Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 535970, 12 pages
http://dx.doi.org/10.1155/2014/535970

Research Article

Software Component Selection Based on Quality Criteria Using the Analytic Network Process

Shah Nazir,[1] Sajid Anwar,[2] Sher Afzal Khan,[3] Sara Shahzad,[1] Muhammad Ali,[2] Rohul Amin,[4] Muhammad Nawaz,[2] Pavlos Lazaridis,[5] and John Cosmas[6]

[1]*Department of Computer Science, University of Peshawar, Peshawar 25120, Pakistan*
[2]*Centre of Excellence in IT, Institute of Management Sciences, Hayatabad, Peshawar, Pakistan*
[3]*Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan*
[4]*Department of Mathematics, University of Peshawar, Peshawar 25120, Pakistan*
[5]*TEI of Thessaloniki, Sindos, 57400 Thessaloniki, Greece*
[6]*Brunel University, Uxbridge UB8 3PH, UK*

Correspondence should be addressed to Shah Nazir; snshahnzr@gmail.com

Received 24 July 2014; Accepted 13 August 2014; Published 15 December 2014

Academic Editor: Saeed Islam

Copyright © 2014 Shah Nazir et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Component based software development (CBSD) endeavors to deliver cost-effective and quality software systems through the selection and integration of commercially available software components. CBSD emphasizes the design and development of software systems using preexisting components. Software component reusability is an indispensable part of component based software development life cycle (CBSDLC), which consumes a significant amount of organization's resources, that is, time and effort. It is convenient in component based software system (CBSS) to select the most suitable and appropriate software components that provide all the required functionalities. Selecting the most appropriate components is crucial for the success of the entire system. However, decisions regarding software component reusability are often made in an ad hoc manner, which ultimately results in schedule delay and lowers the entire quality system. In this paper, we have discussed the analytic network process (ANP) method for software component selection. The methodology is explained and assessed using a real life case study.

1. Introduction

With the passage of time software intensive systems are becoming larger and more complex which ultimately leads to the need to reuse previously developed components in order to raise productivity, reduce cost, and improve quality. Due to these economic considerations, there is a tendency towards components based software development, and studies show that recently such systems development is exceeding 40% of the total developed software systems [1]. In component based software development, software development is characterized by piecing together some prefabricated components into a working software system. These prefabricated components must be well-defined, easy to comprehend, simple to accept, suitably general, and easy to replace.

Compositional approaches for software development have benefitted greatly from the emergence of component based software development which has subsequently generated considerable interest in research and development in industry standards for component interaction, domain specific architectures, toolkits, and many other related areas. However, it is also evident that all this research has not been able to construct large scale industrial systems from existing parts on an economical scale. One of the reasons can be a lack in the compatibility of existing components for a new system to be constructed and another reason can be our inability to locate the desired pieces when they do exist [2].

Literature [3, 4] from the past ten years reveals that the integration process for open source and COTS components is quite different from custom software development. Custom software development is normally characterized by a traditional process consisting of requirements, design, development, test, and deployment activities whereas COTS based development is characterized by assessment, selection, composition, integration, test, and deployment activities [5, 6]. The assessment and selection are the vital steps for the selection of most suitable COTS components and connectors [7]. The suitability for selection requires consideration of many factors for a particular tool in order to build an expert system [8]. The software developer selects the framework for their projects which helps them to establish the best components model [9].

The contribution of this paper is to offer a methodology for the selection of software components. In the proposed method, analytic network process (ANP) developed by Saaty [10] has been used for software component selection. ANP applies feedback and dependencies with a structure of a network. In certain circumstances, where elements are dependent on each other and feedback is also needed, the ANP is sufficient. In ANP elements are grouped in a network of different clusters. The clusters consist of different elements or nodes connected in a network to each other. ANP uses pairwise comparison of elements in different stages and for different attributes. It finds the eigenvector (E.V.) of pairwise comparison. The advantage of ANP over other methods is that it allows tangible or intangible factors. The ANP is also the best approach for weight comparison. It is a powerful tool to deal with complex networks in decision making [11].

The rest of the paper is organized as follows. Next section presents related work. ANP method for software component selection is presented in Section 3. Quality attributes for selection of software components are discussed in Section 4. Section 5 presents ANP and ISO/IEC 25010:2011 mapping. Derivation of weights from expert opinion is presented in Section 6. The paper is concluded in Section 7.

2. Related Work

Several diverse methodologies are used and recommended by the software engineering community for the selection of COTS software components. Rikard et al. present the literature survey of different methodologies used for selection of COTS components and explain the practices of these methods [12]. Kaur and Mann present an approach for evaluation criteria of reusable software components [13]. The approach for component selection is part of the off-the-shelf option (OTSO) method. Fahmi and Choi maintain that the earlier decision and knowledge on component selection reduce time and also help to guide whether the selection was successful or failed. They introduced case based reasoning (CBR) for the selection of components [14]. Aamodt and Plaza present an overview of the basic issues related to case based reasoning [15]. Cai et al. present a survey on software component technologies. They show their merits and demerits and the features which they inherit. They also proposed the QA model for component based software development. The method focuses on superiority analysis of components, maturity of components, customization of components, design and amalgamation of components, and their maintenance [16].

Boehm et al. developed a set of data for three activities which consists of COT assessment, COT tailoring, and glue code development and integration [17]. Merceron and Pinna worked on verification of synchronous setting of components. The methodology is illustrated by using a protocol as a case study [18]. Kwong et al. have proposed a model which has two objectives: maximizing the purposeful performance in the CBSS and maximizing the consistency and decreasing the blend of module of software [19]. Velazquez et al. have proposed a study to find out the impact by using computer assisted software for measurement and selection of components [20]. Horváth measures the complex network properties and describes the method with the help of some examples [21]. Dias-Neto and Travassos designed a Porantim based strategy which selects model based testing for software projects [22]. Liu and Moughal presented a dynamic multicriteria decision making procedure [23]. Nazir et al. proposed a novel fuzzy logic based software component selection for the selection of software components. In the proposed model different fuzzy rules are designed and on the basis of these rules inputs are given to the model. It gives output and selects the highest priority components [24].

Jadhav and Sonar described the general methodology for software selection, the evaluation criteria, and hybrid based knowledge which assess the decision maker in the decision making for the selection of software [25]. Cai et al. proposed quality assurance for both the component and the system of the component. ComPARE is used to assess real life component evaluation and validation [26]. Ke et al. [27] proposed a method called the rCOS, used to focus on the model of a system at different levels, faultlessly, for the development of software process, and their integration, analysis, transition, validation, and verification. Zhiqiao et al. proposed an integrated decision model which assists the decision maker in the selection for component accomplishment and concurrently the best possible number of test cases for corroboration [28]. Ayala et al. explore the industrial practice for component selection which provides early experimental basis to allow the understanding of research and manufacturing activities [29]. Becker and Rauber have proposed evidence based approach which helps in component evaluation [30]. The condition of functional homogeneity of components and high number of component can improve repeatability and reproducibility.

Lee et al. used component specification technique and definitions of some components are described. The operators defined are component version, functional requirements, nonfunctional requirements, and cooperating component. Z scheme is used for the specification of components [31]. Alghabban and Qureshi proposed a component selection framework based on pliability metric for software quality. The method is validated by a sample of online questionnaire [32]. Tang et al. proposed an optimization model to solve the problem of reusability and compatibility. The model evaluates software developers in selecting software components [33].

Panagiotou and Mentzas used KnowBench that supports the knowledge management process for the designing and implementation of software [34]. Khan and Mahmood have proposed a component selection process that uses a graph model which in turn is signed for interdependencies of CBS and group related goals into clusters [35].

However, no previous work has been done on the selection of software components based upon attributes of quality criteria when there are dependencies among the attributes. Hence, to overcome this limitation of dependencies among attributes, the proposed ANP method is suggested that is well-organized and incorporates the quality attributes of ISO/IEC 25010:2011 [36].

3. Analytic Network Process for Software Component Selection

The motivation of the ANP method toward selection of software components is to the dependencies existing among attributes of the network of elements. The ANP method is very efficient in dealing with such phenomena. ANP has various applications such as [37–39]. The details on ANP can be found in Saaty (1996) [10]; however, the main steps are summarized as follows.

(1) The problem is divided into a network of subproblems.

(2) A qualitative scale of importance which is presented by Saaty is as follows: (equal importance allocate similarity to objectives), moderately important (somewhat good turn action above a new), reasonable plus, strong weight (powerfully good turn one action above another), strong plus, very well-built confirmed important (very powerfully favor above another), very, very strong, and excessive importance are given. The scale is converted into a quantitative scale of range between 1 and 9.

(3) Pairwise comparison is done in step (3). The criteria in the "**i**" row are compared with the criteria in the "**j**" column in the form of (i j). If the criteria of "**I**" row are superior to "**j**" column, then it is written as (i j), and (j i) is the reciprocal of (i j).

(4) The relative importance is calculated by finding the principal eigenvalue and the related eigenvector of the comparison matrix. The elements are normalized and are termed as weights of the criteria or subcriteria.

(5) After pairwise comparisons the consistency of the matrix is measured.

Priority vector "w" is calculated as follows:

$$A_w = \lambda_{\max} w, \tag{1}$$

where $\lambda_{\max}$ is the major eigenvalue of matrix "A" and "w" is its eigenvector. The value of "λ" is obtained by summing the column values of the original matrix multiplied by the normalized eigenvector. The principal eigenvector is obtained by the sum of all "λ."

Table 1: Random consistency index.

n	1	2	3	4	5	6	7	8	9
RI	0	0	0.6	0.9	1.1	1.2	1.3	1.4	1.5

$$\begin{array}{c} \\ C_1 e_{11} \\ \\ \\ \\ C_2 \\ \\ \\ C_N \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_n \\ e_{11}e_{12}\ldots e_{1n1} & e_{21}e_{22}\ldots e_{2n2} & & e_{N1}e_{N2}\ldots e_{NnN} \end{array} \\ \begin{array}{c} e_{12} \\ \vdots \\ e_{1n1} \\ e_{21} \\ e_{22} \\ \vdots \\ e_{2n2} \\ e_{N1} \\ e_{N2} \\ \vdots \\ e_{NnN} \end{array} \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & & \vdots \\ v_{N1} & v_{N2} & \cdots & v_{NN} \end{bmatrix} \end{array}$$

Figure 1

The "consistency index (CI)" and "consistency random (CR)" of pairwise comparison matrix are computed by the following equation:

$$c_i = \frac{\lambda_{\max} - n}{n-1}, \tag{2}$$

$$\mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}}. \tag{3}$$

The random consistency (RI) table is given by Saaty [40] (see Table 1).

The value of consistency ratio (CR) must be less than 0.1; otherwise, normalize the matrix.

(6) A super matrix is obtained by combining the entire matrix in a whole. In the super matrix if the column sum is greater than 1, it is unweighted super matrix. Normalize the unweighted super matrix till its column values become equal to or less than 1.

(7) Convert the weighted super matrix (summarized matrix in which the column sum is less than or equal to 1) to the limit matrix.

(8) Decide the most appropriate alternative from limit matrix. The matrix is in the form shown in Figure 1.

Figure 2 visually shows the different steps involved in the ANP method.

Figure 3 shows the phases in the proposed method based on ANP method.

4. Quality Attributes for Selection of Software Component

Various models such as McCall et al. [41], Boehm et al. [42], FURPS [43], Chen et al. [44], ISO/IEC 9126 [45], and ISO/IEC 25010:2011 [36] have been presented. The works by McCall et al. have been created to organize heterogeneous quality attributes of software. Nazir et al. used the attributes

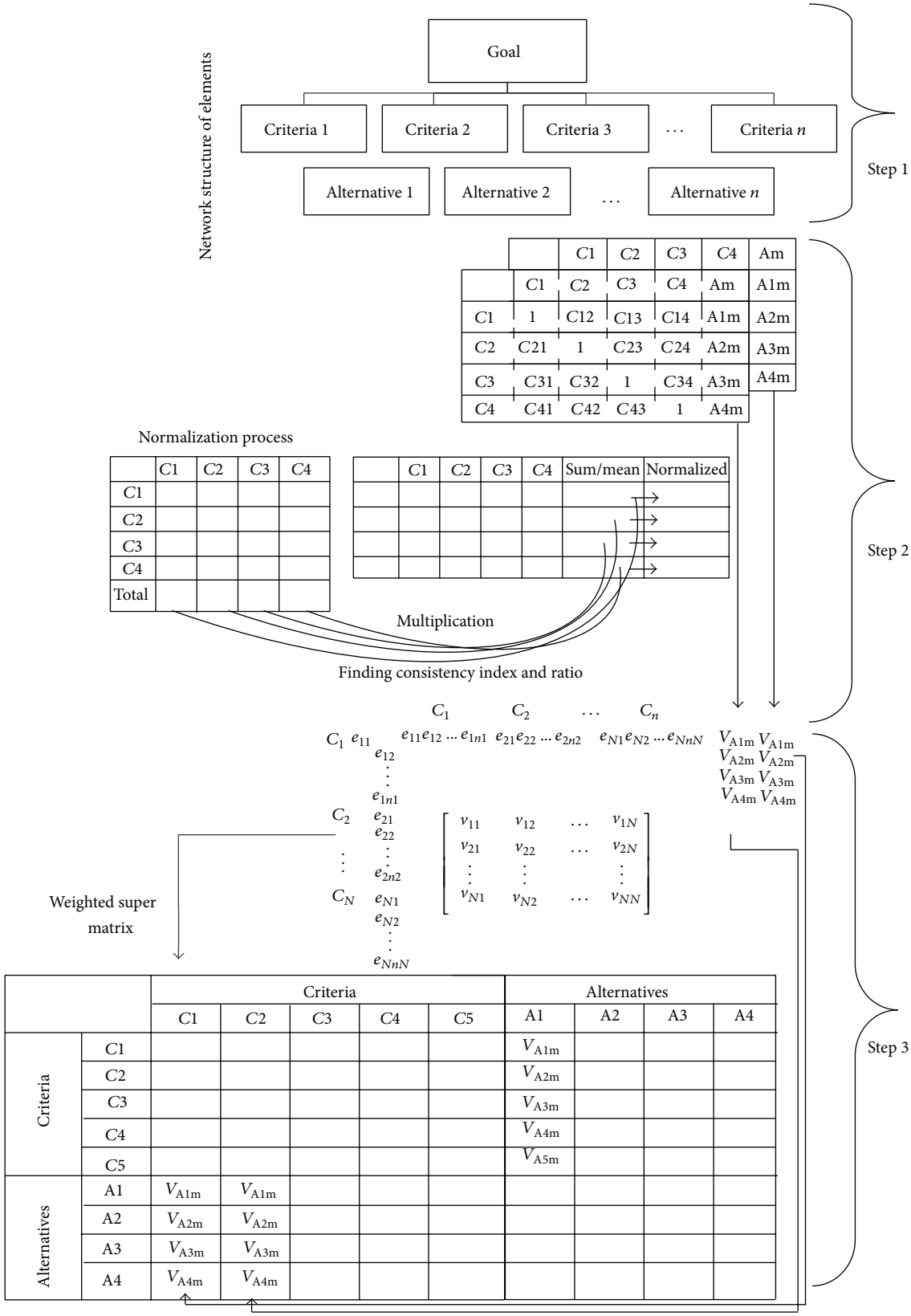


Figure 2: Graphical representation of the steps involved in ANP process.

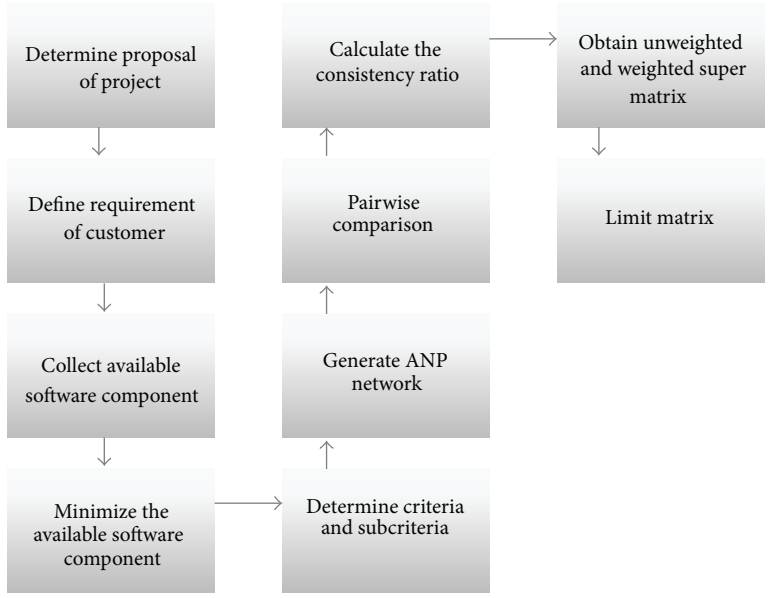


Figure 3: Phases involved in the proposed method based on ANP.

of ISO/IEC 27002 standard for the evaluation of security of software component [46]. All of these approaches generally categorize quality attributes into the following three categories:

(i) product operation: quality attributes expected in operations of final product such as correctness, efficiency, usability, and integrity;

(ii) product revision: quality attributes are essential when making changes in the software such as maintainability, testability, and flexibility;

(iii) product transition: quality attributes are essential when organization is transforming one product into another product such as portability, reusability, and interoperability.

Boehm et al. [42] devised their own quality model which is highly inspired by McCall's quality model. This new model adds further two qualities attributes, that is, understandability and modifiability. The overall quality attributes in this model are modifiability, portability, reliability, efficiency, usability, testability, and understandability.

Another quality model called FURPS quality model was initially applied by Unified Process which was later extended to FURPS+ with some additional quality requirements [43]. The FURPS name was coined after the first letter of the five quality attributes that comprise this model which are functionality, usability, reliability, performance, and supportability. An ISO standard termed ISO/IEC 9126 describes software product quality in terms of internal quality, external quality, and quality in use [45]. This model helps in identifying tradeoffs between various software product capabilities. This model establishes a one-to-one relationship between a quality attribute and its subattributes, thus making it more comprehensive and providing a wider coverage to various software capabilities and their tradeoffs. Liu [47] mentioned that informal methods in software engineering are facing great challenges in making certain software quality. On the other hand formal methods endeavor to deal with these challenges by using some mathematical notation. Ma et al. [48] present a systematic approach to metamodel quality assessment. The model effectively assesses the quality of metamodels and describes and classifies the quality attributes. Wang and Li present the topological structure of vague soft sets [49]. Galli et al. work on the framework to trace software product quality and address ambiguity existing in quality measurement [50].

In the proposed method, the ISO/IEC 25010:2011 software product quality model has been used for the component selection. The model includes attributes which are effectiveness, efficiency, satisfaction, safety, and usability.

5. Mapping ANP and ISO/IEC 25010:2011

In the proposed method the selection of software components is based upon the quality criteria. We also know that for system quality criteria ISO/IEC 25010:2011 standard is used [36]. ANP is based on (1) goal, (2) criteria, and (3) alternatives. Figure 4 visually describes ANP using ISO/IEC 25010:2011 quality model.

Mathematical software component selection is represented as

$$\text{selection} = \sum_{i \in P} P_i, \tag{4}$$

where P = {effectiveness, efficiency, satisfaction, safety, usability}.

According to Saaty's algorithm, the fundamental scales for judgment are given that show us which component is

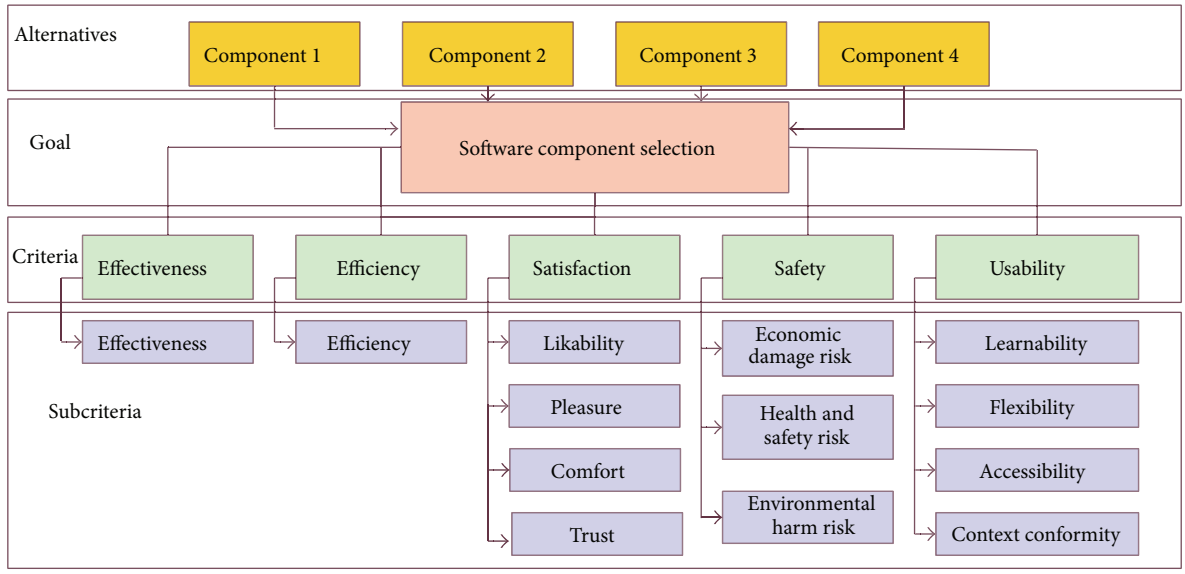


Figure 4: Proposed software component selection model based on ISO/IEC 25010:2011 using ANP.

more important than the other. Table 2 [40] also shows how much one component is important than the other.

6. Derivation of Weights from Expert Opinions

Weights for various parameters are a result of a conscious brainstorming between multiple domain experts. These experts have been actively involved in the development of various ERP solutions in nationally recognized universities. Initially, a set of 15 experts was chosen who were given a complete overview of the model and parameters which contribute to its formation. These experts after a long debate came up with the relative importance of each parameter in pairwise qualitative comparisons. A normalized geometric means of inputs from 15 experts was applied to overcome variations in opinion of experts while extreme values were excluded. This resulted in the relative consensus weight of each parameter. For validation purposes, initially four components, component 1 ($C1$), component 2 ($C2$), component 3 ($C3$), and component 4 ($C4$), were selected and inserted into an $\mathbf{n} * \mathbf{n}$ matrix. Relative weights of components are shown as C_{ij}, where "**i**" indicates the rows and "**j**" indicates the columns of the matrix. If the relative importance of component C_i is equal to component C_j than $C_{ij} = 1$, $C_{ji} = 1$. So, "1" will be inserted into the position where C_i is compared to C_{ji} (main diagonal). As shown in the following matrix (5), we inserted "1" in $C11$, $C22$, $C33$, and $C44$:

$$\begin{pmatrix} & C1 & C2 & C3 & C4 \\ C1 & 1 & & & \\ C2 & & 1 & & \\ C3 & & & 1 & \\ C4 & & & & 1 \end{pmatrix}. \quad (5)$$

Table 2: Scale for weightage.

Values intensity	Explanation
1	Equal importance (give similarity to objectives)
2	Weak
3	Moderately important (somewhat good turn action above a new)
4	Reasonable plus
5	Strong weight (powerfully good turn one action above another)
6	Strong plus
7	Very well-built confirmed important (very powerfully favor above another)
8	Very, very strong
9	Excessive importance (highest possible order affirmation)

Once the parameter weights were decided, the step-by-step ANP process for the selection of student registration components (proposed component selection) was as follows. This matrix (6) shows the comparison of components with respect to effectiveness

$$\begin{pmatrix} & C1 & C2 & C3 & C4 \\ C1 & 1.0 & 2.0 & 4.0 & 5.0 \\ C2 & 0.5 & 1.0 & 2.0 & 2.0 \\ C3 & 0.2 & 0.5 & 1.0 & 3.0 \\ C4 & 0.2 & 0.5 & 0.3 & 1.0 \\ \text{Total} & 1.9 & 4.0 & 7.3 & 11.0 \end{pmatrix}. \quad (6)$$

Matrix (7) presents the normalization process involved in the process of pairwise comparison in matrix (6) as follows:

$$\begin{pmatrix} & C1 & C2 & C3 & C4 \\ C1 & \frac{1}{1.90} = 0.53 & \frac{2}{4} = 0.50 & \frac{4}{7.30} = 0.55 & \frac{5}{11} = 0.45 \\ C2 & \frac{0.5}{1.90} = 0.26 & \frac{1}{4} = 0.25 & \frac{2}{7.30} = 0.27 & \frac{2}{11} = 0.18 \\ C3 & \frac{0.2}{1.90} = 0.11 & \frac{0.5}{4} = 0.13 & \frac{1}{7.30} = 0.14 & \frac{3}{11} = 0.27 \\ C4 & \frac{0.2}{1.90} = 0.11 & \frac{0.5}{4} = 0.13 & \frac{0.3}{7.30} = 0.04 & \frac{1}{11} = 0.09 \end{pmatrix}. \quad (7)$$

Matrix (8) shows the normalized values derived from the normalization process in matrix (7) as follows:

$$\begin{pmatrix} & C1 & C2 & C3 & C4 & \text{E.V.} \\ C1 & 0.53 & 0.50 & 0.55 & 0.45 & 0.507 \\ C2 & 0.26 & 0.25 & 0.27 & 0.18 & 0.242 \\ C3 & 0.11 & 0.13 & 0.14 & 0.27 & 0.160 \\ C4 & 0.11 & 0.13 & 0.04 & 0.09 & 0.091 \end{pmatrix}. \quad (8)$$

For finding the eigenvalues, the sum of the columns of the original matrix (weights derived from expert opinions) and the sum of the row of normalized (new) matrix are multiplied. The process is shown in Figure 5.

Find the consistency ratio by using the formula in (3). The same process of calculation has been done for the remaining matrices below. Matrix (9) shows the pairwise comparison of components with respect to efficiency as follows:

$$\begin{pmatrix} & C1 & C2 & C3 & C4 & \text{E.V.} \\ C1 & 1 & 3 & 2 & 4 & 0.472 \\ C2 & \frac{1}{3} & 1 & 2 & 2 & 0.236 \\ C3 & \frac{1}{2} & \frac{1}{2} & 1 & 2 & 0.186 \\ C4 & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & 1 & 0.106 \\ \text{CR} = 0.05 & & & & & \end{pmatrix}. \quad (9)$$

Matrix (10) shows the pairwise comparison of components with respect to satisfaction as follows:

$$\begin{pmatrix} & C1 & C2 & C3 & C4 & \text{E.V.} \\ C1 & 1 & 4 & 4 & 7 & 0.592 \\ C2 & \frac{1}{4} & 1 & 2 & 3 & 0.205 \\ C3 & \frac{1}{4} & \frac{1}{2} & 1 & 2 & 0.131 \\ C4 & \frac{1}{7} & \frac{1}{3} & \frac{1}{2} & 1 & 0.072 \\ \text{CR} = 0.03 & & & & & \end{pmatrix}. \quad (10)$$

Matrix (11) shows the pairwise comparison of components with respect to safety as follows:

$$\begin{pmatrix} & C1 & C2 & C3 & C4 & \text{E.V.} \\ C1 & 1 & 1 & 4 & 7 & 0.443 \\ C2 & 1 & 1 & 2 & 5 & 0.343 \\ C3 & \frac{1}{4} & \frac{1}{2} & 1 & 3 & 0.153 \\ C4 & \frac{1}{7} & \frac{1}{5} & \frac{1}{3} & 1 & 0.060 \\ \text{CR} = 0.02 & & & & & \end{pmatrix}. \quad (11)$$

Matrix (12) shows the pairwise comparison of components with respect to usability as follows:

$$\begin{pmatrix} & C1 & C2 & C3 & C4 & \text{E.V.} \\ C1 & 1 & 2 & 3 & 8 & 0.516 \\ C2 & \frac{1}{2} & 1 & 2 & 3 & 0.260 \\ C3 & \frac{1}{3} & \frac{1}{2} & 1 & 2 & 0.149 \\ C4 & \frac{1}{8} & \frac{1}{3} & \frac{1}{2} & 1 & 0.075 \\ \text{CR} = 0.008 & & & & & \end{pmatrix}. \quad (12)$$

The following values from matrices (13) to (16) are used for the quality attributes with respect to components. This matrix shows the pairwise comparison of component with respect to Component ($C1$)

$$\begin{pmatrix} & \text{Effectiveness} & \text{Efficiency} & \text{Satisfaction} & \text{Safety} & \text{Usability} & \text{E.V.} \\ \text{Effectiveness} & 1 & 2 & 3 & 3 & 4 & 0.384 \\ \text{Efficiency} & \frac{1}{2} & 1 & 2 & 4 & 2 & 0.255 \\ \text{Satisfaction} & \frac{1}{3} & \frac{1}{2} & 1 & 2 & 4 & 0.180 \\ \text{Safety} & \frac{1}{3} & \frac{1}{4} & \frac{1}{2} & 1 & 2 & 0.104 \\ \text{Usability} & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \frac{1}{2} & 1 & 0.077 \\ \text{CR} = 0.07 & & & & & & \end{pmatrix}. \tag{13}$$

Matrix (14) represents the pairwise comparison of component with respect to Component ($C2$)

$$\begin{pmatrix} & \text{Effectiveness} & \text{Efficiency} & \text{Satisfaction} & \text{Safety} & \text{Usability} & \text{E.V.} \\ \text{Effectiveness} & 1 & 3 & 3 & 5 & 4 & 0.444 \\ \text{Efficiency} & \frac{1}{3} & 1 & 2 & 3 & 4 & 0.239 \\ \text{Satisfaction} & \frac{1}{3} & \frac{1}{2} & 1 & 2 & 3 & 0.158 \\ \text{Safety} & \frac{1}{5} & \frac{1}{3} & \frac{1}{2} & 1 & 2 & 0.093 \\ \text{Usability} & \frac{1}{4} & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 & 0.066 \\ \text{CR} = 0.05 & & & & & & \end{pmatrix}. \tag{14}$$

The following matrix (15) represents the pairwise comparison of component with respect to Component ($C3$)

$$\begin{pmatrix} & \text{Effectiveness} & \text{Efficiency} & \text{Satisfaction} & \text{Safety} & \text{Usability} & \text{E.V.} \\ \text{Effectiveness} & 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & 0.092 \\ \text{Efficiency} & 2 & 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{5} & 0.099 \\ \text{Satisfaction} & 3 & 2 & 1 & \frac{1}{2} & \frac{1}{4} & 0.159 \\ \text{Safety} & 3 & 3 & 2 & 1 & \frac{1}{2} & 0.247 \\ \text{Usability} & 2 & 5 & 4 & 2 & 1 & 0.403 \\ \text{CR} = 0.09 & & & & & & \end{pmatrix}. \tag{15}$$

Matrix (16) represents the pairwise comparison of component with respect to Component ($C4$)

$$\begin{pmatrix} & \text{Effectiveness} & \text{Efficiency} & \text{Satisfaction} & \text{Safety} & \text{Usability} & \text{E.V.} \\ \text{Effectiveness} & 1 & \frac{1}{3} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & 0.071 \\ \text{Efficiency} & 3 & 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{5} & 0.109 \\ \text{Satisfaction} & 2 & 2 & 1 & \frac{1}{2} & \frac{1}{4} & 0.135 \\ \text{Safety} & 3 & 3 & 2 & 1 & \frac{1}{3} & 0.221 \\ \text{Usability} & 4 & 5 & 4 & 3 & 1 & 0.464 \\ \text{CR} = 0.07 & & & & & & \end{pmatrix}. \quad (16)$$

6.1. Weighted Super Matrix. When the total sum of a column of a matrix is less than or equal to 1, then this matrix is called weighted super matrix. The value which is greater than 1 will have to be normalized until the sum of the column values is less than or equal to 1. The following matrix (17) shows the weighted super matrix:

$$\begin{pmatrix} & & & \text{Quality attributes} & & & \text{Avaliable components} & & & \\ & \text{Effectiveness} & \text{Efficiency} & \text{Satisfaction} & \text{Safety} & \text{Usability} & C1 & C2 & C3 & C4 \\ \text{Effectiveness} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.384 & 0.444 & 0.092 & 0.071 \\ \text{Efficiency} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.255 & 0.239 & 0.099 & 0.109 \\ \text{Satisfaction} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.180 & 0.158 & 0.159 & 0.135 \\ \text{Safety} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.104 & 0.093 & 0.247 & 0.221 \\ \text{Usability} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.077 & 0.066 & 0.403 & 0.464 \\ C1 & 0.507 & 0.472 & 0.592 & 0.443 & 0.516 & 0.000 & 0.000 & 0.000 & 0.000 \\ C2 & 0.242 & 0.236 & 0.205 & 0.343 & 0.260 & 0.000 & 0.000 & 0.000 & 0.000 \\ C3 & 0.160 & 0.186 & 0.131 & 0.153 & 0.149 & 0.000 & 0.000 & 0.000 & 0.000 \\ C4 & 0.091 & 0.106 & 0.072 & 0.060 & 0.075 & 0.000 & 0.000 & 0.000 & 0.000 \end{pmatrix}. \quad (17)$$

6.2. Limit Matrix. The result obtained in the limit matrix is to raise the power of the weighted super matrix until its values of column become the same and stable. It consists of the summary of the whole pairwise comparison matrices. Limit matrix consists of limit priority of all indirect influences among elements. The limit matrix (matrix (18)) is the final matrix where the decision maker can make decisions easily

$$\begin{pmatrix} & & & \text{Quality attributes} & & & \text{Avaliable components} & & & \\ & \text{Effectiveness} & \text{Efficiency} & \text{Satisfaction} & \text{Safety} & \text{Usability} & C1 & C2 & C3 & C4 \\ \text{Effectiveness} & 0.3264 & 0.3264 & 0.3264 & 0.3260 & 0.3264 & 0 & 0 & 0 & 0 \\ \text{Efficiency} & 0.2139 & 0.2139 & 0.2139 & 0.2137 & 0.2139 & 0 & 0 & 0 & 0 \\ \text{Satisfaction} & 0.1673 & 0.1673 & 0.1673 & 0.1671 & 0.1673 & 0 & 0 & 0 & 0 \\ \text{Safety} & 0.1337 & 0.1337 & 0.1337 & 0.1335 & 0.1337 & 0 & 0 & 0 & 0 \\ \text{Usability} & 0.1583 & 0.1583 & 0.1584 & 0.1582 & 0.1584 & 0 & 0 & 0 & 0 \\ C1 & 0 & 0 & 0 & 0 & 0 & 0.5064 & 0.5064 & 0.5063 & 0.5064 \\ C2 & 0 & 0 & 0 & 0 & 0 & 0.2508 & 0.2508 & 0.2508 & 0.2508 \\ C3 & 0 & 0 & 0 & 0 & 0 & 0.1580 & 0.1580 & 0.1580 & 0.158 \\ C4 & 0 & 0 & 0 & 0 & 0 & 0.0843 & 0.0843 & 0.0843 & 0.0843 \end{pmatrix}. \quad (18)$$

Figure 6 visually describes the final weights (output).

From Figure 6 it is clear that "component ($C1$)" is the best choice for selection followed by "component ($C2$)" followed by "component ($C3$)" and then by "component ($C4$)." This selection of component has been done based on the predefined ISO/IEC 25010:2011 quality criteria. The selected component is well functional and reliable for the purpose of its selection.

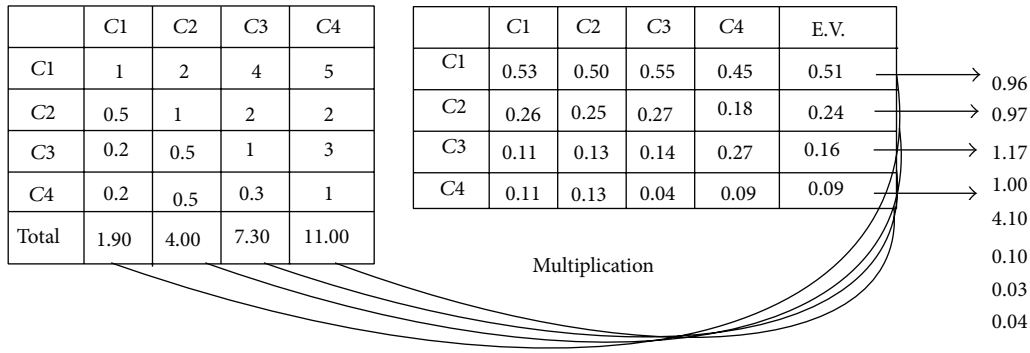


Figure 5: Finding eigenvalues.

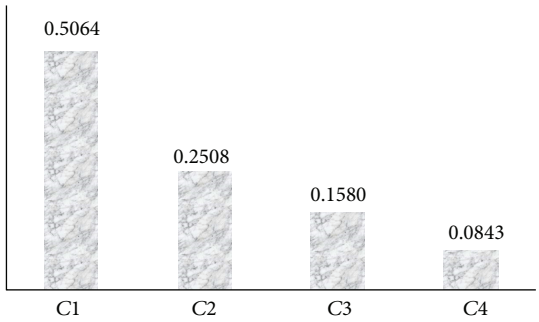


Figure 6: Graphical weights of available components.

7. Conclusion

The selection of the most suitable software component can increase productivity, reduce cost, and improve overall quality to its full potential. These economic considerations of component based software development have ultimately resulted in considerable investment in the development of domain specific architecture, several toolkits, and industry standards development for components and interaction. However, despite all these efforts paramount to the success of the entire component based software development process, is the selection of the most suitable component(s) based upon some certain quality criteria.

In this research study, we evaluated the applicability of ANP method for the selection of software component(s) based on a set of quality attributes. ANP method is widely used for the problems of decision making in complex environments. Initially in the proposed method of software component selection the network of goal criteria and their associated alternatives were designed. The network was structured according to the criteria of ANP. These network structures, different components along with their criteria, help developer to easily understand the details of the components with their criteria. After the network structure designing weights were derived from experts' opinions. The pairwise comparison has been done for both the components and their criteria. When the calculations of the pairwise comparisons are done, the consistency ratio is also found. All the pairwise comparison is summarized in weighted and limit matrix. The limit matrix shows the final weight of the available software components and from this matrix developer can make decisions about the most appropriate and suitable software component.

The results of the proposed method clearly show that the method is quite beneficial and favorable in decision making regarding the most suitable software component selection. Hence, it is concluded that ANP is one of the best choices for software component selection.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] A. Rawashdeh and B. Matalkah, "A new software quality model for evaluating COTS components," *Journal of Computer Science*, vol. 2, pp. 373–381, 2006.

[2] D. Garlan, R. Allen, and J. Ockerbloom, "Architectural mismatch or why it's hard to build systems out of existing parts," in *Proceedings of the IEEE 17th International Conference on Software Engineering (ICSE '95)*, pp. 179–185, Seattle, Wash, USA, April 1995.

[3] A. A.-e.-S. Abd-Allah, *Composing heterogeneous software architectures [Ph.D. thesis]*, Faculty of the Graduate School, University of Southern California, Los Angeles, Calif, USA, 1996.

[4] D. Yakimovich, *A Comprehensive Reuse Model for COTS Software Products*, University of Maryland, College Park, Md, USA, 2001.

[5] C. Albert and L. Brownsword, "Evolutionary Process for Integrating COTS-Based Systems (EPIC): An Overview," Tech. Rep., Software Engineering Institute, 2002.

[6] Y. Yang, J. Bhuta, B. Boehm, and D. N. Port, "Value-based processes for COTS-based applications," *IEEE Software*, vol. 22, no. 4, pp. 54–62, 2005.

[7] G. L. Hamza-Lup, A. Agarwal, R. Shankar, and C. Iskander, "Component selection strategies based on system requirements' dependencies on component attributes," in *Proceedings of the IEEE International Systems Conference Proceedings (SysCon '08)*, pp. 322–326, April 2008.

[8] K. Eldrandaly, "An Intelligent MCDM approach for selecting the suitable expert system building tool," *The International Arab Journal of Information Technology*, vol. 4, pp. 365–372, 2007.

[9] H. Aris and S. Salim, "State of component models usage: justifying the need for a component model selection framework," *The International Arab Journal of Information Technology*, vol. 8, no. 3, pp. 310–317, 2011.

[10] T. L. Saaty, "Fundamental of the analytic network process," in *Proceedings of the ISAHP*, pp. 1–14, Kobe, Japan, 1999.

[11] P. Palanisamy, A. Zubar, and S. Kapoor, "A model for supplier selection using analytic network process," in *Proceedings of the 10th International Conference on Operations and Quantitative Management (ICOQM '10)*, pp. 808–814, Nashik, India, 2011.

[12] L. Rikard, B. Laurens, C. Michel, and C. Ivica, "COTS selection best practices in literature and in industry," in *Proceedings of the 10th International Conference on Software Reuse: High Confidence Software Reuse in Large Systems*, Springer, Beijing, China, 2008.

[13] A. Kaur and K. S. Mann, "Component selection for component based software engineering," *International Journal of Computer Applications*, vol. 2, pp. 109–114, 2010.

[14] S. A. Fahmi and H.-J. Choi, "A study on software component selection methods," in *Proceedings of the 11th International Conference on Advanced Communication Technology (ICACT '09)*, vol. 1, pp. 288–292, Phoenix Park, Ireland, February 2009.

[15] A. Aamodt and E. Plaza, "Case-based reasoning: foundational issues, methodological variations, and system approaches," *AI Communications*, vol. 7, no. 1, pp. 39–59, 1994.

[16] X. Cai, M. R. Lyu, K. F. Wong, and R. Ko, "Component-based software engineering: technologies, development frameworks, and quality assurance schemes," in *Proceedings of the 7th Asia-Pacific Software Engineering Conference*, pp. 372–379, Washington, DC, USA, 2000.

[17] B. Boehm, D. Port, and Y. Yang, "WinWin spiral approach to developing COTS-based applications," in *Proceedings of the 5th International Workshop on Economics-Driven Software Engineering Research (EDSER-5)*, pp. 1–6, Oregon, Ore, USA, 2003.

[18] A. Merceron and G. M. Pinna, "Component-based verification in a synchronous setting," *International Journal of Software Engineering and Knowledge Engineering*, vol. 11, no. 2, pp. 181–203, 2001.

[19] C. K. Kwong, L. F. Mu, J. F. Tang, and X. G. Luo, "Optimization of software components selection for component-based software system development," *Computers and Industrial Engineering*, vol. 58, no. 4, pp. 618–624, 2010.

[20] O. C. Velazquez, E. Y. Woo, J. P. Carpenter, M. A. Golden, C. F. Barker, and R. M. Fairman, "Decreased use of iliac extensions and reduced graft junctions with software-assisted centerline measurements in selection of endograft components for endovascular aneurysm repair," *Journal of Vascular Surgery*, vol. 40, no. 2, pp. 222–227, 2004.

[21] Á. Horváth, "The cxnet complex network analyser software," *Acta Polytechnica Hungarica*, vol. 10, pp. 43–58, 2013.

[22] A. C. Dias-Neto and G. H. Travassos, "Model-based testing approaches selection for software projects," *Information and Software Technology*, vol. 51, no. 11, pp. 1487–1504, 2009.

[23] S. Liu and T. A. Moughal, "A novel method for dynamic multicriteria decision making with hybrid evaluation information," *Journal of Applied Mathematics*, vol. 2014, Article ID 864628, 11 pages, 2014.

[24] S. Nazir, M. A. Khan, S. Anwar, H. Khan, and M. Nazir, "A novel fuzzy logic based software component selection modeling," in *Proceedings of the International Conference on Information Science and Applications (ICISA '12)*, pp. 1–6, May 2012.

[25] A. S. Jadhav and R. M. Sonar, "Framework for evaluation and selection of the software packages: a hybrid knowledge based system approach," *Journal of Systems and Software*, vol. 84, no. 8, pp. 1394–1407, 2011.

[26] X. Cai, M. R. Lyu, and K.-F. Wong, "Component-based embedded software engineering: development framework, quality assurance and a generic assessment environment," *International Journal of Software Engineering and Knowledge Engineering*, vol. 12, no. 2, pp. 107–133, 2002.

[27] W. Ke, X. Li, Z. Liu, and V. Stolz, "rCOS: a formal model-driven engineering method for component-based software," *Frontiers of Computer Science in China*, vol. 6, no. 1, pp. 17–39, 2012.

[28] W. Zhiqiao, C. K. Kwong, J. Tang, and J. W. K. Chan, "Integrated model for software component selection with simultaneous consideration of implementation and verification," *Computers and Operations Research*, vol. 39, no. 12, pp. 3376–3393, 2012.

[29] C. Ayala, Ø. Hauge, R. Conradi, X. Franch, and J. Li, "Selection of third party software in Off-The-Shelf-based software development: an interview study with industrial practitioners," *Journal of Systems and Software*, vol. 84, no. 4, pp. 620–637, 2011.

[30] C. Becker and A. Rauber, "Improving component selection and monitoring with controlled experimentation and automated measurements," *Information and Software Technology*, vol. 52, no. 6, pp. 641–655, 2010.

[31] J. Lee, C. Yoo, and O. Chang, "Component contract-based interface specification technique using Z," *International Journal of Software Engineering and Knowledge Engineering*, vol. 12, no. 4, pp. 453–469, 2002.

[32] W. G. Alghabban and M. R. J. Qureshi, "The proposal of improved component selection framework," *Life Science Journal*, vol. 10, pp. 3538–3544, 2013.

[33] J. F. Tang, L. F. Mu, C. K. Kwong, and X. G. Luo, "An optimization model for software component selection under multiple applications development," *European Journal of Operational Research*, vol. 212, no. 2, pp. 301–311, 2011.

[34] D. Panagiotou and G. Mentzas, "Leveraging software reuse with knowledge management in software development," *International Journal of Software Engineering and Knowledge Engineering*, vol. 21, no. 5, pp. 693–723, 2011.

[35] M. A. Khan and S. Mahmood, "A graph based requirements clustering approach for component selection," *Advances in Engineering Software*, vol. 54, pp. 1–16, 2012.

[36] S. Wagner, "Quality models," in *Software Product Quality Control*, Springer, Berlin, Germany, 2013.

[37] H. Lee, S. Lee, and Y. Park, "Selection of technology acquisition mode using the analytic network process," *Mathematical and Computer Modelling*, vol. 49, no. 5-6, pp. 1274–1282, 2009.

[38] B. Pang and S. Bai, "An integrated fuzzy synthetic evaluation approach for supplier selection based on analytic network process," *Journal of Intelligent Manufacturing*, vol. 24, no. 1, pp. 163–174, 2013.

[39] Y.-C. Hu, "Analytic network process for pattern classification problems using genetic algorithms," *Information Sciences*, vol. 180, no. 13, pp. 2528–2539, 2010.

[40] T. L. Saaty, "Relative measurement and its generalization in decision making. Why pairwise comparisons are central in mathematics for the measurement of intangible factors. The analytic hierarchy/network process," *Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales A: Matematicas*, vol. 102, no. 2, pp. 251–318, 2008.

[41] J. A. McCall, P. K. Richards, and G. F. Walters, "Factors in software quality," in *Preliminary Handbook on Software Quality for an Acquisition Manager*, Technical Report, General Electric, 1977.

[42] B. Boehm, J. Brown, H. Kaspar, M. Lipow, G. McLeod, and M. Merritt, *Characteristics of Software Quality*, TRW Series of Software Technology, 1978.

[43] B. G. Robert, *Practical Software Metrics for Project Management and Process Improvement*, Prentice-Hall, 1992.

[44] J. Chen, L. Ma, C. Wang, H. Zhang, and J. Wan, "A new uncertainty evaluation method and its application in evaluating software quality," *Journal of Applied Mathematics*, vol. 2014, Article ID 145285, 9 pages, 2014.

[45] IS Organization, "Software Engineering—Product Quality," Parts 1–4, ISO/IEC9126, 2001.

[46] S. Nazir, S. Shahzad, M. Nazir, and H. U. Rehman, "Evaluating security of software components using analytic network process," in *Proceedings of the 11th International Conference on Frontiers of Information Technology (FIT '13)*, IEEE, Islamabad, Pakistan, 2013.

[47] S. Liu, "Formal engineering methods for software quality assurance," *Frontiers of Computer Science in China*, vol. 6, no. 1, pp. 1–2, 2012.

[48] Z. Ma, X. He, and C. Liu, "Assessing the quality of metamodels," *Frontiers in Computer Science*, vol. 7, no. 4, pp. 558–570, 2013.

[49] C. Wang and Y. Li, "Topological Structure of Vague Soft Sets," *Abstract and Applied Analysis*, vol. 2014, Article ID 504021, 8 pages, 2014.

[50] T. Galli, F. Chiclana, J. Carter, and H. Janicke, "Towards introducing execution tracing to software product quality frameworks," *Acta Polytechnica Hungarica*, vol. 11, pp. 5–24, 2014.