision matrix

Criteria		Weights	F1	F2	F3	F4	F5
B1	Agility	0,092	2	4	2	3	2
B2	Competence	0,038	3	5	2	3	2
B3	Maintainability	0,015	5	5	4	5	3
I1	License	0,045	1	5	3	4	1
I2	Target platforms	0,106	2	5	3	3	3
I3	Long-term feasibility	0,397	4	5	2	2	2
D1	Development environment and testing	0,067	5	4	4	4	5
D2	User interface design	0,014	5	6	6	5	5
D3	Continuous delivery	0,143	5	4	3	3	3
D4	Pace of development	0,031	4	6	3	4	3
A1	Security	0,005	3	2	2	2	2
A2	Hardware access	0,008	4	4	3	4	3
A3	System integration	0,012	4	4	3	3	2
A4	Performance	0,021	6	5	4	4	4
A5	Robustness	0,003	5	4	4	4	3
A6	Usage patterns	0,002	5	5	4	5	4

Table 10. The final assessment and ranking of alternatives frameworks

	D	D-	R_i	Rank
F1	0,070	0,122	0,637	**2**
F2	0,048	0,170	0,782	**1**
F3	0,170	0,040	0,192	**4**
F4	0,172	0,025	0,128	**5**
F5	0,170	0,046	0,215	**3**

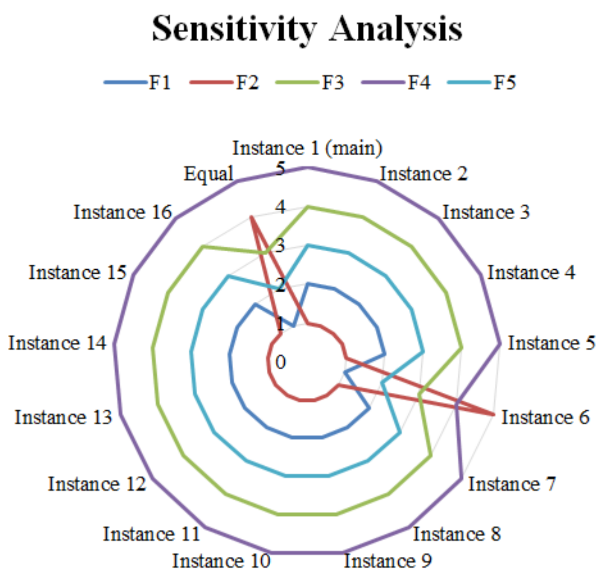
criteria, whereas the others remain unchanged. For every instance, the A*, A– and the R_i are computed to show the new outcomes. As a consequence, seventeen combinations (by adding the identical weight criterion) of the 16 sub-criteria are evaluated. Afterwards, the details of all the instances are summarized and the R_i with ranking of the framework's alternatives are presented together in Table 11 and graphically displayed in Figure 3.

Table 11 and Figure 3 show that the first instance illustrates the main findings of the combined approach (AHP-TOPSIS). Moreover, out of seventeen instances, the framework F2 has the maximum score in almost all instances. Furthermore, the finding of the sensitivity analysis reveals that the ranking of alternatives is different in a significative way regarding to equal score (weights) of the sub-criteria. Consequently, according to the evaluations, the decision-making process is generally unaffected by the weights of the criteria, with F2 coming out on top in almost all cases.

Table 11. Results of sensitivity analysis.

	Alternatives frameworks					Ranking
	F1	F2	F3	F4	F5	
Instance 1 (main)	0,637	0,782	0,192	0,128	0,214	F2 - F1 – F5 – F3 – F4
Instance 2	0,632	0,765	0,213	0,155	0,232	F2 - F1 – F5 – F3 – F4
Instance 3	0,622	0,805	0,144	0,105	0,190	F2 - F1 – F5 – F3 – F4
Instance 4	0,646	0,741	0,181	0,121	0,257	F2 - F1 – F5 – F3 – F4
Instance 5	0,650	0,770	0,204	0,130	0,225	F2 - F1 – F5 – F3 – F4
Instance 6	0,758	0,306	0,688	0,430	0,696	F1 – F5 – F3 – F4 – F2
Instance 7	0,634	0,797	0,169	0,118	0,198	F2 - F1 – F5 – F3 – F4
Instance 8	0,628	0,822	0,142	0,105	0,173	F2 - F1 – F5 – F3 – F4
Instance 9	0,641	0,745	0,243	0,153	0,259	F2 - F1 – F5 – F3 – F4
Instance 10	0,631	0,783	0,191	0,140	0,213	F2 - F1 – F5 – F3 – F4
Instance 11	0,632	0,806	0,136	0,104	0,171	F2 - F1 – F5 – F3 – F4
Instance 12	0,631	0,822	0,137	0,119	0,172	F2 - F1 – F5 – F3 – F4
Instance 13	0,635	0,823	0,149	0,121	0,171	F2 - F1 – F5 – F3 – F4
Instance 14	0,632	0,817	0,140	0,106	0,173	F2 - F1 – F5 – F3 – F4
Instance 15	0,633	0,818	0,144	0,116	0,171	F2 - F1 – F5 – F3 – F4
Instance 16	0,630	0,823	0,136	0,114	0,171	F2 - F1 – F5 – F3 – F4
Equal	0,653	0,431	0,501	0,372	0,546	F1 – F5 – F3 – F2 – F4

Figure 3. Sensitivity analysis under different criteria weights



CONCLUSION

The diversity of mobile platforms presented on the smartphone market makes the development of mobile applications rather complex and very costly. CP development presents the ideal and efficient solution for this business problem. On the other hand, the selection of the appropriate framework has become one of the most significant topics to start with in a mobile project.

The contribution lies in the application of the approach founded on a combined multi-criteria decision-making process. The proposed approach is made up of both methods: The Analytical Hierarchy Process (AHP) and the Order of Preference by Similarity to Ideal Solution (TOPSIS).

The proposed approach is illustrated with a case study and it has been found to work adequately. Five CP mobile development frameworks are selected to prove how this methodology is used and leads to the selection of a framework that is compliant with the maximization of the underlying methods for all decision makers.

ACKNOWLEDGMENT

The publisher has waived the Open Access Processing fee for this article.

REFERENCES

Ahti, V., Hyrynsalmi, S., & Nevalainen, O. (2016). An evaluation framework for cross-platform mobile app development tools: A case analysis of adobe phonegap framework. *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*, 41–48. doi:10.1145/2983468.2983484

Anastasiadou, K., Gavanas, N., Pitsiava-Latinopoulou, M., & Bekiaris, E. (2021). Infrastructure Planning for Autonomous Electric Vehicles, Integrating Safety and Sustainability Aspects: A Multi-Criteria Analysis Approach. *Energies*, *14*(17), 5269. doi:10.3390/en14175269

Aurelius, M. (2020). *Mobile Application Development: Framework with key criteria for choosing native or cross-platform application development*. Academic Press.

Biørn-Hansen, A., Grønli, T.-M., & Ghinea, G. (2019). Animations in cross-platform mobile applications: An evaluation of tools, metrics and performance. *Sensors (Basel)*, *19*(9), 2081. doi:10.3390/s19092081 PMID:31060290

Biørn-Hansen, A., Rieger, C., Grønli, T.-M., Majchrzak, T. A., & Ghinea, G. (2020). An empirical investigation of performance overhead in cross-platform mobile development frameworks. *Empirical Software Engineering*, *25*(4), 2997–3040. doi:10.1007/s10664-020-09827-6

Botella, F., Escribano, P., & Peñalver, A. (2016). Selecting the best mobile framework for developing web and hybrid mobile apps. *Proceedings of the XVII International Conference on Human Computer Interaction*, 1–4. doi:10.1145/2998626.2998648

Broniewicz, E., & Ogrodnik, K. (2020). Multi-criteria analysis of transport infrastructure projects. *Transportation Research Part D, Transport and Environment*, *83*, 102351. doi:10.1016/j.trd.2020.102351

Charkaoui, S., Adraoui, Z., & Benlahmar, E. H. (2014). Cross-platform mobile development approaches. *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, 188–191. doi:10.1109/CIST.2014.7016616

Ciman, M., & Gaggi, O. (2017). An empirical analysis of energy consumption of cross-platform frameworks for mobile development. *Pervasive and Mobile Computing*, *39*, 214–230. doi:10.1016/j.pmcj.2016.10.004

Corbalan, L., Fernandez, J., Cuitiño, A., Delia, L., Cáseres, G., Thomas, P., & Pesado, P. (2018). Development frameworks for mobile devices: A comparative study about energy consumption. *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*, 191–201. doi:10.1145/3197231.3197242

Dağdeviren, M., Yavuz, S., & Kılınç, N. (2009). Weapon selection using the AHP and TOPSIS methods under fuzzy environment. *Expert Systems with Applications*, *36*(4), 8143–8151. doi:10.1016/j.eswa.2008.10.016

Dalmasso, I., Datta, S. K., Bonnet, C., & Nikaein, N. (2013). Survey, comparison and evaluation of cross platform mobile application development tools. *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 323–328. doi:10.1109/IWCMC.2013.6583580

Delia, L., Galdamez, N., Corbalan, L., Pesado, P., & Thomas, P. (2017). Approaches to mobile application development: Comparative performance analysis. *2017 Computing Conference*, 652–659. doi:10.1109/SAI.2017.8252165

Dhillon, S., & Mahmoud, Q. H. (2015). An evaluation framework for cross-platform mobile application development tools. *Software, Practice & Experience*, *45*(10), 1331–1357. doi:10.1002/spe.2286

Ding, J. F. (2012). Using Fuzzy MCDM Model to Select Middle Managers for Global Shipping Carrier-based Logistics Service Providers. *WSEAS Transactions on Systems, 11*(3).

Ebone, A., Tan, Y., & Jia, X. (2018). A performance evaluation of cross-platform mobile application development approaches. *2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 92–93.

El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. M. (2017). Taxonomy of Cross-Platform Mobile Applications Development Approaches. *Ain Shams Engineering Journal*, *8*(2), 163–190. doi:10.1016/j.asej.2015.08.004

Emovon, I., & Oghenenyerovwho, O. S. (2020). Application of MCDM method in material selection for optimal design: A review. *Results in Materials*, *7*, 100115. doi:10.1016/j.rinma.2020.100115

Hanine, M., Boutkhoum, O., Tikniouine, A., & Agouti, T. (2016). Application of an integrated multi-criteria decision making AHP-TOPSIS methodology for ETL software selection. *SpringerPlus*, *5*(1), 1–17. doi:10.1186/s40064-016-1888-z PMID:27006872

Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2013). Evaluating Cross-Platform Development Approaches for Mobile Applications. In J. Cordeiro & K.-H. Krempels (Eds.), *Web Information Systems and Technologies* (pp. 120–138). Springer. doi:10.1007/978-3-642-36608-6_8

Hudli, A., Hudli, S., & Hudli, R. (2015). An evaluation framework for selection of mobile app development platform. *Proceedings of the 3rd International Workshop on Mobile Development Lifecycle*, 13–16. doi:10.1145/2846661.2846678

Huiqun, H., & Guang, S. (2012). ERP software selection using the rough set and TPOSIS methods under fuzzy environment. *Advances in Information Sciences and Service Sciences*, *4*(3), 111–118. doi:10.4156/aiss.vol4.issue3.15

Hwang, C.-L., & Yoon, K. (1981). Methods for multiple attribute decision making. In *Multiple attribute decision making* (pp. 58–191). Springer. doi:10.1007/978-3-642-48318-9_3

Jadhav, A. S., & Sonar, R. M. (2011). Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach. *Journal of Systems and Software*, *84*(8), 1394–1407. doi:10.1016/j.jss.2011.03.034

James, A. T., Vaidya, D., Sodawala, M., & Verma, S. (2021). Selection of bus chassis for large fleet operators in India: An AHP-TOPSIS approach. *Expert Systems with Applications*, *186*, 115760. doi:10.1016/j.eswa.2021.115760

Jia, X., Ebone, A., & Tan, Y. (2018). A performance evaluation of cross-platform mobile application development approaches. *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*, 92–93. doi:10.1145/3197231.3197252

Kabir, G., Ahsan, M., & Hasin, A. (2012). Framework for benchmarking online retailing performance using fuzzy AHP and TOPSIS method. *International Journal of Industrial Engineering Computations*, *3*(4), 561–576. doi:10.5267/j.ijiec.2012.03.003

Khachouch, M. K., Korchi, A., Lakhrissi, Y., & Moumen, A. (2020). Framework Choice Criteria for Mobile Application Development. *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 1–5. doi:10.1109/ICECCE49384.2020.9179434

Lamhaddab, K., Lachgar, M., & Elbaamrani, K. (2019). Porting mobile apps from iOS to android: A practical experience. *Mobile Information Systems*.

Latif, M., Lakhrissi, Y., Nfaoui, E. H., & Es-Sbai, N. (2016). Cross platform approach for mobile application development: A survey. *2016 International Conference on Information Technology for Organizations Development (IT4OD)*, 1–5. doi:10.1109/IT4OD.2016.7479278

Lengacher, D., & Cammarata, C. (2012). A two-phase data envelopment analysis model for portfolio selection. *Advances in Decision Sciences*, *2012*, 2012. doi:10.1155/2012/869128

Liberatore, M. J., & Nydick, R. L. (1997). Group decision making in higher education using the analytic hierarchy process. *Research in Higher Education*, *38*(5), 593–614. doi:10.1023/A:1024948630255

Lin, M.-C., Wang, C.-C., Chen, M.-S., & Chang, C. A. (2008). Using AHP and TOPSIS approaches in customer-driven product design process. *Computers in Industry*, *59*(1), 17–31. doi:10.1016/j.compind.2007.05.013

Marzouk, M., & Sabbah, M. (2021). AHP-TOPSIS social sustainability approach for selecting supplier in construction supply chain. *Cleaner Environmental Systems*, *2*, 100034. doi:10.1016/j.cesys.2021.100034

Mohamed, L., & Abdelmounaïm, A. (2017). Decision Framework for Mobile Development Methods. *International Journal of Advanced Computer Science and Applications*, *8*(2). Advance online publication. doi:10.14569/IJACSA.2017.080215

Mohammed, H. J., Mat Kasim, M., & Mohd Shaharanee, I. N. (2018). Evaluation of E-Learning Approaches Using AHPTOPSIS Technique. *Journal of Telecommunication, Electronic and Computer Engineering, 10*, 7–10.

Myers, J. H., & Alpert, M. I. (1968). Determinant Buying Attitudes: Meaning and Measurement. *Journal of Marketing*, *32*(4_part_1), 13–20. doi:10.1177/002224296803200404

Nawrocki, P., Wrona, K., Marczak, M., & Sniezynski, B. (2021). A Comparison of Native and Cross-Platform Frameworks for Mobile Applications. *Computer*, *54*(3), 18–27. doi:10.1109/MC.2020.2983893

Nunkesser, R. (2018). Beyond web/native/hybrid: A new taxonomy for mobile app development. *2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 214–218.

Ohrt, J., & Turau, V. (2012). Cross-Platform Development Tools for Smartphone Applications. *Computer*, *45*(9), 72–79. doi:10.1109/MC.2012.121

Palmieri, M., Singh, I., & Cicchetti, A. (2012). Comparison of cross-platform mobile development tools. *2012 16th International Conference on Intelligence in Next Generation Networks*, 179–186. doi:10.1109/ICIN.2012.6376023

Panda, B. N., Biswal, B. B., & Deepak, B. (2014). *Integrated AHP and fuzzy TOPSIS approach for the selection of a rapid prototyping process under multi-criteria perspective*. Academic Press.

Que, P., Guo, X., & Zhu, M. (2016). A comprehensive comparison between hybrid and native app paradigms. *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*, 611–614.

Ribeiro, A., & da Silva, A. R. (2012). Survey on cross-platforms and languages for mobile apps. *2012 Eighth International Conference on the Quality of Information and Communications Technology*, 255–260. doi:10.1109/QUATIC.2012.56

Rieger, C., & Majchrzak, T. A. (2019). Towards the definitive evaluation framework for cross-platform app development approaches. *Journal of Systems and Software*, *153*, 175–199. doi:10.1016/j.jss.2019.04.001

Russo, R. F. S. M., & Camanho, R. (2015). Criteria in AHP: A systematic review of literature. *Procedia Computer Science*, *55*, 1123–1132. doi:10.1016/j.procs.2015.07.081

Saaty, T. L. (1990). How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, *48*(1), 9–26. doi:10.1016/0377-2217(90)90057-I

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, *1*(1), 83–98. doi:10.1504/IJSSCI.2008.017590

Sedghiyan, D., Ashouri, A., Maftouni, N., Xiong, Q., Rezaee, E., & Sadeghi, S. (2021). Prioritization of renewable energy resources in five climate zones in Iran using AHP, hybrid AHP-TOPSIS and AHP-SAW methods. *Sustainable Energy Technologies and Assessments*, *44*, 101045. doi:10.1016/j.seta.2021.101045

Sommer, A., & Krusche, S. (2013). Evaluation of cross-platform frameworks for mobile applications. *Software Engineering*.

Tsaur, R.-C. (2011). Decision risk analysis for an interval TOPSIS method. *Applied Mathematics and Computation*, *218*(8), 4295–4304. doi:10.1016/j.amc.2011.10.001

Umuhoza, E., & Brambilla, M. (2016). Model driven development approaches for mobile applications: A survey. *International Conference on Mobile Web and Information Systems*, 93–107. doi:10.1007/978-3-319-44215-0_8

Vilček, T., & Jakopec, T. (2017). Comparative analysis of tools for development of native and hybrid mobile applications. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1516–1521.

Vitols, G., Smits, I., & Bogdanov, O. (2013). Cross-platform Solution for Development of Mobile Applications. *ICEIS*, (2), 273–277.

Xanthopoulos, S., & Xinogalos, S. (2013). A comparative analysis of cross-platform development approaches for mobile applications. *Proceedings of the 6th Balkan Conference in Informatics on - BCI '13*, 213. doi:10.1145/2490257.2490292

Yoo, K. E., & Choi, Y. C. (2006). Analytic hierarchy process approach for identifying relative importance of factors to improve passenger security checks at airports. *Journal of Air Transport Management*, *12*(3), 135–142. doi:10.1016/j.jairtraman.2005.11.006

Zaidan, A. A., Zaidan, B. B., Al-Haiqi, A., Kiah, M. L. M., Hussain, M., & Abdulnabi, M. (2015). Evaluation and selection of open-source EMR software packages based on integrated AHP and TOPSIS. *Journal of Biomedical Informatics*, *53*, 390–404. doi:10.1016/j.jbi.2014.11.012 PMID:25483886

Zaragoza, M. G., Kim, H.-K., & Han, D. S. (2016). Mobile Application Development, Approaches, Advancement and Process. *International Journal of Software Engineering and Its Applications*, *10*(10), 79–88. doi:10.14257/ijseia.2016.10.10.09

Zlaugotne, B., Zihare, L., Balode, L., Kalnbalkite, A., Khabdullin, A., & Blumberga, D. (2020). Multi-criteria decision analysis methods comparison. *Scientific Journal of Riga Technical University. Environmental and Climate Technologies*, *24*(1), 454–471. doi:10.2478/rtuect-2020-0028

Mohamed Lachgar, Ph.D., in Computer Science at the Cadi Ayyad University in 2017, is a professor in Computer Science at the National School of Applied Sciences, Chouaib Doukkali University, El Jadida, Morocco. His research interests are in the areas of automation tools development in embedded software, software modeling & design, metamodel design, model transformation, model verification & validation method, machine learning and deep learning.

Hanine Mohamed is an Assistant Professor in Computer Science at the National School of Applied Sciences, University of Chouaib Doukkali, EL Jadida (Morocco). He received a Diploma degree (M.Sc) in computer science from the University of Cadi Ayyad (Morocco), in 2010. In 2017, he obtained the Ph.D. degree in computer science from the University of Cadi Ayyad (Morocco). Then in 2018 he joined the Department of Telecommunications, Networks and Computer Science at the National School of Applied Sciences, teaching engineering students in the area of Big Data and Business Intelligence. His research interests include Business Intelligence, Big data, and Decision Support Systems.

Benouda Hanane is an Assistant Professor in Computer Science at the Higher School of Technology of Khenifra, Sultan Moulay Sliman University, Morocco. She got her Master Degree in Computer Engineering from Mohammed First University of Oujda, Morocco in 2013. In 2018, she obtained the Ph.D. degree in computer science from Mohammed First University of Oujda. Her main research interests include software engineering, mobile applications, software modeling, model transformation, reverse engineering, and Artificial Intelligence.

Younes Ommane is postdoctoral researcher in Machine learning and data analysis at MSDA (Modeling Simulation and Data Analysis) laboratory of Mohamed VI Polytechnic University (UM6P), Kingdom of Morocco. He holds a PhD in Statistics from Cadi Ayyad University, Kingdom of Morocco. He received a Masters degree in Computer science, Statistics and Mathematics Applied to Production management from Université de Toulouse Jean Jaurés, France. He received a bachelor's degree in Fundamental and applied mathematics from Cadi Ayyad University, Kingdom of Morocco. His research interests are time series forecasting using classical and ML techniques, Bayesian statistical decision theory, nature-inspired optimization algorithms and mathematical foundations of ML methods.