Is a Multi-Criteria Evaluation Tool Reserved for Experts?

C. Sanga and I. M Venter
University of the Western Cape, South Africa
csanga@uwc.ac.za
iventer@uwc.ac.za

Abstract:
The objective of this investigation was to determine whether the analytical hierarchy process algorithm is suitable for the evaluation of software by evaluators with little Information Technology experience. The scope of the research was the evaluation of two free and open source e-learning systems at the Open University of Tanzania using 33 stakeholders with diverse levels of Information Technology experience.
Both quantitative and qualitative research methods were used. The qualitative methods comprised participative observation and interviews. Questionnaires and the analytical hierarchy process, a multiple-criteria decision-making algorithm, represented the quantitative methods. The results showed that of the two e-learning systems evaluated, Moodle was preferred over ATutor. Furthermore it was found that the analytical hierarchy process algorithm is appropriate for the evaluation of software in a situation where Information Technology experience is limited. It is anticipated that the paper contributes to the theory and practise of decision making in developing countries such as Tanzania.

Keywords: free and open source software, e-learning systems, software quality, multi-criteria evaluation tool, analytical hierarchy process, novice user, developing country

1. Introduction

Finding suitable software for a specific purpose is a universal problem. Available software needs to be considered and some method found to determine which application is more suitable for the purpose of the application and the needs of the user. Evaluating available software for a specific purpose is in most cases left to software experts and the views of stakeholders are used only if the experts deem it necessary.

In the commercial world, software packages are developed for specific usages and the opinion of stakeholders' are valued and incorporated throughout the software development cycle. This is not the case with free and open source software (FOSS). FOSS is often developed by software specialists interested in a specific application (Nichols and Twidale, 2003). They develop software for this application and then make it freely available for everyone who is interested in adapting it or changing it for their own needs. Experts are able to adapt FOSS for their needs, however it is difficult for end-users to evaluate, choose an appropriate software application (from the many that are freely available on the Internet) and adapt it for their specific needs.

In many organisations the selection of FOSS is done by trial and error: Software packages are introduced and if they do not "*live up to expectations*" another package is found and introduced until most of the needs of the users are satisfied. However there are methods for the evaluation of software, for example the qualitative weight and sum (QWS) method, the analytical hierarchy process algorithm (AHP) etc. These methods, each with its pros and cons, are employed by software specialists to evaluate software.

The Open University of Tanzania (OUT) is a distance learning institution that is in the process of considering using e-learning system to distribute their study material and manage their students. Since this is a new approach, OUT is still in the process of appointing more Information Technology (IT) personnel to implement and maintain an e-learning system. Currently IT experience at OUT is limited; therefore if a method could be found where any stakeholder could participate in the evaluation of the software it would benefit this environment. It is from this stance that we evaluated two e-learning systems at OUT using the analytical hierarchy process algorithm. The reason for choosing AHP is because it allows the incorporation of both subjective judgments and objective decision-making. For the subjective evaluation we made use of 33 evaluators: 30 students (so-called stakeholders) and three IT experts or staff. These were the only stakeholders that had experience with both the e-learning packages that were being considered.

Reference this paper as:
Sanga, C and Venter, I,M.(2009) "Is a Multi-criteria Evaluation Tool Reserved for Experts?" *The Electronic Journal Information Systems Evaluation Volume 12 Issue 2 2009, (pp165 - 176), available online at www.ejise.com*

Our assumption is that AHP is a sophisticated tool which is generally used by technical users when evaluating software. The question that needs to be asked is: Can any user of a computer system, deployed in a developing country, be asked to participate in the evaluation of FOSS with AHP?
Traditional methods of software evaluation involve expert judgments (Boehm, 1981a). Usually an expert evaluates software independently; the results from different experts are then merged to find the most appropriate software for a specific purpose. This approach is subjective and can thus be biased according to Somerville (2004). A more objective approach was suggested by Basili, Caldeira, and Rombach (1994) who used the Goal Question Metric (GQM) paradigm to evaluate software. They found that the advantage of GQM is that it is very flexible. They however indicated that a disadvantage of GQM is that it can only be used by experienced users who can define and answer questions relevant to the purpose of the evaluation (which differs depending on the environments). A further disadvantage is that GQM neither allows for the comparison of software systems nor does it provide a means of combining different attributes into a single value for the sake of comparison.

To evaluate FOSS e-learning systems, Graf and List applied a qualitative weight and sum (QWS) method (Graf and List, 2005:164). QWS has the advantage of being simple and novice users can be used in the evaluation. However, its disadvantage is that it is subjective and it is complex to compare the evaluation of the different software packages. This is because symbols are used (rather than numerical values) when assigning software quality attributes.

Saaty (1980:78) showed that the AHP model can be used to solve any multi-criteria decision making problem. The advantage of using AHP is that it is simple and caters for a step-wise evaluation of software which makes it suitable for evaluating FOSS.

In developing countries the success of FOSS depends on the following factors:

- user acceptance (keeping in mind that users are from varied cultural and educational backgrounds with differing computer expertise).
- resources (since resources are scarce) and
- the quality of the software

These factors (mentioned above) impact on the evaluation of e-learning systems according to Wesson (2003:52). Other factors that may influence users' impression of the software may be the quality of the content/course materials available for the e-learning system being evaluated. Researchers have categorized the quality of software as either technological or pedagogical (Colace *et al*., 2003). The focus of this paper is not on the pedagogical, but rather on the technical factors, that determine software quality. But what determines the quality of software? The quality of software is determined by: (i) the quality of the development process; (ii) the quality of the product as well as; (iii) the quality of the product-in-use (Fenton and Pfleeger, 1997:100). Manieri, Begin, Meglio, and Rippa (2008) argue that most FOSS applications are poorly documented in terms of requirement specification, system architecture use and system prototypes tested. Consequently it is difficult to determine the quality of the FOSS development process and thus the best way to evaluate FOSS software, is to consider the software product-in-use.

Nichols and Twidale (2003) argue that the rationale for involving novice users in the evaluation of software is that software developers of FOSS do not necessarily design and develop software with features that take the typical user into account. It is therefore important to involve the ordinary user in the evaluation process. Technical and non-technical users perceive software differently and thus we advocate the participation of all stakeholders to bridge the gap between the novice and the experienced user.

OUT uses traditional distance learning methods for teaching and learning (McHarazo and Olden, 2004:205). Currently there is an initiative to improve learning and teaching at OUT through the implementation of e-learning. Many FOSS e-learning systems are available on the Internet. From these, OUT chose ATutor and Moodle as the e-learning systems to test in a pilot project. The idea is to select one of these two for full implementation in the near future. This will enable OUT to have access to an inexpensive but effective e-learning system to support learning and teaching. However the cost of a FOSS e-learning system lies not only in the creation of content for the system, but also in the customization and the maintenance of the system.

E-learning is becoming an attractive alternative to traditional distance learning in Tanzania because of the recent growth of Internet usage (and coverage) (Isamuyo, 2006:10). Students scattered throughout the country will be able to access e-learning systems at work, Internet cafés, community centres, telecentres or at home. The use of blended learning, traditional learning and teaching complemented by e-learning, is thus an appropriate option for OUT. Furthermore the university as well as the government has enacted information communication and technology (ICT) policy to guide the incorporation of ICTs into teaching and learning, creating opportunities for e-learning.

We argue that, in order for the uptake and implementation of FOSS to be successful in Tanzania or another developing country, certain criteria of software quality must be considered. In our case study, we considered three software quality criteria, namely: usability, maintainability, and deployability, which we consider to be the most important criteria to consider in a developing country. These criteria depend on the environment and context where the system is implemented. They are not universal in all developing countries due to the difference in availability of resources, experience of users to maintain the systems and the environment where the software systems must be deployed.

The quality of software depends on the user’s ability to interact with the software; the knowledge to maintain the software and the resources available to deploy the system in particular environment. These three criteria are quite challenging in Tanzania because of: (i) the computer illiteracy of the population; (ii) the low bandwidth of networks and; (iii) the fact that some user interfaces will be in a language which is foreign to the Swahili user (Swahili is the language spoken in Tanzania).

Both qualitative and quantitative methods were used in this study. The qualitative method comprised collecting data on site using participative observation and focused group interviews. Quantitative methods consisted of data collection using questionnaires and data analysis using AHP.

AHP, a multi-criteria decision-making (MCDM) algorithm, uses pairwise comparisons to derive weights (of importance) for the attributes identified (Saaty, 1980). AHP was used to evaluate two FOSS e-learning systems: Moodle and ATutor.

The results showed that the evaluation was consistent for the attributes that were identified for the criteria: deployability; maintainability and usability. It confirmed that ordinary users can be employed in the evaluation process and indicated that the e-learning package Moodle is preferred over ATutor.

2. Method

The basic steps for software product evaluation, as identified by Comella-Dorda *et al*. (2002:87), were followed. They involve planning the evaluation process; identifying criteria for evaluation; collecting data; and finally analyzing the data.

2.1 Planning the evaluation process

The participants were selected by means of purposive sampling (Van Vuuren and Maree, 1999:281). Purposive sampling was used because there were only a limited number of users, with the necessary experience, who could participate in the study. Thirty three participants from OUT (30 second year BSc. Information Communication Technology students and three IT lecturers) with experience in both ATutor and Moodle were selected to participate in the study. They were given a consent form to complete before taking part in the research project.

2.2 Identification of attributes that were used in the data collection

For the three main criteria (usability, maintainability, and deploy-ability), sub-criteria were identified as proposed in the literature (Boehm *et al*., 1976b:595; Cavano and McCall, 1978:136; IEEE, 1998; ISO, 2001). For each of these sub-criteria, the authors and participants decided on attributes that best describe the selected sub-criteria. These attributes were combined with attributes identified in the literature (Coleman *et al*., 1994; ISO, 2001; Bertoa and Vallecillo, 2002:65).

Table 1: The hierarchical structure of the quality attributes for the proposed evaluation

LEVEL 1	LEVEL 2	LEVEL 3
criteria	sub criteria	attributes

usability	learnability	time to configure
		time to expertise
		time to use
	understandability	assistance / training
		user documentation
		help system
		demonstration coverage
	operability	effort to operate
		tailorability
		administrability
	attractiveness	confidence
		error correction and prevention
		user control
		satisfaction
		general user support
		informative feedback to user
		compatibility with user conventions and expectations
		consistency of screen presentation
	usability compliance (conformance)	standardization
	complexity	complexity of the provided interface
maintai nability	stability	occurrence of error
	analyzability	tracing error
	changeability	customizability
		extensibility
		portability
	testability	observability
		controllability
		accesibility
	trackability	look & feel
	flexibility	scalability
	upgradeability	easy to upgrade
deploy-ability	portability	software system independence
		machine independence
	installability	ease of installation
	adaptability	suitability for personalization
		adaptivity
	configurability	technical documentation
	distributability	distributed system

The identified criteria, sub-criteria, and attributes were combined and used in the formulation of a questionnaire (some of the questions were adopted from a Software Usability Measurement Inventory (SUMI) questionnaire (Ryu, 2005: 192)).

Attributes which the authors felt were duplicated, were removed from the questionnaire. The aim was to come up with a list of representative attributes agreed to by all the participants (see Table 1).

2.3 Data collection

The questionnaire was refined by means of a pilot study. In the actual study participants were asked to complete the questionnaire and some of the participants were interviewed using open ended questions or probes. In addition, the authors took field notes (i.e. participative observation) during the

data collection process. This provided rich data which informed the research process. Focus groups helped to obtain a wish-list of weights for the criteria and sub-criteria (weights allocated to level 1 and level 2 of Table 1).
It was decided to use output from the questionnaire as input to the AHP method. This was done instead of allowing the users to directly use AHP (Turban, 1993:221). Some researchers might argue that this is not ideal but we did this to facilitate non technical users' involvement in the evaluation process.

2.4 Data analysis

All the questions in the questionnaire required categorical responses. Numerical values were arbitrarily allocated to these categorical responses (Stevens, 1946:679) (see Table 2). Mapping software quality phenomena to a numerical value is accepted in measurement theory and software measurement (Fenton and Pfleeger, 1997).

According to Turban (1993:221) defining intensities (measures) the way we did in this research, is the best approach to avoid comparison ambiguity associated with large number of criteria and alternatives.

Table 2: An example of the interpretation of the attribute symbols

Symbol	A	B	C	D	E
Nominal	Strongly agree	Agree	Disagree	Strongly disagree	I don't have enough information to answer
Scale	9	7	5	3	1

There are two methods for computing the combined group decisions in AHP: (i) first method by using either the arithmetic mean or geometric mean of individual respondents' comparison judgments (i.e. opinions) and (ii) second method by using either the arithmetic mean or geometric mean of the individually calculated priorities (Forman and Peniwati, 1998). The first method is used if the group of the respondents wants to act as a unit while the second method is used if the group wants to act as a combination of individuals. In our case study we used the first method.

2.4.1 Detailed explanation of data analysis

The collected data was analyzed empirically using AHP (Saaty, 1980) since AHP simplifies the quantification of software quality attributes and allows for a more objective evaluation process. In AHP a problem is decomposed into a number of hierarchical levels (similar to Table 1), these are then compared pairwise and prioritization is determined.

The pairwise comparison of elements in each level was done using Saaty's comparison technique (Appendix A and Appendix C show the technique used to obtain a pairwise comparison matrix from the numerical values of criteria, sub-criteria and attributes). After computing the pairwise comparison matrices for each of the levels of the hierarchy in Table 1, normalization matrices (Saaty, 1980) were calculated. From the normalization matrix, a priority vector was calculated. The priority vector (eigenvector) determined the relative ranking of the alternatives for each criterion. To check for consistency the Saaty's procedure was followed (see Appendix B). Checking for consistency of measurement is the basis of objective measurement for the subjective evaluations (Moses and Farrow, 2008:276).

3. Results

The AHP algorithm involves linear modelling during the analysis of data. The advantage of this algorithm is that it does not need many historical datasets (Cavano and McCall, 1978:137).

The linear equation used is as follows:

$$\begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} a_{c_1} & b_{c_1} \\ a_{c_2} & b_{c_2} \\ a_{c_3} & b_{c_3} \end{bmatrix} = \begin{bmatrix} P_a & P_b \end{bmatrix} \quad (1)$$

where: vectors **a** and **b** are alternatives with respect to a specific objective, and c_1, c_2 and c_3 are the weights assigned to criterion 1, criterion 2 and criterion 3 respectively, and (a_{c_1} , b_{c_1}), (a_{c_2} , b_{c_2}), (a_{c_3} , b_{c_3}), are the priority of the alternatives **a** and **b** with respect to the three criteria. In our case ATutor and Moodle were represented by **a** and **b**; and c_1, c_2 and c_3 were used to depict the criteria: usability, deploy-ability, and maintainability.

The result for equation (1) gives the total priority for each e-learning system (see Appendix D for the detailed calculation). Table 3 shows the ranked results. Moodle ranks above ATutor for each of the scales used.

Table 3: Ranking

Total Priority	
ATutor	Moodle
0.468554988	0.531445012

Finally the consistency indices for each pairwise comparison were computed and were equal to approximately zero (see Appendix B and Appendix C which means that all the pairwise comparison judgments were consistent (Saaty, 1980). It concurs with Moses and Farrow (2008:286) who argue that the best approach to validate results for subjective judgments is to check consistency.

4. Discussion and conclusion

The objective of this study was to determine the suitability of the analytical hierarchy process algorithm in the evaluation of FOSS when the evaluators are mostly users with little technical IT experience. In the evaluation, qualitative methods (using participative observation and focus group interviews) were complimented by quantitative methods (questionnaires and AHP).

We have shown that AHP is suitable for evaluating software in a developing country where Information Technology experience is limited because of its simplicity, accuracy and flexibility in making a logical, consistent and informed decision.

Since the AHP algorithm involves linear modelling, contributing metrics values can in certain cases create a non-compensatory effect. This means that a system under evaluation might be ranked higher than another, even though one or more of its constituent criteria, sub-criteria, and attributes have lower weights than the system it is compared to. The use of the analytical network process algorithm (ANP) may remedy this effect as it takes the dependence of criteria, sub-criteria, and attributes at the same level of the hierarchy, into account.

AHP deals with crisp (real) values of evaluation judgments, but human reasoning is imprecise, uncertain and fuzzy (Mikhailov and Tsvetinov, 2004:23). Furthermore when the number of criteria considered increases, the number of pair-wise comparisons will increase geometrically.This can lead to inconsistencies or even that the AHP algorithm fails completely. Fuzzy AHP could address this problem and is proposed as a possible alternative method for imprecise problems or problems with more criteria. In a future study it will be investigated how ANP and Fuzzy AHP can complement the decision making of AHP.

The results of this study agree with the results from Graf and List (2005:165) who found in their study (using QWS) that Moodle is preferred above 36 FOSS e-learning systems. The stakeholders at OUT regarded usability as the most important criterion required for the successful implementation of an e-learning system (see Table 4).

In our study it has been shown that — with a well planned evaluation process, good data collection and analysis methods — both novice and technical users can be successfully involved in the evaluation of software. The results contribute to the theory and methodology of the evaluation of FOSS and can be used to design and develop a framework for the evaluation of FOSS for developing countries.

We feel that using AHP in a developing country is suitable because it simplifies a complex problem by breaking it up into smaller steps that can be understood by all. The steps of AHP includes: defining the goals and outcomes of the problem; decomposing the problem into a hierarchical structure of criteria, sub-criteria, attributes, and alternatives; computing pairwise comparisons; employing the Eigenvalue method to estimate relative weights; checking consistency and finally combining the relative weight to obtain the overall rating for the alternatives.

Deployability, maintainability and usability were the only criteria considered in this evaluation as they were assumed to be the most important criteria needed for the implementation of software in a developing country. Computer competency of technical users affects maintainability, bandwidth of networks affects deployability and the fact that the user interface is in a language foreign to the OUT user, affects its usability.

The subjective evaluation was consistent for the attributes identified for the criteria deployability; maintainability and usability. The evaluation was shown to be consistent since the computed consistency indices, for all pairwise comparisons, were equal to zero thus confirming that all stakeholders can participate in the evaluation process. Involving all stakeholders in the selection of software has the added advantage of greater acceptance of the system.

5. Appendix A: AHP computations

5.1 Pairwise comparison matrix

The first pairwise comparison matrix is obtained by comparison of criteria with respect to the objective.

$$\begin{array}{llll} Objective & criterion\ 1 & criterion\ 2 & criterion\ 3 \\ criterion\ 1 & \frac{c_1}{c_1} = a_{11} & \frac{c_1}{c_2} = a_{12} & \frac{c_1}{c_3} = a_{13} \\ criterion\ 2 & \frac{c_2}{c_1} = a_{21} & \frac{c_2}{c_2} = a_{22} & \frac{c_2}{c_2} = a_{23} \\ criterion\ 3 & \frac{c_3}{c_1} = a_{31} & \frac{c_3}{c_2} = a_{32} & \frac{c_3}{c_3} = a_{33} \end{array} \quad \text{(i)}$$

Then three pairwise comparison matrices are computed with respect to the three criteria. In our case **A** can stands for ATutor and **B** for Moodle.

Pairwise matrix for alternative **A** and **B** with respect to criterion 1

$$\begin{array}{lll} criterion\ 1 & A & B \\ A & \frac{a_{c_1}}{a_{c_1}} & \frac{a_{c_1}}{b_{c_1}} \\ B & \frac{b_{c_1}}{a_{c_1}} & \frac{b_{c_1}}{b_{c_1}} \end{array} \quad \text{(ii)}$$

Pairwise matrix for alternative **A** and **B** with respect to criterion 2

$$\begin{array}{lll} criterion\ 2 & A & B \\ A & \frac{a_{c_2}}{a_{c_2}} & \frac{a_{c_2}}{b_{c_2}} \\ B & \frac{b_{c_2}}{a_{c_2}} & \frac{b_{c_2}}{b_{c_2}} \end{array} \quad \text{(iii)}$$

Pairwise matrix for alternative **A** and **B** with respect to criterion 2

criterion 3	*A*	*B*
A	$\frac{a_{c_3}}{a_{c_3}}$	$\frac{a_{c_3}}{b_{c_3}}$
B	$\frac{b_{c_3}}{a_{c_3}}$	$\frac{b_{c_3}}{b_{c_3}}$

(iv)

5.2 Normalization matrix

After obtaining the pairwise comparison matrices, the next step is to normalise the matrices with respect to the weights, and then obtain the principal Eigenvector – which is the priority vector. We did this process for level 1, and then for the level 2. Then we computed the overall composite weight for the alternatives *A* and *B*, which is a normalization of weighted priority vectors.

Normalization is obtained first by computing the total of each column *j* in pairwise comparison matrix (Saaty, 1980) and then each row in a column *j* is divided by the column total($\sum_{i=1}^{n} a_{ij}$). Normalization is given by the following expression:

$$a'_{ij} = a_{ij} \big/ \sum_{i=1}^{n} a_{ij} \quad , i, j = 1, 2, \ldots, n \tag{v}$$

Objective	*criterion* 1	*criterion* 2	*criterion* 3
criterion 1	$\frac{a_{11}}{\sum_{i=1, j=1}^{3} a_{ij}}$	$\frac{a_{12}}{\sum_{i=1, j=2}^{3} a_{ij}}$	$\frac{a_{13}}{\sum_{i=1, j=3}^{3} a_{ij}}$
criterion 2	$\frac{a_{21}}{\sum_{i=1, j=1}^{3} a_{ij}}$	$\frac{a_{22}}{\sum_{i=1, j=2}^{3} a_{ij}}$	$\frac{a_{23}}{\sum_{i=1, j=3}^{3} a_{ij}}$
criterion 3	$\frac{a_{31}}{\sum_{i=1, j=1}^{3} a_{ij}}$	$\frac{a_{32}}{\sum_{i=1, j=2}^{3} a_{ij}}$	$\frac{a_{33}}{\sum_{i=1, j=3}^{3} a_{ij}}$

(vi)

After normalization, the sum of each column (j) of the matrix is equivalent to 1.

$$\sum_{i}^{n} a_{ij} = 1$$

Objective	*criterion* 1	*criterion* 2	*criterion* 3
criterion 1	$\frac{a_{11}}{a_{11} + a_{21} + a_{31}}$	$\frac{a_{12}}{a_{12} + a_{22} + a_{32}}$	$\frac{a_{13}}{a_{13} + a_{23} + a_{33}}$
criterion 2	$\frac{a_{21}}{a_{11} + a_{21} + a_{31}}$	$\frac{a_{22}}{a_{12} + a_{22} + a_{32}}$	$\frac{a_{23}}{a_{13} + a_{23} + a_{33}}$
criterion 3	$\frac{a_{31}}{a_{11} + a_{21} + a_{31}}$	$\frac{a_{32}}{a_{12} + a_{22} + a_{32}}$	$\frac{a_{33}}{a_{13} + a_{23} + a_{33}}$

(vii)

In order to get the weight vector (w_i) for the criterion 1, criterion 2 and criterion 3, we take the average of each row of the normalization matrix (assume the normalized matrix is a'_{ij}). Therefore

$$\frac{1}{n} . \sum_{i}^{n} a'_{ij} = w_i \tag{viii}$$

This process is done for the other matrices in order to obtain the weight vector for each element in the hierarchy.

It gave the weight vector for each pairwise matrix which represents the relative weights of each alternative p_j with respect to criterion *i* (*i*=1, 2, 3).

5.3 Computing global priorities

This is the step where by the relative importance of each element within the level (local priorities) is merged/multiplied with the relative importance of each element in the parent level. This gives the global priorities for each alternative.

The computation for each alternative *j* (p_j) is done in order to obtain the overall composite weight vector, which is given by $\sum_{i=1}^{n} z_{ij} \cdot w_i$. (ix)

where *w* is the weight vector or priority vector of the alternative software products and *z* is the weight vector of the criteria.

6. Appendix B: consistency checking

We implemented consistency checking as outlined by Saaty (1980). Checking consistency is done using the following mathematical formulae:

First procedure

Let ***A*** be a pairwise comparison matrix for *n* criteria. Let *w* be the weight vector computed for ***A***. Compute

$$\boldsymbol{A}w^T \tag{i}$$

Second procedure

Let $\lambda_{\max}$ be a maximum or principal Eigenvalue. Let *n* be the number of criteria. Compute the $\lambda_{\max}$ given that the number of criteria is *n*.

$$\lambda_{\max} = \frac{1}{n}\sum_{i=1}^{n}\frac{(Aw^T)_i}{(w^T)_i} \tag{ii}$$

Third procedure

Let *CI* be the consistency index. Compute *CI*

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

Fourth procedure

Let *CR* be the consistency ratio. Let *RI* be the random index.

If *CI* = 0 then ***A*** is consistent; otherwise if the *CR* = $\frac{CI}{RI} \le 0.10$ then ***A*** is consistent enough. If *CR* = $\frac{CI}{RI} \ge 0.10$ then ***A*** is seriously inconsistent; the Random index RI_n for any square matrix is given as a constant value by Saaty (1980).

Table 4: Random Index (adapted from Saaty, 1980)

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

7. Appendix C: example how AHP computations were done

To obtain priority/ weight vector for level 1: respondents were asked to indicate the level of importance of three criteria usability, deployability and maintainability (c_1, c_2, c_3) respectively. Then the mean of all the respondents were computed to obtain the input values for AHP. We got (c_1=9, c_2=7and c_3=7) using scale shown in Table 2. After that the equation (i) from Appendix A was applied and we obtained the following pairwise matrix:

1	1.285714286	1.285714286
0.777778	1	1
0.777778	1	1

Then we computed normalization matrix using equation (vi) from Appendix A and we obtained the following matrix:

0.391304	0.391304348	0.391304348
0.304348	0.304347826	0.304347826
0.304348	0.304347826	0.304347826

After that the priority vector was computed using equation (ix) from Appendix A and we obtained the following weights for usability, deployability and maintainability:

0.391304348
0.304347826
0.304347826

This answer is reflected in level 1 of appendix D.

Then we checked the consistency using the formulae shown in Appendix B.

Equation (i) in Appendix B gave the following result:

1.173913043
0.913043478
0.913043478

The equation (ii) in appendix B was applied and we got
=1/3*(1.173913043/0.391304348+0.913043478/0.304347826+0.913043478/0.304347826) = 3

After this computation, we calculated the *CI* using equation (iii) in Appendix B and the answer was

= (3-3)/2 = 0 (the number of criteria (*n*) was 3)

This indicated that the consistency index was equal to 0 (means consistent judgments).

These computations were done for all hierarchy levels and the results are shown in Appendix D.

8. Appendix D: priority weight obtained after using AHP

LEVEL 1	LEVEL 2		LEVEL 3			
					ATutor	Moodle
			ATutor	Moodle	Priority	Priority
		Weight	0.5000	0.5000	0.0114	0.0114
	operability	0.0685	0.5000	0.5000	0.0114	0.0114
			0.6250	0.3750	0.0143	0.0086
	usability compliance	0.0685	0.2000	0.8000	0.0137	0.0548
usability	complexity	0.0880	0.5833	0.4167	0.0514	0.0367
0.3913			0.5000	0.5000	0.0110	0.0110
	understandability	0.0880	0.5833	0.4167	0.0128	0.0092
			0.4167	0.5833	0.0092	0.0128

LEVEL 1	LEVEL 2		LEVEL 3			
					ATutor	Moodle
			ATutor	Moodle	Priority	Priority
			0.5833	0.4167	0.0128	0.0092
	learnability	0.0098	0.5000	0.5000	0.0016	0.0016
			0.6923	0.3077	0.0023	0.0010
			0.5000	0.5000	0.0016	0.0016
	attractiveness	0.0685	0.5000	0.5000	0.0043	0.0043
			0.5000	0.5000	0.0043	0.0043
			0.4167	0.5833	0.0036	0.0050
			0.4667	0.5333	0.0040	0.0046
			0.4667	0.5333	0.0040	0.0046
			0.5000	0.5000	0.0043	0.0043
			0.1429	0.8571	0.0012	0.0073
			0.5000	0.5000	0.0043	0.0043
	portability	0.0830	0.1667	0.8333	0.0069	0.0346
			0.6429	0.3571	0.0267	0.0148
deployability	installability	0.0646	0.4375	0.5625	0.0282	0.0363
0.3043	adaptability	0.0461	0.5000	0.5000	0.0115	0.0115
			0.5556	0.4444	0.0128	0.0102
	configurability	0.0646	0.4375	0.5625	0.0282	0.0363
	distributability	0.0461	0.4375	0.5625	0.0202	0.0259
	stability	0.0609	0.4375	0.5625	0.0266	0.0342
maintainability	analyzability	0.0261	0.6364	0.3636	0.0166	0.0095
0.3043			0.4375	0.5625	0.0063	0.0082
	changeability	0.0435	0.5000	0.5000	0.0072	0.0072
			0.4667	0.5333	0.0068	0.0077
	testability	0.0435	0.5000	0.5000	0.0072	0.0072
			0.5000	0.5000	0.0072	0.0072
			0.5000	0.5000	0.0072	0.0072
	trackability	0.0261	0.5000	0.5000	0.0130	0.0130
	flexibility	0.0435	0.5000	0.5000	0.0217	0.0217
	upgradeability	0.0609	0.5000	0.5000	0.0304	0.0304
Total					**0.4686**	**0.5314**

Acknowledgement

We would like to thank DAAD and UWC for financial support. Also the authors wish to thank the reviewers for their valuable comments and suggestions that contributed significantly to make this paper better.

References

Basili, V.R., Caldiera, G. and Romback, H.D. (1994) ‘Goal question metric paradigm’, In J.C. Marciniak (Eds.), *Encyclopedia of Software Engineering*, vol.1, pp. 528-532, New York: John Wiley & Sons.

Bertoa, M.F. and Vallecillo, A. (2002) ‘Quality attributes for COTS components’, in: *Proceedings of the Sixth ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering* (QAOOSE 2002), Malaga, Spain, pp. 54-66.

Boehm, B. W. (1981a) *Software engineering economics*. Prentice-Hall, NJ: Englewood Cliffs.

Boehm, B. W., Brown, J. R., and Lipow, M. (1976b) ‘Quantitative evaluation of software quality’. In *Proceedings of the 2nd international Conference on Software Engineering* (San Francisco, California, United States, October 13 - 15, 1976). International Conference on Software Engineering. IEEE Computer Society Press, Los Alamitos, CA, pp. 592-605.

Cavano, J. P. and McCall, J. A. (1978) 'A framework for the measurement of software quality'. in Jackson, S. and Lockett, J. A. (ed.), *Proceedings of the Software Quality Assurance Workshop on Functional and Performance Issues*, pp. 133-139.

Colace, F., DeSanto, M. and Vento, M. (2003) 'Evaluating on-line learning platforms: a case study'. In *Proceedings of the 36th Annual Hawaii international Conference on System Sciences (Hicss'03) - Track 5* (January 06 - 09, 2003). HICSS. IEEE Computer Society, Washington, DC, pp. 5, 154.

Coleman, D., Ash, D., Lowther, B. and Oman, P. (1994) 'Using metrics to evaluate software system maintainability'. *Computer,* vol. 27, no, 8, pp. 44-49.

Comella-Dorda, S., Dean, J. C., Morris, E. and Oberndorf, P. (2002) 'A process for COTS software product evaluation', *Proceedings of the First International Conference on COTS Based Software Systems*, Springer-Verlag LNCS 2255, pp. 86-96.

Fenton, N. E. and Pfleeger, S. L. (1997) *Software metrics - a rigorous and practical approach*, PWS Publishing Company.

Forman, E. and Peniwati, K. (1998) 'Aggregating individual judgments and priorities with the analytic hierarchy process', European Journal of Operational Research, Volume 108, Issue 1, pp. 165-169

Graf, S. and List, B. (2005) 'An evaluation of open source e-learning platforms stressing adaptation issues'. In *Proceedings of the Fifth IEEE international Conference on Advanced Learning Technologies* (July 05 - 08, 2005). ICALT. IEEE Computer Society, Washington, DC, pp. 163-165.

IEEE (1998). Std. 1061–1998 IEEE standard for a software quality metrics methodology.

ISO. (2001). ISO/IEC 9126-1, Software engineering - product quality - part 1: quality model, Geneva, International Organization for Standardization.

Isamuyo, Z. (2006) '*Open access ICT infrastructure in rural Tanzania: prototype design*, Masters Thesis', Royal Institute of Technology (KTH), Stockholm, Sweden.

Manieri, A., Begin, M.E., Meglio, A. Di. and Rippa, A. (2008) 'Automation in quality assurance environments', The *First International Conference on Open Source*, Qualipso Conference 2008, Rome, Italy

McHarazo, A.A.S. and Olden, A. (2000) 'Fulfilling the learning resource requirements of students at the Open University of Tanzania'. *Journal of Librarianship and Information Science*, vol.32, no.4, pp. 204-214.

Mikhailov, L. and Tsvetinov, P. (2004) 'Evaluation of services using a fuzzy analytic hierarchy process', *Applied Soft Computing*, vol.5, no.1, pp. 23-33

Moses, J. and Farrow, M. (2008) 'Tests for consistent measurement of external subjective software quality attributes', *Empirical Softw. Engg.,* vol.13, no.3, pp. 261-287.

Nichols, D.M. and Twidale, M.B. (2003) 'The usability of open source software'. First Monday, vol.8, no.1.

Ryu. Y.S. (2005) '*Development of questionnaires for electronic mobile products and decision making methods'*. PhD Thesis, Virginia Polytechnic Institute and State University.

Saaty, T.L. (1980). *The analytic hierarchy process: planning, priority setting, resource allocation,* New York: McGraw-Hill International.

Somerville, I. (2004) *Software engineering*, 6th edition (International Computer Science Series). Addison-Wesley.

Stevens, S.S. (1946) 'On the theory of scales of measurement', *Science*, vol.103, pp. 677-680.

Turban, E. (1993). *Decision Support and Expert Systems: Management Support System*, third edition, New York: Macmillan.

Van Vuuren, D. and Maree, A. (1999) 'Survey methods in market and media research: applied methods for the social sciences', in Terre Blanche, M. and Durrheim, K. (ed.), *Research in Practice*, Cape Town: UCT Press.

Wesson, J. (2003) 'Usability issues for e-commerce and e-learning: a developing country perspective', *Proceeding of the International workshop on Utility, Usability and Complexity of e-Information systems*, pp. 43-55