Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 11 No. 1, 2020 - www.cisdijournal.net

Evaluation of Adaptability Level of Similar Functional Object-Oriented Software

Udo, E.N, *Usip, P.U & Dan, E.A.
[1]Department of Computer Science
Faculty of Science
University of Uyo Uyo,
Akwa-Ibom State, Nigeria
*** Corresponding Author**: **E-mai**l- patienceusip@uniuyo.edu.ng

ABSTRACT

Adaptability, as one of the external quality attributes of software, is of great importance in software engineering community especially now that the environment in which software products operate is becoming highly unpredictable due to rapid changes in computing technologies and operating system requirements and should therefore be a major concern to software developers at the design stage. Many researchers have analyzed and evaluated the source codes of different object-oriented software products to establish the absence and/or the level of presence of external quality factors like usability, testability, reusability, reliability, functionality and maintainability but are yet to evaluate such software codes to reveal their adaptability level not to talk of evaluating such level in a similar functional object-oriented software using multi-criteria decision analysis tool like analytic hierarchy process. This work therefore analyses the source codes of three school management software that is basically handling the same functions, which the researchers were privy to be part of the team members that developed the software, to reveal their adaptability level using software analyzer. The results are evaluated and compared using three different approaches: Weighted Scoring Method (WSM), direct computation from software analyzer and Analytical Hierarchy Process (AHP) analysis. The multi-criteria decision strength of the three tools is also considered. Analysis from WSM and computation from software analyzer indicate software B as the one with higher level of adaptability while AHP indicates software C. This variation may be due to error level in the calculation of the metrics values by the software analyzer and the differences in source codes attributes such as number of classes and complexity level. Therefore, software B and C can be chosen when adaptability is to be considered as a software quality.

Keywords: Software Evaluation, Adaptability Level, Weighted Scoring Method, AHP

CISDI Journal Reference Format
Udo, E.N, Usip, P.U & Dan, E.A. (2020): Evaluation of Adaptability Level of Similar Functional Object-Oriented Software. Computing, Information Systems & Development Informatics Journal, Vol 11 No 1, Pp 85-96.
Available online at www.computing-infosystemsjournal.info

1. INTRODUCTION

One of the properties of software that is of great significance is adaptability [1]. Adaptability is defined as "the ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed" [2]. Adaptability should therefore be given adequate attention during software quality measurement, evaluation and predictions especially now that the environment in which software products operate becomes highly unpredictable due to rapid changes in hardware platform as well as changes in the operating system requirements.

The increasing demand and use of commercial-off-the-shelf (COTS) software products by small, medium and large scale organizations makes the evaluation of adaptability level of software an important task in software engineering. Development of COTS software products for use in different areas of human endeavours is now an interesting business and the software market is flooded with such software. Users of such software products are now faced with the problem of choice considering the quality of the available software. One of such quality concern is the software adaptability level.

Choosing the right software among several alternatives could be confusing especially when there are no appropriate software evaluation criteria to aid in the selection. Many individual software users and organizations use intuitive methods to select software products such as people's comment about the software, the software company's profile, key members of the software development team, physical attributes of the software product etc. A hurried and uneducated choice could lead to several various problems [3]. These problems may include, among others, unfulfilled required functionality, introduction of overhead costs on system integration and maintenance.

Choosing the correct software can only be achieved through predetermined evaluation and selection processes. In this work the adaptability level of three school management software (SMS) that is basically handling the same functions and management activities are evaluated. This is done by analyzing the internal quality attributes of the SMSs which directly affects adaptability: coupling, cohesion, inheritance and complexity [4]. The analysis is carried out by measuring the source code of the SMSs to reveal their adaptability level. The measurement is done using the software analyzer (AdaptAnalyzer) developed and applied in [5]. The results from AdaptAalyzer are evaluated and compared using three different approaches: Weighted Scoring Method (WSM), direct computation from AdaptAnalyzer and analysis using Analytical Hierarchy Process (AHP).

Weighted Scoring method is a tool that provides systematic steps for selecting solution or product among several alternatives based on many criteria. It is a technique for putting a semblance of objectivity into a subjective process. Using a consistent list of criteria, weighted according to the importance or priority of the criteria, a comparison of similar solutions or products can be completed. If numerical values are assigned to the criteria priorities and the ability of the product to meet a specific criterion, a "weighted" value can be derived. By summing the weighted values, the product with the highest weighted score will be selected.

AHP was proposed by [6] as one of the most practical tools used by decision makers and researchers in making decision based on multiple criteria. This tool aims at quantifying relative priorities for a given set of alternatives on a ratio scale, based on the judgment of the decision-maker, and stresses the importance of the intuitive judgments of a decision-maker as well as the consistency of the comparison of alternatives in the decision-making process [7].

2. RELATED WORKS

Various researchers have analyzed the source codes of different software products to investigate different quality factors or software attributes. Reference [8] carried out a survey on object-oriented software metrics. Reference [9] investigated the result of object oriented design software metrics on fault- proneness for java applications which were empirically analyzed and tested using software tool at the source code level. Reference [10] also analyzed the complexity of java programs, at source code level, using object oriented software metrics.

Reference [11] evaluated different object-oriented metrics, show how they affect reusability, flexibility, understandability, extensibility, effectiveness and functionality and also show, at the design level, how they relates to the quality of 6 different software projects. Reference [12] developed a model for assessing software testability using six internal and external software properties. Evaluation and selection of software packages based on some external quality attributes like usability, reliability, reusability and testability is a commonplace activity within the software engineering community. The evaluation of such software products uses Multi Criteria Decision Analysis (MCDA) tool such as WSM and AHP. Reference [13] used AHP to select the best ensemble method to predict software defect using public domain software defect datasets. Reference [14] also used AHP to select the best development method for software in Ministry of Foreign Affairs of the Republic of Indonesia due to frequent changes among IT team members. Reference [15] used AHP to evaluate software testability; [16] evaluated software usability using AHP and [17] used comprehensive weighted method and AHP techniques to analyze and calculate hardware testability; [18 - 19] evaluated reusability and testability assessment on aspect oriented software using AHP.

It is observed that researchers are yet to evaluate the source code of object-oriented software in order to predict their adaptability level and also evaluate similar functional software products to select the best in terms of adaptability, which is one of the most significant external software properties, using MCDA tool.Therefore this work analyzes the source code of three similar functional object oriented software using software analyzer and evaluate their adaptability level using WSM and AHP (MCDA tools) to select the best among the three. The software options from the tools: WSM, AdaptAnalyzer and AHP are compared.

3. MULTI-CRITERIA DECISION ANALYSIS

This is a framework that helps in decision making when multiple and conflicting objectives which are valued differently by different stakeholders from different perspective are involved. It describes approaches that take into account multiple criteria to aid in the selection of a particular product or action among several alternatives.

3.1Analysis Using Weighted Scoring Method (WSM)

As a form of multi-attribute or multi-criteria analysis, WSM is a method of scoring options or solutions against a prioritize requirements list to determine which option best fits the selection criteria. It involves identification of all attributes that are relevant to the project; the allocation of weights to each of the attribute to reflect their relative importance.

The attributes used in this work in order to determine the adaptability level of a given software project are given below and as defined by [20 - 21]

i. **Coupling** – It is a measure of the strength of association established by a connection from one entity to another. It shows class interconnectivity/relationship across project.
ii. **Cohesion** – The degree to which methods within a class are related to one another and work together to provide well-bounded behavior. That is internal consistency (methods interactions with data within a class).
iii. **Inheritance** – Inheritance is a type of relationship among classes that enables programmers to reuse previously defined objects including variables and operators. It shares attributes and operations among classes based on hierarchical relationship.
iv. **Complexity** – This describes the interactions between numbers of entities. As the number of entities increases, the number of interactions between them would increase exponentially, and it would get to a point where it would be impossible to know and understand all of them. Number of methods within a program influences the modification of the software

The sub-attributes for each of the attributes used in this work are shown in Table 1. The AdaptAnalyzer will also calculates values of these sub- attributes for each of the SMSs using the object-oriented metrics suite of [20] which are coupling between objects (CBO), response for a class (RFC), lack of cohesion in methods 1 (LCOM1), depth of inheritance tree (DIT), number of children(NOC) and weighted method per class (WMC). It also makes use of two other metrics by [22] which are number of methods (NOM) and lack of cohesion in methods 2 (LCOM 2).

Table 1 –Sub-attributes for the attributes

S/N	ATTRIBUTES	SUB-ATTRIBUTES
1.	Coupling	Excessive Coupling
		Independency of a Class
		Number of Couples
		Lesser Number of Methods invoked from a Class
2.	Cohesion	High Cohesiveness of Methods within a Class
		Low Disparateness of Methods
		Decreased Encapsulation (Low Value of LCOM2)
		Increased Complexity (High Value of LCOM2)
3.	Inheritance	Lower Classes Depth in the Hierarchy
		Greater Number of Children
4	Complexity	Larger Number of Methods/Complexity
		Number of Methods Declared
		Number of Methods not Declared
		Larger Number of Classes

Steps used for the calculation of the Weighted Score (WS) are as follows:

i. **Assignment of Priority Values** - Priority values were assigned to the requirements (sub-attributes) in Table 1 with their importance identified by those values. The conventional use of values 0, 1, 3 and 5 are interpreted as follows:

0 - Requirement does not apply
1 - Requirement is of low importance
3 - Requirement must be met
5 - Requirement is of high importance

ii. **Assignment of Scores** - Scores are assigned to each school management software product based on how the software product meets the requirement independently. The conventional scoring values used are 0, 2, 4 and 6 which depicts:

0 - The product does not have the requirement
2 - The product meets some aspect of the requirement but not all
4 - The product meets the requirement
6 - The product exceeds the requirement

iii. **Calculation of Weighted Score** (WS) –This is the product of priority value and score.

$$WS = PV \times AS \qquad (1)$$

where PV is priority value and AS is attribute score.
Table 2 shows the priority values, scores and the WS for each of the School Management Software (SMS).

Table 2 – Priority values, scores and weighted scores

S/N	Requirements	Priority	SMS A		SMS B		SMS C	
			Score	WS	Score	WS	Score	WS
1	Excessive Coupling	0	4	0	4	0	4	0
2	Independency of a Class	3	4	12	4	12	4	12
3	Number of Couples	1	6	6	6	6	6	6
4	Lesser Number of Methods invoked from a Class	5	2	10	2	10	2	10
5	High Cohesiveness of Methods within a Class	5	0	0	4	20	0	0
6	Low Disparateness of Methods	1	2	2	2	2	4	4
7	Decreased Encapsulation (High Value of LCOM2)	1	6	6	6	6	2	2
8	Increased Complexity (High Value of LCOM2)	1	6	6	6	6	2	2
9	Lower Classes Depth in the Hierarchy	5	6	30	6	30	6	30
10	Greater Number of Children	5	4	20	4	20	4	20
11	Larger Number of Methods/Complexity	0	4	0	6	0	6	0
12	Number of Methods Declared	5	4	20	6	30	6	30
13	Number of Methods not Declared	3	2	6	4	12	2	6
14	Larger Number of Classes	0	4	0	4	0	4	0
	TOTAL			**118**		**154**		**122**

iv. **Summation of WS** – The adaptability level (A_L) of the SMS is gotten by adding together the weighted scores of all the requirements and the SMS with the largest WS is the one with the higher adaptability level.

$$A_L = \sum_{r=i}^{n} WS, \quad \forall r > 1 \tag{2}$$

where r is the requirement and n is the number of requirements .

3.2 Analysis using AdaptAnalyzer

Scores were calculated directly from the AdaptAnalyzer rather than assigning scores based on how the software meets the requirement. The scores for each of the requirement were calculated using the formula in Table 3.

Table 3 – Calculation of scores using AdaptAnalyzer's values

S/N	Requirements	Formular for Calculation of Score
1	Excessive Coupling	Average CBO x Number of Classes
2	Independency of a Class	Number of Classes / Average CBO
3	Number of Couples	Average CBO x Number of Classes
4	Lesser Number of Methods invoked from a Class	Average RFC
5	High Cohesiveness of Methods within a Class	Average LCOM1
6	Low Disparateness of Methods	Average LCOM1 x Number of Methods
7	Decreased Encapsulation (High Value of LCOM2)	Average LCOM2 x Number of Classes
8	Increased Complexity (High Value of LCOM2)	Average LCOM2 x Number of Classes
9	Lower Classes Depth in the Hierarchy	Average DIT x Number of Classes
10	Greater Number of Children	Average NOC x Number of Classes
11	Larger Number of Methods/Complexity	Average WMC x Number of Classes
12	Number of Methods Declared	Average NOM x Number of Classes
13	Number of Methods not Declared	Result of 11 – Result of 12
14	Larger Number of Classes	Number of Classes

Applying the formula yielded the scores; AdaptAnalyser's Score (AAS) in Table 4.

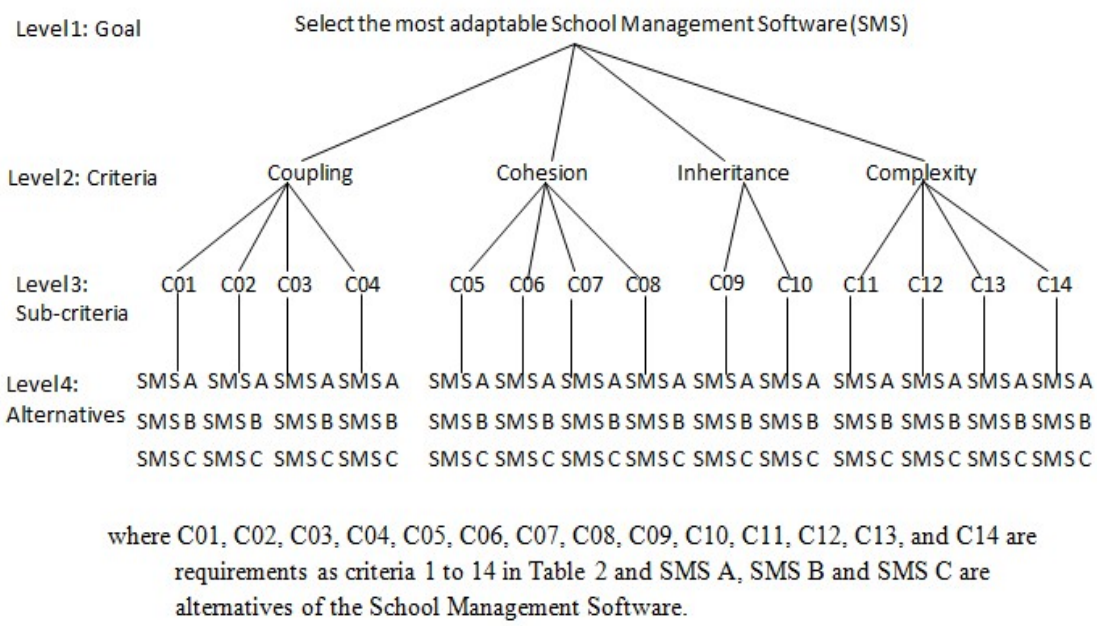
Table 4 – Scores calculated from the Metrics Values of the Analyzer's Results

S/N	Requirements	Priority	SMS A		SMS B		SMS C	
			AAS	W.S	AAS	W.S	AAS	W.S
1	Excessive Coupling	0	21	0	35	0	22	0
2	Independency of a Class	3	35	105	14	42	35	105
3	Number of Couples	1	21.	21	35	35	22	22
4	Lesser Number of Methods invoked from a Class	5	5	25	13	65	9	45
5	High Cohesiveness of Methods within a Class	5	0.04	0.20	0.09	0.45	0.02	0.10
6	Low Disparateness of Methods	1	0.40	0.40	6.47	6.47	0.01	0.01
7	Decreased Encapsulation (High Value of LCOM2)	1	4.05	4.05	69.08	69.08	10.08	10.08
8	Increased Complexity (High Value of LCOM2)	1	4.05	4.05	69.08	69.08	10.08	10.08
9	Lower Classes Depth in the Hierarchy	5	19	95	36	180	50	250
10	Greater Number of Children	5	1	5	1	5	1	5
11	Larger Number of Methods/Complexity	0	10	0	81	0	46	0
12	Number of Methods Declared	5	10	50	72	360	30	150
13	Number of Methods not Declared	3	2	6	9	27	16	48
14	Larger Number of Classes	0	27	0	22	0	28	0
	TOTAL			**315.70**		**859.08**		**645.27**

3.3 Analysis using Analytic Hierarchy Process (AHP)

The same school management software A, B and C considered in the approaches in section 3.1 and 3.2 are also used here. The assigned scores in table 2 are also used in the AHP analysis. We use AHP to decide which of the three school management software to adopt based on the desired attributes. The alternatives are then weighed and prioritized. The priority values, scores and the WS for each of the School Management Software (SMS) in Table 2 are adopted in the first step of the AHP. The hierarchy of attributes of SMS and adaptability level is given in figure 1.

The top level of the diagram shows the overall goal of the hierarchy, "Select the most adaptable School Management Software". The second level lists the attributes/criteria each of the three SMS should have. The third level gives sub-attributes/criteria for each of the attributes in the second level. The fourth level gives the three software management software for each sub-criterion.



where C01, C02, C03, C04, C05, C06, C07, C08, C09, C10, C11, C12, C13, and C14 are requirements as criteria 1 to 14 in Table 2 and SMS A, SMS B and SMS C are alternatives of the School Management Software.

Figure 1: Hierarchy of attributes and adaptability level of SMS

We go forward to set up the paired comparison of each of the criteria and SMS choices within each criterion. An automation of the paired comparison was done using MATLAB. The resulting pairwise matrices for criteria C01 to C14 are given in Table 5 to 18 respectively.

This is followed by the pairwise comparison for the decision alternatives shown in Tables 19 to 21 with their overall priority for each of the alternatives. In addition to the pairwise comparison for the decision alternatives, the same pairwise comparison procedure is also used to set priorities for all 14 sub-criteria in terms of importance of each in contributing the overall goal. Overall priority for each alternative is given by the sum of all priority vectors for each criteria in that alternatives multiplied by the priority vectors for the criteria by criteria matrix of that alternative. The school management software with the highest/biggest value of overall priority is the best software.

Table 5: Criteria: C01 – Excessive Coupling

	A	B	C	PV
A	1.0000	1.6667	1.0476	1.1744
B	0.6000	1.0000	0.6286	0.7046
C	0.9545	1.5909	1.0000	1.1210

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 6: Criteria: C02 – Independency of a Class

	A	B	C	PV
A	1.0000	0.4000	1.0000	0.6667
B	2.5000	1.0000	2.5000	1.6667
C	1.0000	0.4000	1.0000	0.6667

Table 7: Criteria: C03 – Number of Couples

	A	B	C	PV
A	1.0000	1.6667	1.0476	1.1744
B	0.6000	1.0000	0.6286	0.7046
C	0.9545	1.5909	1.0000	1.1210

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 8: Criteria: C04 – Lesser Number of Methods Invoked from a Class

	A	B	C	PV
A	1.0000	2.6000	1.8000	1.5463
B	0.3846	1.0000	0.6923	0.5947
C	0.5556	1.4444	1.0000	0.8590

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 9: Criteria: C05 – High Cohesiveness of Methods within a Class

	A	B	C	PV
A	1.0000	2.2500	0.5000	0.8710
B	0.4444	1.0000	0.2222	0.3871
C	2.0000	4.5000	1.0000	1.7419

λ_{max}: 3 CI:-4.440892e-16 RI: 0.58 CR:-7.656711e-16

Table10: Criteria: C06 – Low Disparateness of Methods

	A	B	C	PV
A	1.0000	16.1750	0.0250	0.0731
B	0.0618	1.0000	0.0015	0.0045
C	40.0000	647.0000	1.0000	2.9224

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 11: Criteria: C07 – Decreased Encapsulation (High Value of LCOM2)

	A	B	C	PV
A	1.0000	17.0568	2.4889	2.0542
B	0.0586	1.0000	0.1459	0.1204
C	0.4018	6.8532	1.0000	0.8254

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 12: Criteria: C08 – Increased Complexity (High Value of LCOM2)

	A	B	C	PV
A	1.0000	17.0568	2.4889	2.0542
B	0.0586	1.0000	0.1459	0.1204
C	0.4018	6.8532	1.0000	0.8254

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 13: Criteria: C09 – Lower Classes Depth in the Hierarchy

	A	B	C	PV
A	1.0000	1.8947	2.6316	1.5725
B	0.5278	1.0000	1.3889	0.8299
C	0.3800	0.7200	1.0000	0.5976

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 14: Criteria: C10 – Greater Number of Children

	A	B	C	PV
A	1	1	1	1
B	1	1	1	1
C	1	1	1	1

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 15: Criteria: C11 – Larger Number of Methods/Complexity

	A	B	C	PV
A	1.0000	8.1000	4.6000	2.2374
B	0.1235	1.0000	0.5679	0.2762
C	0.2174	1.7609	1.0000	0.4864

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 16: Criteria: C12 – Number of Methods Declared

	A	B	C	PV
A	1.0000	7.2000	3.0000	2.0377
B	0.1389	1.0000	0.4167	0.2830
C	0.3333	2.4000	1.0000	0.6792

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 17: Criteria: C13 – Number of Methods not declared

	A	B	C	PV
A	1.0000	450.0000	800.0000	2.9896
B	0.0022	1.0000	1.7778	0.0066
C	0.0013	0.5625	1.0000	0.0037

λ_{max}: 3 CI: 0 RI: 0.58 CR: 0

Table 18: Criteria: C14 – Larger Number of Classes

	A	B	C	PV
A	1.0000	0.8148	1.0370	0.9400
B	1.2273	1.0000	1.2727	1.1536
C	0.9643	0.7857	1.0000	0.9064

λ_{max}: 3 CI: 0 RI: 5.800000e-01 CR: 0

Figure 2: Pairwise Comparison Matrix for the Sub-Criteria and Consistency Metrics

Table 5 - Comparison Matrix of SMS A within Each Sub-criterion with Overall Priority of the Alternatives

C01	C02	C03	C04	C05	C06	C07	C08	C09	C10	C11	C12	C13	C14	PV
1.000	1.667	1.000	0.238	0.002	0.019	0.193	0.193	0.905	0.048	0.476	0.476	0.095	1.286	0.0016
0.600	1.000	0.600	0.143	0.001	0.011	0.116	0.116	0.543	0.029	0.286	0.286	0.057	0.771	0.0009
1.000	1.667	1.000	0.238	0.002	0.019	0.193	0.193	0.905	0.048	0.476	0.476	0.095	1.286	0.0016
4.200	7.000	4.200	1.000	0.008	0.080	0.810	0.810	3.800	0.200	2.000	2.000	0.400	5.400	0.0066
525.000	875.000	525.000	125.000	1.000	10.000	101.250	101.250	475.000	25.000	250.000	250.000	50.000	675.000	0.8304
52.500	87.500	52.500	12.500	0.100	1.000	10.125	10.125	47.500	2.500	25.000	25.000	5.000	67.500	0.0830
5.185	8.642	5.185	1.235	0.010	0.099	1.000	1.000	4.691	0.247	2.469	2.469	0.494	6.667	0.0082
5.185	8.642	5.185	1.235	0.010	0.099	1.000	1.000	4.691	0.247	2.469	2.469	0.494	6.667	0.0082
1.105	1.842	1.105	0.263	0.002	0.021	0.213	0.213	1.000	0.053	0.526	0.526	0.105	1.421	0.0017
21.000	35.000	21.000	5.000	0.040	0.400	4.050	4.050	19.000	1.000	10.000	10.000	2.000	27.000	0.0332
2.100	3.500	2.100	0.500	0.004	0.040	0.405	0.405	1.900	0.100	1.000	1.000	0.200	2.700	0.0033
2.100	3.500	2.100	0.500	0.004	0.040	0.405	0.405	1.900	0.100	1.000	1.000	0.200	2.700	0.0033
10.500	17.500	10.500	2.500	0.020	0.200	2.025	2.025	9.500	0.500	5.000	5.000	1.000	13.500	0.0166
0.778	1.296	0.778	0.185	0.001	0.015	0.150	0.150	0.704	0.037	0.370	0.370	0.074	1.000	0.0012
632.253	1053.756	632.253	150.537	1.204	12.043	121.935	121.935	572.039	30.107	301.073	301.073	60.215	812.897	

Table 6 – Comparison Matrix of SMS B within Each Sub-criterion with Overall Priority of the Alternatives

C01	C02	C03	C04	C05	C06	C07	C08	C09	C10	C11	C12	C13	C14	PV
1.000	0.400	1.000	0.371	0.003	0.185	1.974	1.974	1.029	0.029	2.314	2.057	0.257	0.629	0.0021
2.500	1.000	2.500	0.929	0.006	0.462	4.934	4.934	2.571	0.071	5.786	5.143	0.643	1.571	0.0052
1.000	0.400	1.000	0.371	0.003	0.185	1.974	1.974	1.029	0.029	2.314	2.057	0.257	0.629	0.0021
2.692	1.077	2.692	1.000	0.007	0.498	5.314	5.314	2.769	0.077	6.231	5.538	0.692	1.692	0.0056
388.889	155.556	388.889	144.444	1.000	71.889	767.556	767.556	400.000	11.111	900.000	800.000	100.000	244.444	0.8118
5.410	2.164	5.410	2.009	0.014	1.000	10.677	10.677	5.564	0.155	12.519	11.128	1.391	3.400	0.0113
0.507	0.203	0.507	0.188	0.001	0.094	1.000	1.000	0.521	0.014	1.173	1.042	0.130	0.318	0.0011
0.507	0.203	0.507	0.188	0.001	0.094	1.000	1.000	0.521	0.014	1.173	1.042	0.130	0.318	0.0011
0.972	0.389	0.972	0.361	0.003	0.180	1.919	1.919	1.000	0.028	2.250	2.000	0.250	0.611	0.0020
35.000	14.000	35.000	13.000	0.090	6.470	69.080	69.080	36.000	1.000	81.000	72.000	9.000	22.000	0.0731
0.432	0.173	0.432	0.160	0.001	0.080	0.853	0.853	0.444	0.012	1.000	0.889	0.111	0.272	0.0009
0.486	0.194	0.486	0.181	0.001	0.090	0.959	0.959	0.500	0.014	1.125	1.000	0.125	0.306	0.0010
3.889	1.556	3.889	1.444	0.010	0.719	7.676	7.676	4.000	0.111	9.000	8.000	1.000	2.444	0.0081
1.591	0.636	1.591	0.591	0.004	0.294	3.140	3.140	1.636	0.045	3.682	3.273	0.409	1.000	0.0033
444.874	177.950	444.874	165.239	1.144	82.238	878.055	878.055	457.585	12.711	1029.566	915.170	114.396	279.635	

Table 7 - Comparison Matrix SMS C within Each Sub-criterion with Overall Priority of the Alternatives

C01	C02	C03	C04	C05	C06	C07	C08	C09	C10	C11	C12	C13	C14	PV
1.000	1.591	1.000	0.409	0.001	0.000	0.458	0.458	2.273	0.045	2.091	1.364	0.727	1.273	0.0003
0.629	1.000	0.629	0.257	0.001	0.000	0.288	0.288	1.429	0.029	1.314	0.857	0.457	0.800	0.0002
1.000	1.591	1.000	0.409	0.001	0.000	0.458	0.458	2.273	0.045	2.091	1.364	0.727	1.273	0.0003
2.444	3.889	2.444	1.000	0.002	0.001	1.120	1.120	5.556	0.111	5.111	3.333	1.778	3.111	0.0007
1100.000	1750.000	1100.000	450.000	1.000	0.500	504.000	504.000	2500.000	50.000	2300.000	1500.000	800.000	1400.000	0.3298
2200.000	3500.000	2200.000	900.000	2.000	1.000	1008.000	1008.000	5000.000	100.000	4600.000	3000.000	1600.000	2800.000	0.6596
2.183	3.472	2.183	0.893	0.002	0.001	1.000	1.000	4.960	0.099	4.563	2.976	1.587	2.778	0.0007
2.183	3.472	2.183	0.893	0.002	0.001	1.000	1.000	4.960	0.099	4.563	2.976	1.587	2.778	0.0007
0.440	0.700	0.440	0.180	0.000	0.000	0.202	0.202	1.000	0.020	0.920	0.600	0.320	0.560	0.0001
22.000	35.000	22.000	9.000	0.020	0.010	10.080	10.080	50.000	1.000	46.000	30.000	16.000	28.000	0.0066
0.478	0.761	0.478	0.196	0.000	0.000	0.219	0.219	1.087	0.022	1.000	0.652	0.348	0.609	0.0001
0.733	1.167	0.733	0.300	0.001	0.000	0.336	0.336	1.667	0.033	1.533	1.000	0.533	0.933	0.0002
1.375	2.188	1.375	0.563	0.001	0.001	0.630	0.630	3.125	0.063	2.875	1.875	1.000	1.750	0.0004
0.786	1.250	0.786	0.321	0.001	0.000	0.360	0.360	1.786	0.036	1.643	1.071	0.571	1.000	0.0002
3335.250	5306.080	3335.250	1364.421	3.032	1.516	1528.151	1528.151	7580.115	151.602	6973.705	4548.069	2425.637	4244.864	

4. DISCUSSION OF RESULTS

From the MCDA using WSM, it is seen from Table 2 that SMS B has the highest A_L since the sum of its WS is 154 when compared to that of SMS A and SMS C which are 118 and 122 respectively. The scores derived from the AdaptAnalyzer, as evidenced in Table 4, also indicate that SMS B has the highest A_L with a value of 859.08 as against 645.27 and 315.70 for SMS C and A respectively. On the contrary, the result from AHP indicates that SMS C with overall priority of 0.8372 performs better that SMS A and B with overall priority of 0.2928 and 0.1362 respectively.

Overall Priority of SMS A = (0.3915*0.0016) + (0.2222*0.0009) + (0.3915*0.0016) + (0.5154*0.0066) +
(0.2903*0.8304) + (0.0244*0.0830) + (0.6847*0.0082) + (0.6847*0.0082) +
(0.5242*0.0017) + (0.3333*0.0332) + (0.7458*0.0033) + (0.6792*0.0033) +
(0.9965*0.0166) + (0.3133*0.0012)
= 0.2928

Overall Priority of SMS B = (0.2349*0.0021) + (0.5556*0.0052) + (0.2349*0.0021) + (0.1982*0.0056) +
(0.129*0.8118) + (0.0015*0.0113) + (0.0401*0.0011) + (0.0401*0.0011) +
(0.2766*0.0020) + (0.3333*0.0731) + (0.0921*0.0009) + (0.0943*0.0010) +
(0.0022*0.0081) + (0.3845*0.0031)
= 0.1362

Overall Priority of SMS C = (0.3737*0.0003) + (0.2222*0.0002) + (0.3737*0.0003) + (0.2863*0.0007) +
(0.5806*0.3298) + (0.9741*0.6596) + (0.2751*0.0007) + (0.2751*0.0007) +
(0.1992*0.0001) + (0.3333*0.0066) + (0.1621*0.0001) + (0.2264*0.0002) +
(0.0012*0.0004) + (0.3021*0.0002)
= 0.8372

Table 8 shows the comparative analysis of the school management software using the Weighted Sum Method, AdaptAnalyzer and the Analytic Hierarchy Process.

Table 8: Comparative Analysis of School Management Software

	WSM	ADAPTANALYZER	AHP
SMS A	118	645.27	0.2928
SMS B	154	859.08	0.1362
SMS C	122	315.70	0.8372

5. CONCLUSION

In this work, the adaptability levels of three similar functional object-oriented software is evaluated using WSM, computation using AdaptAnalyzer and AHP. The results of the analysis using the three approaches have indicated variances in the level of adaptability. This is due to differences in source codes similarity between the three SMS in terms of number of classes and complexity level as well as the strength of the tools in checking inconsistencies. AHP has a greater ability to check inconsistencies in the criteria because Eigen value is employed to assess the strength of the consistency ratio of the comparative matrix. With this in mind, one can be informed why the result of the analysis using AHP is considering a different SMS from other approaches. From this comparison, SMS C is considered the best as indicated by the AHP analysis while SMS B is the good alternative as seen from the result of WSM analysis and computation from AdaptAnalyzer. Therefore, we conclude that SMS B or C can be selected when considering their adaptability level as a quality factor in choosing among the three. The work can be furthered by comparing the A_L of the three SMSs using fuzzy AHP and other hybrid knowledge-based systems.

REFERENCES

1. Subramanian, N. and L. Chung., "Software Architecture Adaptability: An NFR Approach", in Proceedings of the 4th International Workshop on Principles of Software Evolution, Vienna, Austria, 2001, pp 52 – 61.
2. IEEE "IEEE Standard Glossary of Software Engineering Terminology", Institute of Electrical and Electronics Engineers, New York, 1990, pp 1-84
3. VanStadeen, P and Lubbe, S., "A case Study on the Selection and Evaluation of Software for an Internet Organization", The Electronic Journal of Business Research Methods, Vol 4 No 1, 2006, pp 57-66.
4. Akwukwuma, V.V.N. and Udo, E, N. 2015. "Predicting Adaptability Level of Object-Oriented Software Using Metrics and Threshold Values", Pacific Journal of Science and Technology, Vol 16 No 2, 2015, pp 124-134.
5. Udo, E, N. and Akwukwuma, V.V. N., "Object Oriented Software Quality Assessment using Metrics Threshold Values", World Journal of Applied Science and Technology, Vol 7 No 2, 2015, pp 191 – 199
6. Satty, T. L., 2008, "Decision Making with the Analytic Hierarchy Process", International Journal of Services Science, Vol 1, No 1, 2008, pp 83 – 99.
7. Saaty, T. L., "The Analytic Hierarchy Process", McGraw Hill, New York, 1990
8. Shaik, A., C. Reddy, B. Manda, C. Prakashini, and K. Deepthi, "Metrics for Object Oriented Design Software Systems: A Survey", Journal of Emerging Trends and Applied Sciences (JETEAS) Vol 1 No 2, 2010, pp 190 – 198.
9. Amjan, S., N. Satyanarayana, M. Huzaifa, N. Shaik, M. Naveed, S. Rao, and C. Reddy, "Investigating the Result of Object Oriented Design Software Metrics on Fault Proness in Object Oriented Systems: A Case Study", Journal of Emerging Trends in Computing and Information Sciences.. Vol 2 No 4, 2011, pp 201 – 208.
10. Chhikara, A. and R.S. Chhillar. 2012. "Analyzing the Complexity of Java Programs using Object-Oriented Software Metrics", International Journal of Computer Science Issues, Vol 9, No1, 2012, pp 364 – 372.
11. Chauhan, N. and Gupta, M., "Evaluation of Metrics and Assessment of Quality of Object-Oriented Software", in proceedings of 3rd International Conference of System Modeling and Advancement in Research Trends (SMART), Teerthanker Mahaveer University , Moradabad, , 2014, 273 -279.
12. Singhani, H. and Suri, P. "Testability Assessment of Object-Oriented Software using Internal and external Factor Model and Analytic Hierarchy Process", International Journal of Scientific and Engineering Research, Vol 6 No 6, 2015, pp 1 – 10.
13. Peng, Y., Kon, G., Wang, G., and Wu, W., "Ensemble of Software Defect predictors: An AHP-Based Evaluation method", International Journal of Information Technology and Decision Making, Vol 10 No 1, 2011, pp 187- 206
14. Helingo, M., Puwandari, B., Ririsatria and Solichah, I., "The use of Analytic Hierarchy Process for Software Development Method Selection: A Perspective of e-Government in Indonesia", in proceedings of 4th Information Systems International Conference, ISICO 2017, Bali, 2017, pp 405 – 414
15. Khanna, P., "Testability of Object -Oriented Systems: An AHP-Based Approach for Prioritization of Metrics",in Proceedings of 2nd International Conference on Contemporary Computing and Informatics (IC3I), Amity University, Noida, Indai, 2016, pp 273 –281.
16. Dubey, S., Mittal A. and Rana, A., "Measurement of Object Oriented Software Usability using Fuzzy AHP", International Journal of Computer Science and Telecommunications, Vol 3 No 5, 2012, pp 98– 104.
17. Yang, C., Zheng, Y., Zhu, M., Zuo, Z., Chen, X. and peng, X., "A Testability Allocation Method Based on Analytic Hierarchy Process and Comprehensive Weighted Method", in IEEE 9th Conference on Industrial Electronics and Applications (ICIEA.), Hangzhou, China, June 2014, pp 113 – 116.
18. Singh, P., Sangwan, O., Pratap, A. and Singh, A., "Testability Assessment of Aspect Oriented Software Using Multi-criteria Decision Making Approaches", World Applied Sciences Journal, Vol 32 No 4, 2014, pp 718 -730.

19. Singh, P., Sangwan, O., Pratap, A. and Singh, A.,"Quantitative Evaluation of Reusability for Aspect Oriented Software using Multi-criteria Decision Making Approach", World Applied Sciences Journal, Vol 30 No 12, 2014, pp 1966–1976.
20. Chidamber, S. and Kemerer, C., "A Metrics Suite for Object Oriented Design," IEEE Transactions on Software Engineering. Vol 20 No 6, 1994, pp 476 – 493.
21. Fenton, N. and S. Pfleeger, "Software Metrics, a Rigorous and Practical Approach," International Thompson Computer Press, London, UK, 1996
22. Bansiya, J. and Davies, C. G., 2002. "A Hierarchical Model for Object-Oriented Design Quality Assessment," IEEE Transactions on Software Engineering, Vol 28 No 1, 2002, pp 4 – 17.