

Modelling the interrelation among software quality criteria using Computational Intelligence techniques

Yamilis Fernández Pérez¹, Carlos Cruz Corona^{2*}, José Luis Verdegay³

¹ *University of Informatics Sciences, Cuba,
E-mail: yamilisf@uci.cu*

² *University of Granada, Spain,
E-mail: carloscruz@decsai.ugr.es*

³ *University of Granada, Spain,
E-mail: verdegay@decsai.ugr.es*

Received 14 November 2017

Accepted 18 May 2018

Abstract

Software products quality assessment is a highly complex process, given the variety of criteria to consider. For a better understanding, they are organized in so-called software quality models. An important aspect of these models is their structural complexity, forming a hierarchical structure. At present, they have evolved towards the overlap and interrelation between these criteria. Modeling a general structure to represent quality models, extending the hierarchical taxonomies to form more complicated structures like graphs and taking into account the interrelation between criteria is the objective of this work. The proposed solution incorporates elements of Computational Intelligence, such as fuzzy logic, fuzzy linguistic modeling and the use of fuzzy cognitive maps (FCM). The application of this proposal in a real-world case shows that it is an operative solution, reliable, precise and of easy interpretation for its application in the industry.

Keywords: Software Quality, Computational Intelligence, Fuzzy Cognitive Map, Fuzzy Logic.

1. Introduction

In recent years, the use of software has become a vital product in organizations, both for their most important processes and for those of support; among them the management of human resources, knowledge, decision-making and the exchange of information. Thanks to it, the possibilities of negotiation increase and they become more productive. This situation creates an environment of competition and software specialization between production and marketing organizations. Furthermore, associated to it, attention to the quality of software products acquires a special connotation, becoming one of the differentiating elements in the worldwide scope among the developing companies.

Cuba has designed as one of the fundamental strategies for its development the computerization of society,

through the development of quality computer applications that allow technological independence. This implies a boom in its software industry, currently made up of 30 companies and a non-state sector with more than 1,400 programmers. That is how every day more computer solutions are embedded in the daily work of the institutions of the country.

The National Center for Software Quality (CALISOFT) is a Cuban entity created to offer services for evaluating the quality of computer products, software development processes and the provision of information technology services. It also has the task of establishing technical standards and procedures that normalize the national computer industry, directing it towards higher levels of quality.

The search for the quality of the systems has led to the creation of models and standards, which set the

desirable attributes for the evaluation and selection of software products.

The quality criteria to be taken into account in the assessment of a software product are grouped in different ways for a better understanding and usability. This taxonomy is known as quality models. The most widely used are those known as integral hierarchical models, which decompose the quality into criteria organized in the form of a n-ary tree. Such a hierarchical decomposition is a strategy widely used in different scientific disciplines. The most significant software product quality models are: McCall, Boehm, FURPS, Dromey, ISO 9126 and ISO 25000.

The quality criteria are usually interdependent, because the result of the preference of one criterion over another is influenced by the others. With the increasing level of understanding of quality, which transcends a simple taxonomy, the models have evolved towards the overlap and interrelation between these criteria. There are interrelationships between McCall's quality factors.¹ Besides, there has been determined the interdependence between the six external and internal characteristics for ISO 9126.^{2,3} This causes a group of criteria to influence quality in a contradictory way. For example, greater efficiency implies lower usability; better reliability means better functionality.

In spite of this, the proposed solutions are purely hierarchical.^{4,5,6} Nevertheless, the quality assessment from independent criteria causes some prejudice in favor or against the product assessment. This causes that sometimes a correspondence between the estimated quality of a product and the one that it really demonstrates during its use is not reflected.

As a result of the above problem, it is determined as objective of this paper to define and apply a quality model assessing the interdependence between criteria as a way to achieve a realistic estimate of quality, and therefore a better problem definition of evaluation and selection of software products.

The application of the Computational Intelligence techniques is a recognized way to describe mathematically all these situations. Unfortunately, and as far as we know, until now no much research has been done into this problem using this point of view.

For this, the proposal incorporates Computational Intelligence techniques, such as fuzzy logic, fuzzy linguistic modeling and the use of fuzzy cognitive maps (FCM), to take advantage of the tolerance that implies

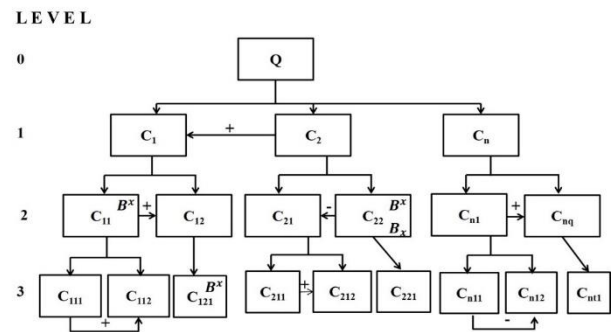


Fig. 1. Quality model.

imprecision and uncertainty, being able to consider elements as the manipulation of heterogeneous, ambiguous and imprecise information, and the analysis of the interdependence between criteria.

Consequently, this paper has the following structure: Section 2 presents the Quality Model proposed, in which the fuzzy cognitive maps, essential criteria and the order of the software tests are described. Section 3 describes a real-world case, and finally Section 4 is devoted to the conclusions.

2. Fuzzy Quality Model proposed

2.1. Quality model

The quality model proposed (QM) can be represented as a graph defined as:

$$QM = (V, E_v, E_h, EC). \quad (1)$$

where:

V is the set of evaluation criteria.

E_v is the set of vertical links.

E_h is the set of horizontal links (influence)

EC is the set of essential criteria

QM (see Fig. 1) is formed by 4 levels: level 0, represents the quality index; level 1, the characteristics; level 2, the sub-characteristics; and level 3, the metrics obtained from the software testing process and from expert assessments. A criterion is only found on one level and the union of all corresponds with the whole to be valued.

$$Level_0 = \{Q\} \cap_{0 \leq j \leq l} Level_j = \emptyset, \cup_{0 \leq j \leq l} Level_j = V. \quad (2)$$

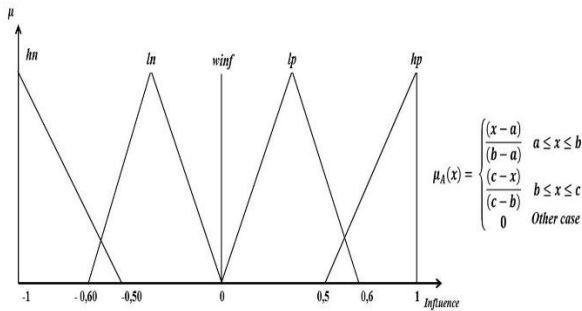


Fig. 2. Linguistic Variable Influence.

Each criterion has a weight (W) associated. At each level, there is a set of vectors of weight. The sum of the weights of the sibling criteria is equal to 1.

The vertical links (E_v), represent the relationship between criteria (vertex) of consecutive levels, i.e. the relationship between parent and children.

$$E_v \subset VxV,$$

$$E_v = \{(y, x) / x, y \in V, x \in Level_j, y \in Level_{j-1}\}. \quad (3)$$

The criteria at all levels have a parent that is at the previous consecutive level, except level 0, which represents the quality index (see Eq. (4)).

$$\forall_{0 < j \leq l} Level_j = \left\{ \begin{array}{l} x / (y, x) \in E_v, x \in V, \\ y \in Level_{j-1} \end{array} \right\}. \quad (4)$$

Horizontal links (E_h) represent the interdependence between criteria. Each link is presented by a triplet (x, y, I_{xy}) , which means the influence of the criterion x on the criterion y , with a weight (I_{xy}).

$$E_h \subset VxVx \{hn, ln, winf, lp, hp\},$$

$$E_h = \left\{ \begin{array}{l} (x, y, I_{xy}) / x, \\ x \in V, \exists z [(z, x) \in E_v, (z, y) \in E_v], \\ I_{xy} \in \{hn, ln, winf, lp, hp\} \end{array} \right\}$$

$$\forall_{x,y} [(x, y, I_{xy}) \in E_h \Rightarrow (y, x, I_{xy}) \notin E_h]. \quad (5)$$

(I_{xy}) is a linguistic variable, which establishes several labels: high negative (hn), low negative (ln), without influence ($winf$), low positive (lp) and high positive (hp), defined in the interval $[-1,1]$ (See Fig. 2).

The horizontal links are modeled through Fuzzy Cognitive Maps, which will be presented below.

2.2. Fuzzy Cognitive Maps

Fuzzy Cognitive Maps (FCM) is a technique developed by Kosko⁹, for quantitative modeling, based on the knowledge and experience of experts, and it is a fuzzy directed graph. The nodes represent concepts and weighted arcs the relationships between concepts. In the FCM, there are three possible types of relations between concepts: positive relation, negative relation or non-existence of relations. The degree of the relationship is described through a fuzzy number or linguistic value, defined in the interval $[-1,1]$. An FCM, consisting of n concepts, is represented in an $n \times n$ matrix, known as the adjacency matrix. This matrix is obtained from the values assigned to the arcs. They have been applied on diverse scientific areas, such as social and political sciences, engineering, information technology, robotics, expert systems, medicine, education, prediction, environment, and so on.^{10,11,12,13}

In this work, it is decided to treat the interdependence between criteria using Fuzzy Cognitive Maps, with the definition of the linguistic variable Influence (I).

To determine the interdependence between criteria of the same level, the FCM is formed from the information provided by the experts. The relationship between the sibling criteria is the first analyzed and a FCM is formed for each group of siblings. Nodes are sibling criteria and edges represent the influence of one criterion on another. The resulting FCM is reviewed by each expert and to each edge is associated a value of the variable I . The FCM obtained by each expert should be added later, using a technique that allows for consensus. It is advisable to use a consensus-building algorithm as proposed in Ref. 14.

The consensus fuzzy cognitive map ($FCMc$) is obtained for each group of siblings. From each $FCMc$, the adjacency matrix (AMc) is found, the different AMc of each level are combined and the matrix of interdependence between criteria is determined.

The unification of the different AMc is simple because there are no common criteria between the different maps. The combination is performed according to Eq. (6).

$$\begin{bmatrix} AMc_1 & winf & winf \\ winf & AMc_2 & winf \\ winf & winf & AMc_k \end{bmatrix}. \quad (6)$$

This matrix is called the criteria interdependence matrix (MI).

$$\widehat{MI}^l = \begin{bmatrix} \widehat{y}_{1,1} & \cdots & \widehat{y}_{1,j} & \cdots & \widehat{y}_{1,n} \\ \vdots & & \vdots & & \vdots \\ \widehat{y}_{i,1} & \cdots & \widehat{y}_{i,j} & \cdots & \widehat{y}_{i,n} \\ \vdots & & \vdots & & \vdots \\ \widehat{y}_{n,1} & \cdots & \widehat{y}_{n,j} & \cdots & \widehat{y}_{n,n} \end{bmatrix}. \quad (7)$$

\widehat{MI}^l represents the interdependence matrix between the n criteria of level l.

2.3. Essential criteria

In the model, there is a subset of criteria classified as essential (EC), determined from the quality requirements and the purpose of the assessment.

$$EC = \{(x, B_x, B^x) \mid B_x, B^x \in \mathbb{R}, B_x < B^x\}. \quad (8)$$

Where,

- x is essential criterion.
 - B_x : lower threshold value of criterion x .
 - B^x : upper threshold value of criterion x .
- A restriction is associated to each essential criterion, and in turn, an interval (B_x, B^x) is linked to it. With this formal and generic definition of the quality model, a better structuring of the problem of assessment and selection of software products is achieved.

2.4. Order of software tests.

From the level 1 criteria to be assessed, the types of tests to be performed are established. The interdependence between criteria of this level, defined through the FCM and their weight, serves as the basis for organizing and ordering software testing in the assessment process. Following, an algorithm is proposed for this purpose (See algorithm 1).

Algorithm 1 Pseudocode of the algorithm to recommend the order of the software tests.

Input:

Matrix of interdependence of level 1 (MI^1),

Weight vector of level 1 (W^1),

Output:

Order of tests to be performed on software products

BEGINNING

1. Find the absolute value of each element of MI^1
2. Calculate Total Effect Matrix
3. Calculate the centrality of each C_i
4. Calculate the ranking by the centrality
5. Calculate the ranking by weight.
6. Find the median of the rankings.
7. Sort by the median.
8. Return the order

END

The weight of the criteria can be established from the use of different methods, either by direct determination of the experts or by the comparison by pairs, thus, obtaining the eigenvector and considering that the sum of the weights of the siblings' criteria must equal 1.

For the calculation of the absolute value of each element, Eq. (9) is used.

$$Abs(\tilde{A}) = (\max(|b| - \delta, 0), |b|, \min(|b| + \delta, 1)),$$

$$\text{siendo } \delta = \begin{cases} \frac{|a-c|}{2} & \text{si } a \neq b \neq c \\ |a-c| & \text{si } a = b \text{ or } b = c \end{cases}. \quad (9)$$

Subsequently, the total effect matrix is found, according to Eq. (10).

$$T = \sum_{i=1}^{n-1} (\widehat{MI}^1)^i. \quad (10)$$

Where MI is the matrix of interdependence of level 1 and n is the number of criteria at that level.

The centrality of each characteristic is determined by Eq. (11):

$$CEN(C_i) = Inf(C_i) + Dep(C_i),$$

$$Inf(C_i) = \sum_{k=1}^n \bar{t}_{ik},$$

$$Dep(C_i) = \sum_{k=1}^n \bar{t}_{ki}. \quad (11)$$

A defuzzification method is applied, according to Eq. (12):

$$c(\tilde{A}) = \frac{a + 4b + c}{6}. \quad (12)$$

The ranking is calculated, according to the defuzzified value of $CEN(C_i)$.

2.5. Assessment and selection of software products.

In a decision-making problem of software products assessment and selection it is necessary to obtain the quality index for each product. This measurement permits its qualitative assessment.

If the objective of the evaluation is to recommend a product or set of products, then the first thing to do is to classify the quality index, from the highest to the lowest, and then recommend those defined as high quality products. However, if the objective is to make an ordering of the products, then a ranking is

recommended according to the quality index. Furthermore, if the assessment quality of a given product were to be obtained, this will show with its overall quality index and a qualitative assessment.

In order to obtain such index, first the different types of software tests should be executed, evaluating each alternative, taking into account the judgment of the experts. With the data collected, the value of each metric is calculated.

After this, the values of the different metrics are normalized and then unified by transforming them into a single data domain -- in this case, the diffuse triangular numbers. This operation is valid, only for problems where the information is heterogeneous. The evaluation matrix is conformed with the values of the normalized and unified measures for each alternative. After this the aggregation of the information is done in order to obtain the quality index of each alternative.

For such aggregation process, it is vital to take into account the interdependence among criteria; that is why, it is recommendable to follow the steps given below:

- (i) Add to the value of each criterion the influence of the others. To do this, aggregate the influence among criteria expressed in the interdependence matrix, with the evaluation matrix, and you will obtain Matrix G .

$$G = f(\widehat{M}e^l + (\widehat{M}e^l \times \widehat{M}I^l)). \quad (13)$$

where:

$f(x)$ is a threshold function $f(x) = \frac{1}{1+e^{-\lambda x}}$.

$\widehat{M}e^l$: evaluation matrix of level l .

$\widehat{M}I^l$: interdependence matrix of level l .

- (ii) The weighted matrix (Gp) is calculated using (14).

$$Gp = G \otimes W, \quad gp_{ij} = g_{ij} \cdot w_j. \quad (14)$$

where:

G , Matrix obtained in the previous step

W , vector of weight defined.

- (iii) Add the measures to obtain the values of the subcharacteristics through the weighed sum. A new evaluation matrix will be obtained with the value of the subcharacteristics for each alternative.

These steps are executed iteratively to get the values of the different characteristics as well as the quality index for each alternative.

3. Case of study

The previous algorithm and proposed model was applied in a controlled environment in CALISOFT, a Cuban software quality certification company, with the objective of planning the software tests for the quality assessment of three products. Taking into account the quality requirements, the characteristics and sub-characteristics to be assessed are determined. In this case, the level 1 criteria (characteristics) to be assessed are: Functional Adequacy (AF), Reliability (R), Efficiency (E), Usability (U) and Portability (P), according to ISO 25010. The weight vector for the quality characteristics (level 1) is determined through peer comparison (See Table 1). A study on the sensitivity of the resulting rankings in terms of slight modifications of the weights is outside the objectives of the article.

Table 1 Weight vector for the characteristics of level 1 used in Cuba.

Quality characteristics	Weight
Functional Adequacy (FA)	0.324
Efficiency (E)	0.075
Reliability (R)	0.324
Portability (P)	0.072
Usability (U)	0.206

Subsequently, the relationship between criteria of level 1 is defined, with the use of FCM (see Fig. 3). To establish the interdependence between criteria of the same level, the FCM_c is formed for the group of siblings. From the FCM_c , the adjacency matrix (AM_c), of the level is derived and obtains the matrix of interdependence between criteria (MI^1).

The proposed algorithm is applied and in Table 2, it shows the value of the Influence, Dependency and Centrality of each characteristic. Table 3 shows the rankings for different aspects.

The proposed order to perform the tests is: the tests of functionality (AF) in the first place, followed by tests of usability (U), reliability (R), Efficiency (E) and finally Portability (P). This order of software testing expedited the assessment process. From the beginning, they detected the Non Conformities (NC) of greater affectation to the general quality index. The result of the assessment is in correspondence with the behavior of the products in the deployment.

An experimental assessment was developed for three software products (S1, S2, S3). Table 4 shows the data of the quality model used, the characteristics, subcharacteristics and metrics selected, the data domain (DD) in which they are defined and whether it is cost (C) or benefit (B). As can be seen, metrics are defined in different data domains, with different scales, corresponding to objective and subjective factors. It is remarkable that measures such as ease of installation, satisfaction and appearance, take values from the perception of experts, who have different criteria and knowledge, which introduces a high degree of uncertainty.

Software tests were performed, the resulting data collected and metrics were obtained. The value of each measure for each software is shown in Table 5. After normalizing and unifying the data, the evaluation matrix (*Me*) was achieved, as it is shown in Table 6.

The value of each quality characteristic is shown in Table 7.

Another iteration of previous steps was performed to achieve the Quality Index (QI). The result obtained, is shown in Table 8. The recommendation as the best quality product is S2, then S3, and finally S1

The quality index obtained by the method suggested is compared with and without interdependence. In addition, the result is compared with a modified fuzzy TOPSIS, which does not include interdependence. The quality index according to each method is shown in table 9.

Furthermore, the correlation between the methods is illustrated in Table 10, using a scale of three colors, whereby green shows similar values while red shows different values.

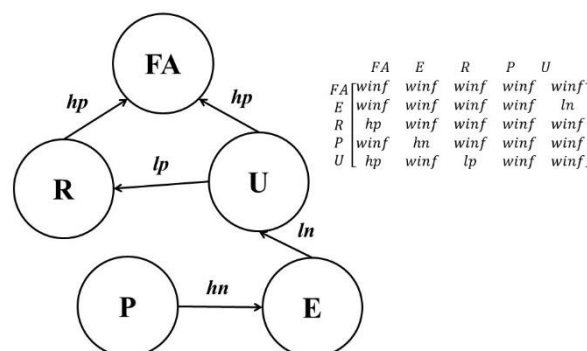


Fig. 3. Fuzzy Cognitive Map of level 1 and the adjacency matrix

Fuzzy TOPSIS does not allow interdependence, and it is considered as the most distant value method, since its conception is different in the sense that it deals with distance values towards a positive ideal and a negative one.

Table 2. Values of Influence, Dependency and Centrality of each characteristic.

	<i>Influence</i>	<i>Dependency</i>	<i>Centrality</i>	<i>Defuzzification</i>
FA	(0,0,0)	(1,3.08,4.52)	(1,3.08,4.52)	2.9733333
E	(0,0.78,1.92)	(0.5,1,1)	(0.5,1.78,2.92)	1.7566667
R	(0.5,1,1)	(0,0.48, 1.32)	(0.5,1.48,2.32)	1.4566667
P	(0.5,1.78,2.92)	(0,0,0)	(0.5,1.78,2.92)	1.7566667
U	(0.5,1.6,2.2)	(0,0.6,1.2)	(0.5,2.2,3.4)	2.1166667

Table 3. Rankings of each characteristic by its centrality, weight and the median of the rankings.

	<i>Ranking by Centrality</i>	<i>Ranking by weight</i>	<i>Median of the rankings</i>
FA	1	1	1
E	3	3	3
R	4	1	2.5
P	3	4	3.5
U	2	2	2

Table 4. Quality software criteria.

Characteristics	Subcharacteristics	Metric	DD	C/B
Functional Adequacy (FA)	Functional completeness (CpF)	Functional implementation coverage (CAF)	[0,1]	B
	Functional correctness (CcF)	Computational correctness (CC)	[0,1]	B
Efficiency (E)	Time behavior (Ct)	Task per time (TpT)	[0,40]	B
	Resource utilization (UR)	Memory utilization (MEM)	[0,10]	C
Reliability (R)	Maturity (M)	Mean time between failures (MTBF)	[0,100]	B
		Failure density against test cases (DF)	[0,1]	C
		Test coverage (CP)	[0,1]	B
Portability (P)	Installability (CapI)	Ease of installation (Fins)	(MB, B, M, A, MA)	B
	Adaptability (A)	Hardware environmental adaptability (Ahw)	[0,1]	B
		System software environmental adaptability (Asw)	[0,1]	B
Usability (U)	Appropriateness recognisability (Comp)	Exhaustive Description (DE)	[0,1]	B
	Learnability (Apr)	Completeness of documentation (IntDoc)	[0,1]	B
	Operability (Op)	Satisfaction (Sat)	(MB, B.M, A, MA)	B
	User interface aesthetics (Est)	Appearance (Apa)	(NA, PA, A, MA, AA)	B

Table 5. The value of each measure for each software product.

Quality metrics	S1	S2	S3
Functional implementation coverage (CAF)	1	0.4	0.2
Computational correctness (CC)	0.2	0.3	0.1
Task per time (TpT)	20	10	2
Memory utilization (MEM)	0.1	0.6	0.1
Mean time between failures (MTBF)	90	20	15
Failure density against test cases (DF)	0.6	0.2	0.4
Test coverage (CP)	0.4	0.7	0.8
Ease of installation (Fins)	B	M	MB
Hardware environmental adaptability (Ahw)	0.8	0.8	0.3
System software environmental adaptability (Asw)	0.9	0.8	0.4
Exhaustive Description (DE)	0.9	0.8	0.5
Completeness of documentation (IntDoc)	0.7	0.7	0.6
Satisfaction (Sat)	A	M	B
Appearance (Apa)	MA	AA	A

Table 6. The value of normalized and unified quality metrics.

Metrics	S1	S2	S3
CAF	MA	M	B
CC	B	B	MB
TpT	M	B	MB
MEM	MA	MA	MA
MTBF	MA	B	B
DF	M	A	M
CP	M	A	A
Fins	B	M	MB
Ahw	A	A	B
Asw	MA	A	M
DE	MA	A	M
IntDoc	A	A	M
Sat	A	M	B
Apa	MA	AA	A

Table 7. The value of quality characteristic.

	S1			S2			S3		
FA	0.623	0.631	0.631	0.675	0.734	0.752	0.500	0.500	0.500
E	0.620	0.645	0.667	0.613	0.637	0.659	0.554	0.558	0.558
F	0.500	0.500	0.500	0.640	0.653	0.665	0.634	0.648	0.661
P	0.649	0.661	0.669	0.647	0.660	0.672	0.628	0.639	0.652
U	0.695	0.754	0.771	0.692	0.753	0.767	0.623	0.685	0.699

Table 8. Quality index for each software product.

	QI	QI defuzificado	Ranking
S1	(0.516,0.586,0.621)	0.580513299	3
S2	(0.522,0.594,0.630)	0.588252946	1
S3	(0.521,0.591,0.623)	0.584475493	2

Table 9. Quality index (QI) obtained by different methods

QI	With interdependence	Without interdependence	Fuzzy TOPSIS
S1	0.5805	0.4712	0.7336
S2	0.5883	0.5310	0.5091
S3	0.5845	0.2544	0.3304

Table 10. Correlation among the methods by the quality index.

	With interdependence	Without interdependence	Fuzzy TOPSIS
With interdependence	1.0000	0.1922	-0.5670
Without interdependence	0.1922	1.0000	0.6994
Fuzzy TOPSIS	-0.5670	0.6994	1.0000

The experiment corroborates that the incorporation of the interdependence among criteria introduces changes in the solution of the problem; this is a key element to take into account in the selection and evaluation of software products.

4. Conclusions

Through the analysis of the models of quality of greater use in the industry and the determination of their similarities, it was possible to obtain a solution to the problem of modeling a generic and operative structure, through a graph. The proposal values the interdependence between criteria and the essential criteria. It also integrates the manipulation of ambiguous, imprecise information from different sources. It is based on elements of Computational Intelligence, such as fuzzy logic, fuzzy linguistic modeling and the use of fuzzy cognitive maps (FCM). It was inspired by real practical experiences developed by CALISOFT company.

It was demonstrated that the fuzzy cognitive map is a useful tool to identify and aggregate the interdependence between the criteria according to the opinion of the experts. Consequently, operating with FCM facilitates to make the order of the software tests.

The model facilitates and reduces the time for decision making, by creating a logical, rational and transparent basis for analysis. It also achieves a better structuring of the problem and therefore, greater participation and influence of all. Besides, it increases the depth of analysis, which leads to an increase in the quality of the decision.

The application of the proposal in a real-world environment shows that it is an operative, reliable and precise solution, which is easily interpreted for its application in industry.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness with the support of the project TIN2014-55024-P, and by the Regional Government of Andalusia—Spain with the support of the project P11-TIC-8001 (both including funds from the European Regional Development Fund). Last but not the least; also we would like to thank to EurekaSD, Erasmus Mundus Project for their support.

References

1. W. Perry, *Effective methods for EDI quality assurance*. Prentice-Hall, New Jersey, USA, 1987.
2. H. Zulzalil, A. Ghani, A. Azim, M. Hasan Selamat and R., "Using fuzzy integral to evaluate the web-based applications," *Interes. Gr. (MSEIG)*, 2011.
3. A. K. Pandey, "Analytical Network Process based Model to Estimate the Quality of Software Components," pp. 678–682, 2014.
4. K. Lee and S. J. Lee, "A Quantitative Evaluation Model Using the ISO / IEC 9126 Quality Model in the Component Based Development Process," in *Computational Science and Its Applications - ICCSA 2006 LNCS 3983*, 2006, pp. 917–926.
5. C.-W. Chang, C.-R. Wu, and H.-L. Lin, "Integrating fuzzy theory and hierarchy concepts to evaluate software quality," *Softw. Qual. J.*, vol. 16, no. 2, pp. 263–276, Nov. 2007.
6. A. Singh and S. K. Dubey, "Evaluation of Usability Using Soft Computing Technique," *Int. J. Sci. Eng. Res.*, vol. 4, no. 12, pp. 162–166, 2013.
7. H. J. Zimmermann, "Fuzzy set theory," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 3, pp. 317–332, 2010.
8. L. A. Zadeh, "Soft computing and fuzzy logic," *IEEE Softw.*, vol. 11, no. 6, pp. 48–56, 1994.
9. B. Kosko, "Fuzzy Cognitive Maps," *Int. J. Man -Machine Stud.*, vol. 24, no. April 1985, pp. 65–75, 1986.
10. E. I. Papageorgiou and J. L. Salmeron, "A review of fuzzy cognitive maps research during the last decade," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 1, pp. 66–79, 2013.
11. G. Nápoles and G. F. G. N., "A review on methods and software for Fuzzy Cognitive Maps," *Artif. Intell. Rev.*, no. August, pp. 1–31., 2017.
12. A. Mourhir, T. Rachidi, E. I. Papageorgiou, M. Karim, and F. S. Alaoui, "A cognitive map framework to support integrated environmental assessment," *Environ. Model. Softw.*, vol. 77, pp. 81–94, 2016.
13. A. Baykasoğlu and İ. Gölcük, "Development of a novel multiple-attribute decision making model via fuzzy cognitive maps and hierarchical fuzzy TOPSIS," *Inf. Sci. (Ny)*, vol. 301, pp. 75–98, 2015.
14. P. P. Groumpos, "Fuzzy Cognitive Maps: Basic theories and their application to complex systems," *Fuzzy Cogn. Maps*, vol. 247, pp. 1–22, 2010.