

An aggregate indicator for mobile application quality assessment

Paul POCATILU*, Ion IVAN, Alin ZAMFIROIU, Catalin BOJA

Department of Economic Informatics and Cybernetics, Faculty of Economic Cybernetics, Statistics, and Informatics,
Bucharest University of Economic Studies, Bucharest, Romania

Received: 16.04.2013

Accepted/Published Online: 26.06.2013

Printed: 10.06.2015

Abstract: The mobile applications environment has a multitude of particularities generated by the complex system of devices, hardware, and software profiles and categories of users. The mobile industry has also evolved rapidly in the last 5 years from single-core mobile platforms to multiple-core mobile application platforms, able to compete in terms of user services with notebooks. New types of devices, like tablets, or shifts in human-computer interface paradigms from button click to screen touch, have opened new perspectives both for users and software developers. This requires an adaptable approach for performance analysis in terms of total quality management for software projects. In this context, quality is analyzed from the viewpoints of developers, of users and, respectively, of one who aims to recover an investment. This paper proposes a mobile application quality estimation model that requires the identification of features such as mode of interaction, command speeds, interaction time, drivability, volume of provided information, error management, self-healing, integrability, data security, transaction security, coefficients determining importance and building an aggregate indicator, and which properties should be highlighted for ensuring usability in a practical environment. The indicator is integrated in a quality metric used to assess mobile applications' quality. Emphasizing the usefulness of the indicator is done on a representative set of mobile computing applications. The paper proposes a set of indicators normalized on the $[0; 1]$ interval used to measure the application quality level.

Key words: Total quality management, mobile application, aggregate indicator, software metric

1. Introduction

In software engineering literature [1–3], software quality and software quality management are treated separately and the important role that they play in successfully completing software applications projects is highlighted.

A special category of software applications is that of mobile applications. Mobile devices and mobile applications are seeing continuous development and their quality management processes are influenced by their particular specifications. It is considered that the implementation of specific total quality management (TQM) methods and techniques within software companies will help developers meet the objectives related to the continuous quality improvement of the processes and results [4,5].

This paper is organized into sections that analyze different aspects of the proposed problem as follows.

Section 2, on the particularities of mobile applications, describes characteristics of mobile applications based on the aspects that have a significant effect on their quality.

In Section 3, on mobile applications' quality characteristics, specific features of the mobile applications are identified and described. Each of them has an associated metric used to measure its level.

*Correspondence: ppaul@ase.ro

Section 4, on the structure of the aggregate indicator (IA) associated with TQM processes, analyzes how specific mobile application indicators are aggregated into a global indicator.

In Section 5, on aggregate indicator properties, the characteristics of this aggregate indicator are analyzed in order to be validated.

Section 6, on the integrating of the IA within the TQM metric for mobile applications, shows how the IA reflects all the specific elements of TQM.

Section 7 describes and analyzes the empirical results obtained in practice by assessment of a real mobile application by experts in the field. The results of the analysis are presented by means of integrating the mobile application's quality characteristics scores given by experts.

The development of mobile applications requires monitoring the high quality of the resources and technologies used in the development process. These are specific to the development cycle and directly affect the risk of generating increased costs, risks that can be strictly controlled through the reduction of the repetitive steps needed for corrections. Defining metrics and measuring the components that are assimilated into the final product over the development cycle should be completed by addressing global resources-processes-product quality, which is possible only in the context of defining an IA of quality.

The paper ends with the presentation of conclusions and future work, emphasizing the elements that extend the use of the model into other software fields and identifying paths for future research.

2. Particularities of mobile applications

A mobile application is a set of instructions processed by a mobile device in order to solve a problem. Below is a list of a series of features for mobile applications.

- Each application is an investment, and it is produced by a team with the objective of recovering the invested resources through the utility and the quality of the services the application provides. The investment is covered by: **1.** asking a purchase price for the rights to use the application, **2.** releasing the application for free while add-ons, upgrades, and tools are priced, or **3.** allowing third-party investors to advertise on a paid basis within the application. It is common that online app stores provide a wide range of free applications that, once installed, run advertisements for different products, which helps the investors recover the amount invested, as in the freebie marketing business model.
- There is a target group that is rigorously defined, which is the base for setting benchmarks of efficiency related to the investment in the mobile application. Before the investment is made, on the basis of surveys, the needs, wishes, or expectations of the target group from the software applications are identified; the next step is the drafting of the specifications for application to meet all the above.
- The application should be able to work independently 24/7; the high portability of these applications should allow users to use them regardless of time or space, and the developer must not interfere or make changes when users use the mobile application. Any error or problem in the application has catastrophic long-term effects.
- Applications should not require prior training, considering that most users are accessing software applications without previously engaging in study involving the use and operation of these applications; such software should be designed so as to contain natural interface elements and be user-friendly, even for users who are using the application for the first time.

- Applications usually contain elements of continuity in terms of user experience and user interface; common vocabulary items are used, fields are positioned relatively to the same visual marks, and the used field types and the presentations of messages are similar in the applications frequently used by most users, thereby ensuring user-friendliness.
- It should allocate resources for fast recovery in the case of errors generated by the application or clients; usually errors occur regardless of the users' experience or the quality of the application because the mobile platform state is influenced by many factors, many of them external to the application. However, in the case of the latter, there must be availability for rapid restoration to the error-free state.
- It should have the ability to integrate existing components such as maps with GPS localization for localization-based applications [6] that indicate restaurants, museums, or other services within a city; a calendar allowing selection of the day, month, and year for applications with date input fields, thus ensuring that data input is made by selection rather than by typing; e-mail availability for sending the result provided by the application or data entry into the application for information received via e-mail; web browsing for searching for certain information or use of web services to access databases stored on the server; the ability to send text messages with solutions provided by application; an address book to select contact information or contact details; an auto-fill facility to fill in the fields for the contact information; a camera to take pictures or visual data entry via QR codes; and a microphone to record sounds.
- It should allow integration into existing applications and application systems such as finding the distance between two selected locations, with different vehicles selected by the user, allocation of tickets for vacation, car rentals, etc.
- Maximum reliability should be ensured considering the fact that the target group is heterogeneous and the behavior of its members is likely to generate vulnerabilities.
- Security of the database and of any kind of transactions must also be ensured.
- It should have the ability for automatic updates, and users should perceive these not as a change in the application process but rather as an addition of new functions/processing modules or as an increase of performance of the existing functions. For instance, in order to pay a phone bill, currently the user provides an IBAN account, amount, invoice number, and customer ID, whereas when making a template payment order, the user only needs to enter the amount and invoice number, the other information being already provided in the context of database integration. When a client who has a bank account is logged into the application and selects the mobile network company using the identification code, a database of network providers is accessed and data are extracted related to the amount, customer code, supplier's IBAN account, invoice number, etc., where the payment order is automatically issued using these data; the client's only input is the change in the amount to be paid, greater if paid in full or smaller if the payment is made in installments.

All these elements drive the development processes and implementation of mobile applications to TQM, the only form that ensures meeting the requirements for fully operational mobile applications.

3. Mobile application quality characteristics

In the context of the developed information society and the significant penetration of mobile applications and mobile devices in many human activities, they can be considered as promising tools to improve the quality of life for all people [7]. Their usability depends strictly upon:

- problem to be solved;
- target group structure;
- frequency of activation and use of application;
- confidence in the application and returned results;
- distinctive advantages, such as eliminating lost time and saving money;
- availability to automatically collect data to establish user profiles in relation to their behavior in the interaction with mobile applications, where based on the custom profile created the application adjusts itself so that it is closer to the user by data entry and results are delivered;
- user satisfaction with regards to solving his/her problem;
- efficiency and correctness of solutions provided for the problem;
- existing shortcuts for quick use of the application;
- elimination of disturbing elements and optimization of displaying only the elements that are deemed important for the user;
- the order in which the information is provided being presented naturally for easy understanding;
- use of terminology and vocabulary familiar to the user; in [8], mobile applications were used to build the vocabulary in a learning environment;
- and user support throughout the use of the application.

Any mobile application is assessed from the following points of view:

- that of the investor, who seeks to minimize the duration of the investment recovery, the risks associated with the extension of this duration, and the risks of premature removal from service;
- that of the developer, who aims for the completion of the development cycle within a reasonable time, with costs that ensure a stimulating level of efficiency;
- that of the client, who wants to obtain fast, low-cost services without erroneous allocation of resources, with minimal learning effort and with an error management that is strictly correlated to his/her individual performance;
- and that of the user, who aims to achieve solutions to his/her problem with minimal learning effort and minimal interaction with the application, and with time availability of the application.

Compared to the other software products, mobile applications assessed from the viewpoint of developers require the consideration of a set of characteristics represented by $MC = \{C_1, C_2, \dots, C_{NC}\}$. The set of characteristics for this case consists of ten features listed below.

The mode of interaction (MI) represents the use of appropriate controls to build the application user interface so that the user will have a natural interaction with it. An investigation of communicative modalities in relation to cars with mobile devices was presented in [9]. The mode of interaction is influenced by the number of displayed controls at a time and their complexity. The measurement is a score received from the user after he/she interacts with the application. The mark, MI_{mark} , is a value in the range $[A_{inf}; A_{sup}]$, where

$A_{\text{inf}} < A_{\text{sup}}$ $A_{\text{inf}}, A_{\text{sup}} \in N$; for this case we have $A_{\text{inf}} = 0$; $A_{\text{sup}} = 10$. For determining the mode of interaction indicator, MII, the following relation is used:

$$MII = \frac{MImark}{10}. \quad (1)$$

The command speed (CS) is the speed at which commands within the application are executed. The factors that influence the CS come from mobile device processing power, available memory, memory management of the application, and the number of variables used in developing algorithms that solve the commands initiated by the user. The measurement of the CS is made by measuring the time needed to complete the commands, with less time being better [10]. In order to scale the CS indicator (CSI) in the interval [0; 1], the following calculation is used:

$$CSI = 1 - \frac{val_c}{\max_{1 \leq i \leq NrM} \{val_i\}}, \quad (2)$$

where val_c is the current measured value and val_i is the measured value for executing the command i .

The interaction time (IT) represents the time spent by the user to solve the entire problem, not only of a single command. The factors affecting the IT are similar to the factors that influence the CS; measurement is performed also by determining the time measured in seconds, as a reference, with less being better. The IT indicator, ITI, is calculated according to:

$$ITI = 1 - \frac{dc}{\max_{1 \leq i \leq NrM} \{d_i\}}, \quad (3)$$

where dc is current time measured and d_i is the time measured at moment i .

Drivability (D) describes the easiness of using the application; the indicator is influenced by the controls used in building the mobile application interface. The interface should be as natural as possible, allowing the user to understand without the need for reading the application's user guides; measurement is done by the user giving a score, $Dmark$, in the range of 1 to 10, 10 signifying that the user manages to use the application without problems even if he/she has not read the documentation or the user guide for that application. For calculating the drivability indicator, DI, the following formula is used:

$$DI = \frac{Dmark}{10}. \quad (4)$$

The volume of provided information (VPI) is the amount of data and information displayed on the screen at a certain moment in time via the mobile device application. In [11], the context data distribution in a mobile environment was analyzed. The volume of distributed data should be reduced according to the rate transfer of Internet services. Considering the set of moments $MV = \{MV_1, MV_2, \dots, MV_{nv}\}$, where MV_i represents the i moment when the volume of provided information in the application is measured, we then have the set $MVIF = \{VPI_1, VPI_2, \dots, VPI_{nv}\}$. VPI_i represents the amount of information measured at the MV_i moment; using the given notations the VPI of the application is defined, which is calculated by the following mathematical relation:

$$VPI = \frac{\sum_{i=1}^{nv} VPI_i}{nv}, \quad VPI \geq 0, \quad (5)$$

where a high value of the indicator is ideal, at the same time observing the user interaction characteristics.

Error management (EM) or fault tolerance is the ability of mobile applications to function even in cases where operational errors appear, which are managed properly for continuous use of the application [12]. The developer has a major influence on the management application, which should, during the development process, provide alternative options for running the application in the event of the occurrence of errors and for their management so as not to influence the results provided by the application. Measuring is done by observing the number of instructions successfully fulfilled from the total number of instructions given by the user. Thus, the error management indicator (EMI) is calculated using the following relation:

$$EMI = \frac{NT}{N_{total}}, \quad (6)$$

where NT is the number of successful instructions and N_{total} is the number of total instructions.

Self-healing (SH) is the capacity of the mobile application to recover modules affected by certain attacks or events that corrupt the integrity and the correctness of the binary code [13,14]. The developer should consider developing the mobile application in modules and, in the case of a corrupted module, only that module/part is to be rebuilt or recovered, not the entire application. The measurement is done by counting the number of existing modules in the application. The self-healing indicator (SHI) is calculated as:

$$SHI = \frac{NRm}{NRf}, \quad (7)$$

where NRm is the number of modules in the application and NRf is the number of functions performed by the application.

Integrability (IN) is the ability of mobile applications to include in their data processing flow existing components already used by users; the indicator describes also how to design the mobile application and the data processing flow to prepare it to become a component in other applications. The integrability indicator (INI) is calculated with the following equation:

$$INI = \frac{\min(nrA, nrAI)}{\max(nrA, nrAI)}, \quad (8)$$

where nrA is the number of mobile applications available on a mobile device and nrAI is the number of mobile applications that can integrate with the current application.

Data security (DS) ensures data privacy in the mobile device. When used by unauthorized persons, the accessing of confidential data on the device may harm the owner, and thus it is necessary to secure access to this information if the mobile device is lost or stolen, [15,16]. The data security indicator (DSI) is given by the number of cases where the application does not use the mobile device resources correctly, thus allowing uncontrolled events to occur per unit time (UT):

$$DSI = 1 - \frac{TNEU}{UT}, \quad (9)$$

where TNEU is the total number of events that are unpredictable and uncontrolled.

Transaction security (TS) ensures the confidentiality of personal data or important applications in transactions with other mobile devices [17], or the security of the server where the database is stored. This

was addressed in [18], where relevant services and security aspects in mobile banking were presented. The transaction security indicator (TSI) is calculated as:

$$TSI = 1 - \frac{TNEU}{NrTr}, \tag{10}$$

where NrTr is the number of transactions performed in the mobile application.

Usability has a very important role in mobile applications (MI, D); an ample analysis of this characteristic is available in [19].

Another important role in the quality of mobile applications is that of energy consumption or the battery life cycle characteristics, as treated in detail in [20,21]. In [22] the energy problem was treated, where an alternative was analyzed using the server’s computing power via cloud computing. In this way, the energy consumption for processing and computing power is decreased at the mobile device level by the server, which has the resources without imposed limits.

In [23], the mode for developing software metrics for mobile and wireless systems was presented.

Initially the set of quality characteristics and attributes that affect the analyzed system is defined, objects to be measured are defined, and the new metric is designed and then analyzed through experiments and tests. In this scope a sample group is formed by contacting NS specialists $SP = \{SP_1, SP_2, \dots, SP_{NS}\}$. A scoring system is defined as a given set $A = \{a_1, a_2, \dots, a_{NC}\}$, with the total amount of the scores ST:

$$ST = \sum_{i=1}^{NC} a_i, \tag{11}$$

where NC is the number of characteristics/number of marks in set A, and a_i is score i from set A.

The specialist group $S = \{S_1, S_2, \dots, S_{NS}\}$ provides unique scores from set A to each characteristic from the MC set. The relation between set A and set MC is 1:1; any element of set MC corresponds to a single element of set A and every element of set A corresponds to a single element from set MC. Thus, the resulting subset of scores given by expert SP_i is:

$$A_{SP_i} = \{as_1^{SP_i}, as_2^{SP_i}, \dots, as_{NC}^{SP_i}\}, i = 1, 2, \dots, NS. \tag{12}$$

To verify compliance with the principle of providing unique results, the scores are calculated as amounts on lines to be equal to ST:

$$\sum_{k=1}^{NC} as_k^{SP_i} = ST; \quad i = 1, 2, \dots, NS, \tag{13}$$

The amounts are calculated in columns, SC_j , representing the amount of received scores by characteristic C_j from all NS specialists:

$$SC_j = \sum_{l=1}^{NS} as_j^{SP_l} \quad j = 1, 2, \dots, NC, \tag{14}$$

After calculating the totals for each characteristic, the sum of totals is calculated for all characteristics:

$$SS = \sum_{j=1}^{NC} SC_j. \tag{15}$$

The summing of total SS is equal to the number of specialists, NS, multiplied by the amount scores, ST:

$$SS = NS \times ST. \quad (16)$$

Weight W_j for characteristic C_j is calculated as follows:

$$W_j = \frac{SC_j}{SS} \quad j = 1, 2, \dots, NC \text{ where } \sum_{j=1}^{NC} W_j = 1. \quad (17)$$

Obtained weights are used in the construction of the IA.

All these indicators enter the structure of quality metrics for mobile applications. It is necessary to refine the software metric by ranking the indicators, introducing new representative indicators or eliminating the indicators that are not representative.

4. Structure of the IA associated with TQM processes

For each element C_i of set MC, a new set of estimation models is defined as MM^{C_i} for each of the characteristics: $MM^{C_i} = \{M_1^{C_i}, M_2^{C_i}, \dots, M_{NMC_i}^{C_i}\}$, $i = 1, 2, \dots, NC$, where NMC_i is the number of models for the C_i characteristic. From MM^{C_i} there is extracted the model considered by developers to be representative for C_i compared with a level of accepted experience. The set of pairs is built:

$$\langle C_i | M_i \rangle \quad i = 1, 2, \dots, NCM_i \in MM^{C_i}, \quad (18)$$

where C_i is characteristic I and M_i is the selected model for characteristic i.

It is important that the model M_i be developed so that that its domain is [0; 1]. Model development has to take into account factors of influence for selected characteristics. Further on, indices for the calculation of each of the characteristics from the MC set are built on the basis of the selected models.

For the IA, the linear analytical expression is as follows:

$$IA = \sum_{i=1}^{NC} W_i \times I_{C_i}, \quad (19)$$

where W_i is the weight of characteristic C_i and I_{C_i} is the indicator associated with characteristic C_i .

The IA has to be representative for the specific application, an aspect that needs to be demonstrated. The representativity of the indicator is built upon:

- using a sufficiently large number of mobile applications;
- estimating the levels of quality features;
- calculating the aggregated indicator using the estimated weights and levels;
- studying the behavior of mobile applications in relevant time intervals;
- measuring the effective level of the quality characteristics;
- calculating the IA using effective weights and levels;

- calculating the difference between the estimated and the effective IA;
- counting the differences that are below the threshold of acceptability;
- and calculating the weight of acceptable differences compared to the total number of mobile applications.

Based on practice and experience, if the ratio is greater than 0.78, the indicator is accepted as being good. If the ratio is greater than 0.92, it is considered very good. In both cases, the indicator is representative.

Figure 1 shows the representativity interval for the IA, in this case between [0.78; 1]. Thus, for this interval, the indicator is representative. If the indicator value is outside the determined range it is not representative for use.

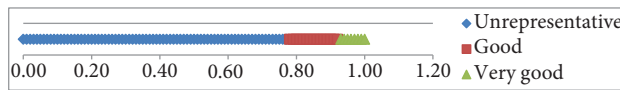


Figure 1. Interval representativity for IA.

4.1. IA properties

The theory of the construction and use of metrics requires the analysis of all the components that are included. Each indicator is analyzed in terms of the degree to which a number of requirements are satisfied. An indicator is sensitive if small variations of the variables are included in the analytical expression. Small variations are obtained, respectively, if large variations of the same variables determine large variations of the indicator.

Let us consider the aggregate indicator $IA^0(X)$ of mobile application X defined by:

$$IA^0(X) = \sum_{i=1}^{NC} W_i \times P_i^0, \tag{20}$$

where P_i^0 is the quality level measure of the C_i characteristic. According to the analytic structure of the IA, this indicator will have values in the (0; 1) interval.

Furthermore, the aggregate indicator $IA^1(X)$ is defined as:

$$IA^1(X) = \sum_{i=1}^{NC} W_i \times P_i^1, \tag{21}$$

with

$$P_k^1 = P_k^0 + \Delta; \quad \text{and} \quad P_i^0 = P_i^1, i = 1, 2, \dots, k - 1, k + 1, \dots, NC, \tag{22}$$

and thus:

$$IA^1(X) = \sum_{i=1}^{NC} W_i \times P_i^1 = \sum_{i=1}^{k-1} W_i \times P_i^0 + W_k \times (P_k^0 + \Delta) + \sum_{i=k+1}^{NC} W_i \times P_i^1 \tag{23}$$

$$IA^1(X) = \sum_{i=1}^{NC} W_i \times P_i^0 + W_k \times \Delta = IA^0(X) + W_k \times \Delta$$

This shows that if Δ is small the influence on the IA is small, and if Δ has a higher value the contribution to the indicator is greater.

The indicator is noncompensatory if for different values of the variables within the analytical expression different values of the indicator are produced. In this case, the initial indicator is considered in the form of:

$$IA^0(X) = \sum_{i=1}^{NC} W_i \times P_i^0, \tag{24}$$

and the indicator with the changed values is:

$$IA^1(X) = \sum_{i=1}^k W_i P_i^1 + W_k (P_k^0 + \Delta_k) + \sum_{i=k+1}^r W_i P_i^1 + W_r (P_r^0 + \Delta_r) + \sum_{i=r+1}^{NC} W_i P_i^1, \tag{25}$$

with

$$P_i^0 = P_i^1 \quad i = 1, 2, \dots, k - 1, k + 1, \dots, r - 1, r + 1, \dots, NC. \tag{26}$$

This results in:

$$IA^1(X) = IA^0(X) + W_k \times \Delta_k + W_r \times \Delta_r, \tag{27}$$

and so:

$$IA^1(X) = IA^0(X), \tag{28}$$

if and only if:

$$W_k \times \Delta_k + W_r \times \Delta_r = 0, \tag{29}$$

which leads to:

$$\frac{W_k}{W_r} = -\frac{\Delta_r}{\Delta_k}. \tag{30}$$

In conclusion, this shows that, because there are different values Δ_k and Δ_r for which $IA^1(X) = IA^0(X)$, the IA is compensatory.

An indicator is considered to be catastrophic if for small variations of the variables of the right part of the analytical expression large variations of the aggregate of the indicator are obtained. Usually, analytical expressions that contain division operators or power or elementary functions such as $exp()$; $log()$; $sin()$; $cos()$; $tg()$; $ctg()$; and $arctg()$ lead to catastrophic indicators. Since the IA does not contain such functions, the index is considered to be noncatastrophic.

In [24,25] it was demonstrated that there is no indicator that simultaneously satisfies the three properties.

The IA is operational if it develops a data acquisition process at a reasonable cost so that its level can be calculated at any time. Since different data acquisition methods have been developed in mobile telephony, mobile applications are designed to be loaded into the backend database behavior metrics for all users, thus allowing instantaneous calculation of aggregate quality indicator level, which means that this indicator is operational.

The indicator is representative if domain segments for which its levels are defined are mapped to qualitative levels as unsatisfactory, satisfactory, good, very good, or acceptable, respectively, or unacceptable. The risk of making an unrealistic qualitative classification must also be within the accepted limits as per the statistical hypothesis testing theory for the null hypothesis.

According to [26], emphasis of representativeness is done similarly with metrics validation.

Considering the mobile computing applications used to estimate the IA, and the effective levels measured during tests, we compare the estimated and the effective levels. After that, the applications are counted for which the difference between the estimated and effective level is below a given threshold.

If the weight in relation to all the applications resulting from the comparison is sufficiently large, it can be agreed that the IA is sufficiently representative.

Mobile application developers should be interested in the efficient management of the IA throughout the execution cycles, since returning to previous stages to make corrections generates costs that affect organizational efficiency.

1. Integrating the IA within the TQM metrics for mobile applications

TQM of mobile applications is characterized by:

- production driven by the particular needs of users, because mobile applications are user-oriented;
- quality defined by the demands and requirements that satisfy the user needs, which is measured by customer satisfaction and the continuous improvement of the software;
- quality, the component that is outlined using the resources and costs involved during the entire existence of the mobile application;
- focus on the qualitative but not quantitative side of the application, as mobile applications are not targeting a huge amount of information but only selecting the necessary useful or vital information;
- ensuring the level of interaction established by other applications;
- open organizational culture;
- development process oriented towards the coverage of all aspects of the application lifecycle, and especially mobile application user experience, maintenance, and reengineering;
- systematic approaches to improve the abilities and experience of designers, programmers, and other staff to obtain high-quality source code and thus better mobile applications;
- and permanent concern of the entire organization to purchase the tools, techniques, and technologies for measuring developers; performance, in this way creating the conditions to obtain mobile applications with execution free of incidents and any other problems.

In this context, TQM objectives are presented in Figure 2. The indicator should reflect all the specific characteristics of TQM.

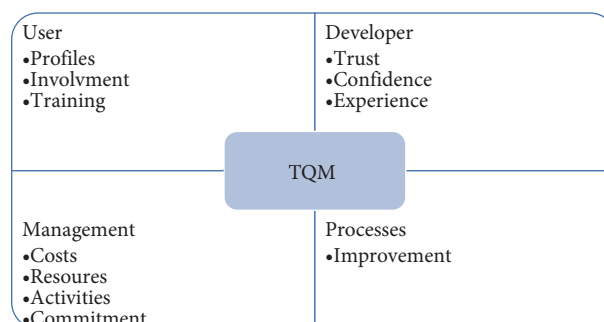


Figure 2. TQM goals.

Let us consider $E_1, E_2, \dots, E_{ntqm}$ as the elements that define TQM. For each E_j there is built an aggregate indicator, IA_j . The composition of an IA that concerns different aspects of the mobile application development process is done by considering the following aspects:

- the system of weights is built as $Q = \{q_1, q_1, \dots, q_{ntqm}\}$ with

$$\sum_{i=1}^{ntqm} q_i = 1 \quad q_i \in [0; 1]; \quad (31)$$

- the aggregated indicators for E_j are measured;
- and the analytical expression is built:

$$IAA = \prod_{i=1}^{ntqm} IA_i(E_i)^{q_i}. \quad (32)$$

The problem of integrability is necessary because an organization that develops mobile applications has the appropriate experience in estimating, planning, and measuring the quality of its products both as finished products and intermediate products as the outputs of the production process.

If the IA is designed outside a process of integrability, then at the moment of transition towards production in the context of TQM, everything should start from zero, thus losing a lot of the time and capital invested.

If it there is an option to have a slow transition to production, then sufficient elements must be assured that describe resources, activities, inputs and outputs, and the behavior of developers, analysts, testers, and users, and especially the behavior of the product itself, assuming correct use.

It is required only to establish the subsets that allow the calculation of the IA based on the categories of elements $E_1, E_2, \dots, E_{ntqm}$. Once the performance data required to establish weights $q_1, q_2, \dots, q_{ntqm}$ are collected, the IA is operational.

The IA is sensitive, compensatory, operational, and noncatastrophic and its representativeness should be confirmed by experiments over time.

5. Experimental results and discussions

In order to test and validate the IA, the following activities were conducted:

- development of a mobile application;
- consultation with specialists in order to determine the weights for each quality characteristic;
- application of the model to the developed mobile application using the calculated weights.

The next sections discuss these activities in detail.

5.1. Application development

The name of the developed prototype application is QMS. For its development applications commonly used by users were analyzed. In this way, we adopt common application aspects to ensure the continuity of elements used in applications. These were analyzed based on the following criteria:

- the most commonly used platform;
- inputs in the application;
- the way the input information is entered;
- application outputs;
- the presentation of the application outputs or results.

The application was developed for the Android platform. The database used is an SQL Server and the interaction between the application and the database is done through web services developed in the C# programming language using Microsoft Visual Studio 2010.

As per the above, access to information in the database is possible only through web services. The web service is regarded as intermediate level, a middleware for improving security and confidentiality of the database, because access is done systematically and only necessary information is made available to end users. An example of the security architecture for this kind of distributed system was presented in [27]. The security mechanisms should satisfy the security requirements in order to protect the system from threats.

The QMS application intermediates the determination of the user profile. The subsets included in the set of users are as follows:

- Software developers, a subset consisting of programmers who create applications by writing code in different programming languages; this subset is defined as: $UT^D = \{UT_1^D, UT_2^D, \dots, UT_{nD}^D\}$.
- Software designers, the subset of users who design software that develops the software implementation specifications; this subset is given by: $UT^P = \{UT_1^P, UT_2^P, \dots, UT_{nP}^P\}$.
- Testers, the subset of users who test applications released by software developers and who will review the application before its release to the final users; they test the applications based on the testing specifications and the case tests from the design plan of the application and the subset is defined as: $UT^T = \{UT_1^T, UT_2^T, \dots, UT_{nT}^T\}$.

The set of users consists of the merger of the three homogeneous subsets:

$$\begin{aligned}
 UT &= UT^D \cup UT^P \cup UT^T \\
 UT &= \{UT_1^D, UT_2^D, \dots, UT_{nD}^D\} \cup \{UT_1^P, UT_2^P, \dots, UT_{nP}^P\} \cup \{UT_1^T, UT_2^T, \dots, UT_{nT}^T\}.
 \end{aligned} \tag{33}$$

The number of users, $NrUT$, is calculated by:

$$NrUT = nD + nP + nT. \tag{34}$$

In order to establish the subset to which one user belongs it is required to build his/her profile. The profile is built using a questionnaire with questions that aim at identifying the corresponding profile of user. A screenshot of the QMS application is presented in Figure 3.

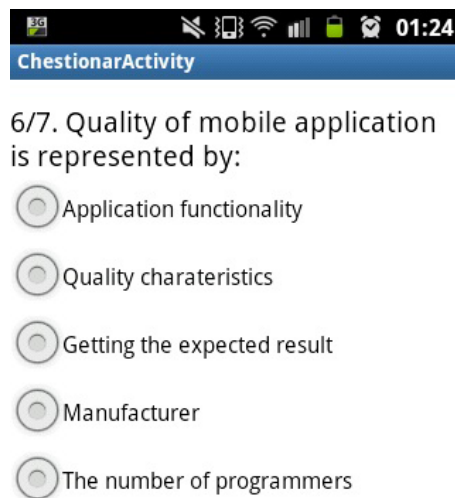


Figure 3. Example of a QMS application screen.

The questionnaire contains questions with a single correct answer, specific to a subset. So that users can respond correctly to all questions, questions with answers for each type of user are included. Thus, the answers to some questions will depend on the category to which the user belongs. For minimal interaction with the mobile device and to allow a rapid response of the controls, answers to the questions on the form are made automatically by marking the desired final response.

5.2. Quality characteristics weighting

In order to calculate the weights, 78 specialists from representative IT companies were selected. The specialists were required to have experience with mobile application development and usage. They scored each characteristic according to their experience. For the current case study, the following parameters were used:

- the number of MC characteristics (NC) is 10;
- the number of specialists (NS) is 78;
- the set of scores that a specialist can choose for a characteristic is represented by the following set: $A = \{ 0, 5, 10, 20, 30, 50, 70, 90, 120, 150 \}$. The total possible score of set A is $ST = 545$ and the higher value is chosen for the most important characteristic.

The subsets A_{SP_i} with $i = 1, 2, \dots, 78$ are available at <http://acs.ase.ro/tqm>. All specialists associated a unique weight from set A to each quality characteristic. Figure 4 shows the distribution of scores marked against each of the characteristics.

It should be noted that the extreme values and regenerability have been scored 32 times with 0 points and data security with a total of 26 of 120 points. This shows the importance given by specialists to each individual characteristic. The columns *amount* and *weight* based on these subsets are shown in Table 1.

From Table 1, the value of SS is 42,510 and the sum of weights is 1.

5.3. IA calculation

Let us consider the characteristics from the MC set for QMS application. The application was distributed to 30 users for testing. The users were selected from specialists with experience in interaction with mobile phones and

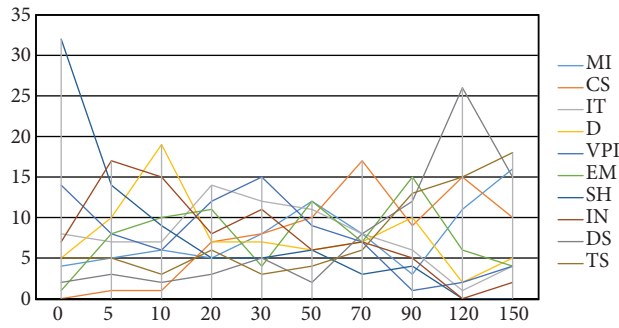


Figure 4. Quality characteristic scores given by specialists.

Table 1. Weights of quality characteristics.

Characteristic	Amount (SC_j)	Weight (W_j)
C1	5575	0.13
C2	6195	0.15
C3	3115	0.07
C4	3270	0.08
C5	2660	0.06
C6	4240	0.1
C7	1280	0.03
C8	2265	0.05
C9	7355	0.17
C10	6555	0.15
Total	42,510	1

use of mobile phones. Due to their having experience both in the use of mobile devices and software application development for these devices, we consider that the application testing is performed by experts in the mobile applications environment. For each user, login time was measured and the total time needed to establish the profile was noted. After all users established the profile, they completed a questionnaire that asked how natural it was interacting with the application and how handy it was. Users awarded grades that were used in the determination of values for the MII and DI indicators. CSI and ITI indicator values were calculated based on durations measured during users' interaction with the application.

Information provided by the application is structured such that only a question and possible answers to it are displayed at one time on the screen. The users were asked about the workload of the screen with the text necessary to provide useful information.

All 30 users successfully managed to use the application for profiling and finished their process.

The measurements were performed and the indicator values for specific indicators were calculated. The calculated values are presented in Table 2.

Based on measurements from Table 2, the IA (QMS) value is calculated:

$$IA(QMS) = \sum_{i=1}^{10} P_i \times W_i = 0.89. \tag{35}$$

According to the interval of representation, due to the fact that the value is in the range of [0.78, 0.92], the IA is considered to be representative and the mobile application QMS is considered to be good.

Table 2. Calculations for mobile application QMS.

No.	Characteristic	Indicator	Weight (W)	The calculated value for indicator (P)	$P_i \times W_i$
1	Mode of interaction	MII	0.13	1	0.13
2	Command speed	CSI	0.15	0.91	0.14
3	Interaction time	ITI	0.07	0.96	0.07
4	Drivability	DI	0.08	1	0.08
5	Volume of provided information	VPI	0.06	0.87	0.05
6	Error management	EMI	0.1	1	0.1
7	Self-healing	SHI	0.03	1	0.03
8	Integrability	INI	0.05	0.4	0.02
9	Data security	DSI	0.17	0.8	0.14
10	Transaction security	TSI	0.15	0.85	0.13
	Total		-	-	0.89

Increasing quality is achieved by improvement of application modules that were marked with small values for quality characteristics and associated indicators such as data security or integrability.

As a large database is built, the maximum and minimum levels are calculated in order to obtain the ideal indicator value, the reference indicator is selected, and the effective level of the indicator is calculated.

Transition to TQM creates the context for calculation of the IA. This requires the design from the beginning of a data acquisition module in order to obtain a grouping of fields for the elements used to calculate the weights. Even though the calculated weights are different from specialist to specialist, compliance should be considered to ensure sample representativeness and the accepted error level by its size.

6. Conclusions and future work

Any measurement system includes conventional elements. It is essential that the hypotheses used for building the aggregate indicator be accepted by those who use it. The acceptance begins with the definition of quality characteristics used for mobile applications. The agreement on the resulting weights W_1, W_2, \dots, W_{NC} is considered to be the continuation of the acceptance.

The process of acceptance ends with understanding the limits of any process of aggregation, characterized by loss of information.

For defining the IA, ten quality characteristics selected on an experimental basis were taken into account. The set of features was chosen to characterize the entire mobile applications and their capabilities and to highlight both strengths and weaknesses of mobile applications.

The IA is analyzed through the fulfillment of the requirements according to all indicators. It is sensitive, compensatory, noncatastrophic, and representative.

For these ten quality characteristics we created a questionnaire and after this the questionnaire was applied to 78 specialists. Weights were obtained for each characteristic of the IA.

Along with the IA the quality level was determined for a mobile application developed for the Android platform. Data for calculating the quality level were also obtained experimentally by testing the application of 30 experienced users in interacting with mobile devices and mobile application development.

Moving towards TQM is a matter of time, understanding that mobile applications exist specifically if they work independently of the developer and the processing is accurate, complete, and user-friendly compared

to requirements, allowing the covering of the cost of investment so that the cycle of releasing other mobile applications is continuous.

The new context requires the use of restructured databases in order to estimate new weights, resumption of their calculation, and data-gathering processes in order to calculate the IA.

The next steps include the calculation of the IA at an organizational level. This will require the identification of all elements, for which the corresponding IAs will be calculated and integrated.

References

- [1] Pressman S. *Software Engineering: A Practitioner's Approach*. 7th ed. New York, NY, USA: McGraw-Hill, 2009.
- [2] Sommerville I. *Software Engineering*. 9th ed. Boston, MA, USA: Addison-Wesley, 2011.
- [3] Akşit M. [The 7 C's for creating living software: a research perspective for quality-oriented software engineering](#). *Turk J Electr Eng Co* 2004; 12: 61–96.
- [4] Hoyle D. *Quality Management Essentials*. Burlington, MA, USA: Butterworth-Heinemann, 2007.
- [5] Gao K, Khoshgoftaar TM, Wang H, Seliya N. [Choosing software metrics for defect prediction: an investigation on feature selection techniques](#). *Software-Pract Exp* 2011; 41: 579–606.
- [6] Junglas IA, Watson RT. [Location based services evaluating user perceptions of location-tracking and location-awareness services](#). *Commun ACM* 2008; 3: 65–69.
- [7] Plaza I, Martin L, Martin S, Medrano C. [Mobile applications in an aging society: status and trends](#). *J Systems Soft* 2011; 84: 1977–1988.
- [8] Redd JB, Schmidt D. [Using mobile devices and gaming as a means of building vocabulary](#). *Intl J Interact Mob Tech* 2011; 4: 30–38.
- [9] Berton A, Bühler D, Minker W. [SmartKom - Mobile car: user interaction with mobile services in a car environment](#). *Lect Notes Artif Int* 2006; 5: 523–537.
- [10] Dunlop MD, Durga N, Motaparti S, Dona P, Medapuram V. [QWERTH: an optimized semi-ambiguous keyboard design](#). In: *14th International Conference on Human-Computer Interaction with Mobile Devices and Services Companion*; 21–24 September 2012; San Francisco, CA, USA. New York, NY, USA: ACM. pp. 23–28.
- [11] Bellavista P, Corradi A, Fanelli M, Foschini L. [A survey of context data distribution for mobile ubiquitous systems](#). *ACM Comput Surv* 2012; 4: 1–49.
- [12] International Organization for Standardization. *ISO/IEC 25010:2011, Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models*. Geneva, Switzerland: International Organization for Standardization, 2011.
- [13] Gomaa H, Hashimoto K, Kim M, Malek S, Menascé DA. [Software adaptation patterns for service-oriented architectures](#). In: *Symposium on Applied Computing*; 22–26 March 2010; Sierre, Switzerland. New York, NY, USA: ACM. pp. 462–469.
- [14] Salehie M, Tahvildari L. [Self-adaptive software: landscape and research, challenges](#). *ACM Trans Auton Adapt Syst* 2009; 2: 1–42.
- [15] Ferguson N, Schneier B, Kohno T. *Cryptography Engineering: Design Principles and Practical Applications*. Indianapolis, IN, USA: Wiley Publishing, 2010.
- [16] Popa M. [Requirements of a better secure program coding](#). *Inform Econ* 2012; 4: 93–104.
- [17] Boja C, Pocatilu P, Zamfiroiu A. [Data security in m-learning messaging services](#). *Intl J Comput Commun* 2011; 5: 119–126.
- [18] Mousa T, Rizik A, Marwah A, Doaa Q. [Innovative secure mobile banking services](#). *Intl J Interact Mob Tech* 2011; 1: 12–21.

- [19] Suduc AM, Bizoi M, Filip FG. User awareness about information systems usability. *Stud Inform Control* 2010; 19: 145–152.
- [20] Mayo RN, Ranganathan P. Energy consumption in mobile devices: why future systems need requirements-aware energy scale-down. *Lect Notes Comput Sc* 2005; 3164: 26–40.
- [21] Flinn J, Satyanarayanan M. PowerScope: a tool profiling the energy usage of mobile applications. In: *2nd Workshop on Mobile Computing Systems and Applications 1999*; 25–26 February 1999; New Orleans, LA, USA. Washington, DC, USA: IEEE. pp. 2–10.
- [22] Namboodiri V, Ghose T. To cloud or not to cloud: a mobile device perspective on energy consumption of applications. In: *IEEE 2012 International Symposium on a World of Wireless, Mobile and Multimedia Networks*; 25–28 June 2012; San Francisco, CA, USA. Piscataway, NJ, USA: IEEE. pp. 1–9.
- [23] Gafni R. Framework for quality metrics in mobile-wireless information systems. *Interdis J Inform Knowl Manage* 2008; 3: 23–38.
- [24] Paun G, Tataram M, Priboi T. *Aggregated Social Indicators and Some Theoretical Requirements*. Bucharest, Romania: University of Bucharest Publishing House, 1980.
- [25] Pele DT. *Metode statistice aplicate în analiza pieței de capital*. PhD, Bucharest University of Economic Studies, Bucharest, Romania, 2007 (in Romanian).
- [26] Ciurea CE. *Metricile sistemelor colaborative*. PhD, Bucharest University of Economic Studies, Bucharest, Romania, 2011 (in Romanian).
- [27] Rosado DG, Fernández-Medina E, López J. Security services architecture for secure mobile grid systems. *J Systems Architecture* 2011; 57: 240–258.