Journal of the Nigerian Association of Mathematical Physics
***Volume*20 (March, 2012), pp451 – 460**

Prioritization of Programmer's Productivity Using Analytic Hierarchy Process (AHP) Model

[1]Nwelih, E.and[2]Chiemeke, S.C.

Department of Computer Science, Faculty of Physical Sciences, University of Benin, Benin City Nigeria

Abstract

This paper focuses on the application of Analytic Hierarchy Process (AHP) model in the context of prioritizing programmer's productivity in University of Benin, Benin City Nigeria. This is achieved by evaluating the way in which the AHP model can be used to select the best programmer for the purpose of developing software and meeting the required time allocated.

Keywords: Analytic Hierarchy process, AHP, Prioritizing Programmer's Productivity, Cost Driver, Performance Measurement.

1.0Introduction

The economic and industrial development of any nation depends to a great extent, on its productivity level, which is a function of sound management practice in the areas of acquisition, utilization, evaluation and disposal of human resources. It is observed in [1] that the acquisition; development and utilization of Human resources are always critical to the economic development of any nation. Human resources are particularly important because they influence the efficiency and effectiveness of the use of other resources (i.e. money, material and time). People design and produce goods and services, control quality, market the products, allocate financial resources and set overall strategies and objectives for the organization. Programming productivity refers to a variety of software development issues and methodologies affecting the quantity and quality of code produced by an individual or team. The relative importance of programming productivity has waxed and waned along with other industry factors, such as:

The relative costs of manpower versus machine

The size and complexity of the systems being built

Highly publicized projects that suffered from delays or quality problems

Development of new technologies and methods intended to address productivity issues

Quality management techniques and standards

A need for greater programmer productivity was the impetus for categorical shifts in programming paradigms. The evaluation of IT professionals, based on non-quantitative consideration (as analysis capabilities, analyst experience, etc.) can be done by prioritization, which is the state of being first in position. This will help the judgment of management in selecting the right person for the job, as stressed by Taylor [2]. Many factors, such as the use of the software tools and methodologies, affect programmer productivity. However, one of the biggest differences in software development performance is between individuals. As early as 1968 a comparison of experienced professional programmers working on

[1]Corresponding author:-E-mail:emmanuelnwelih@yahoo.com, Tel.: +2348023396466

the same programming task found a ratio, in one case, of 1:25 between the shortest and longest time to code the program and, more significantly perhaps, of 1:28 for the time taken to debug it [3].
Chauncey [3] found that the most important influence on programmer productivity seemed to be experience. However, one of the key skills required of a project manager is the ability to produce software development project. If the wrong team members are selected, then the user's requirement will not be satisfied, the software will be delivered late and thereby causing failure. Selection of an appropriate human resource requires consideration of a number of criteria such as type of application, experience, technical requirement, time etc. A project manager needs to choose the best one in a set of competing alternatives that are evaluated under conflicting criteria. This paper illustrates the applicability of using the AHP for selecting the best programmer during any software development project in University of Benin, Benin City, Nigeria. Section two provides an overview of the theoretical concept behind AHP model. In section three, an illustrative application of AHP model in selecting the best programmer is discussed. The paper will conclude with some final remarks on the suitability of using the AHP for programmer's prioritization problem.

2.0 MATERIALS AND METHODS

The Analytic Hierarchy Process (AHP) was developed in the early 1970's by Thomas Saaty to solve prioritization problems. Saaty claims that the AHP serves as a framework for people to structure their own problems and provide judgments based on knowledge, reasons or feelings to derive a set or priorities considered as an optional solution to a decision problem [4, 5].
AHP is one of the best-known decisions making process to help people into hard task of making the best decisions out of set of possible options. Therefore AHP is considered a multi-criteria decision-making process that balances both qualitative and quantitative aspects. AHP is a versatile and powerful decision support tool that allows the user to design a hierarchical structure for decision making and weights the trade-offs between decision criteria and alternatives [6]. A major strength of the AHP the pairwise comparison in which the influence of elements of a particular level over those of a lower level is measured. In pairwise comparisons each criterion is compared one pair at a time in order to construct a matrix of those comparisons. Because two elements of the same level are compared at any given time to a particular element at a higher level, this method simplifies the process to a large extent. [7]. A ratio of relative importance is assigned to each paired comparison, usually according to saaty linear nine-point scale namely $^1/_9$, $^1/_8$, $^1/_7$, … .., 7, 8, 9 [4].
The stages of the AHP model are as follows:

First, AHP is PROCESSS: the process requires elucidating personal criteria and evaluating the relative importance of each criterion and then determine how the alternatives achieve each of the criteria.

Second, AHP organizes the decision into a HIERARCHY of criteria and alternatives: the criteria are organized according to perceived logical and natural groups to improve the clarity and usability of the model and to create properly proportional subcategories that ensure all important criteria are accounted for and receive the proper weight in the decision.

Third, AHP is ANALYTIC: it uses comparisons to help the user express the perceived relative importance of every criterion against every other criterion within each hierarchical group to establish the proportional weight each criterion should receive in the decision and it uses the relative importance of each group to establish group's weighted importance.
For the purpose of our study, the following five steps will be adopted.

Step 1: Identify all the decision variables (i.e. factors for measuring programmer productivity).

Step 2: Develop the hierarchy of criteria for prioritization.

Step 3: Establish a priority model by identifying the relative importance of criteria through pairwise comparison.

Step 4: Assess the value of each criteria for each team selection factor.

Step 5: Determine priority order while also considering other importance factors. Obviously, the highest priority professional is the best choice for the software development project.

To illustrate the algorithm and avoid any ambiguity that may exist in the explanation, a case study for prioritizing programmer productivity is demonstrated using TEN programmers in University of Benin Computer Center as our expert evaluator. Suppose three well qualified and experienced programmers are to be chosen for a software development project from the Ten programmers (P1 to P10). A set of twenty one (21) attributes called cost drivers that is considered to contribute to productivity and as a consequence to cost, is used in order to extract useful pattern. These attributes are based on the analysis provided by [8].

Step 1: Identify the variables in the decision process using expert's opinion. The following variables were identified in programmer productivity prioritization.

RELY: Required software reliability (A)
DATA: Database Size (B)
CPLX: Product Complexity (C)
TIME: Constraints in the executive time (D)
STOR: Constraints in Min Memory (E)
VIRT: Availability of Virtual Machine (F)
TURN: Service Cycle Duration (G)
ACAP: Analysts Capabilities (H)
AEXP: Analysts Experience (I)
PCAP: Programmers Capabilities (J)
VEXP: Virtual Machine Experience (K)
LEXP: Language Experience (L)
CONT: Personnel's Continuity (M)
RVOL: Requirement's Volatility (N)
TOOL: Tool Experience (O)
MODP: Use of Modern Program Practice (P)
SCED: Time Schedule (Q)
LANG: Program Language Used (R)
AT: Application Type (S)
PLAT: Platform on which project was developed (T)
MODE: Software Development Mode (U)

Step2: Develop the hierarchy of criteria for prioritization. This is shown in Figure 1.

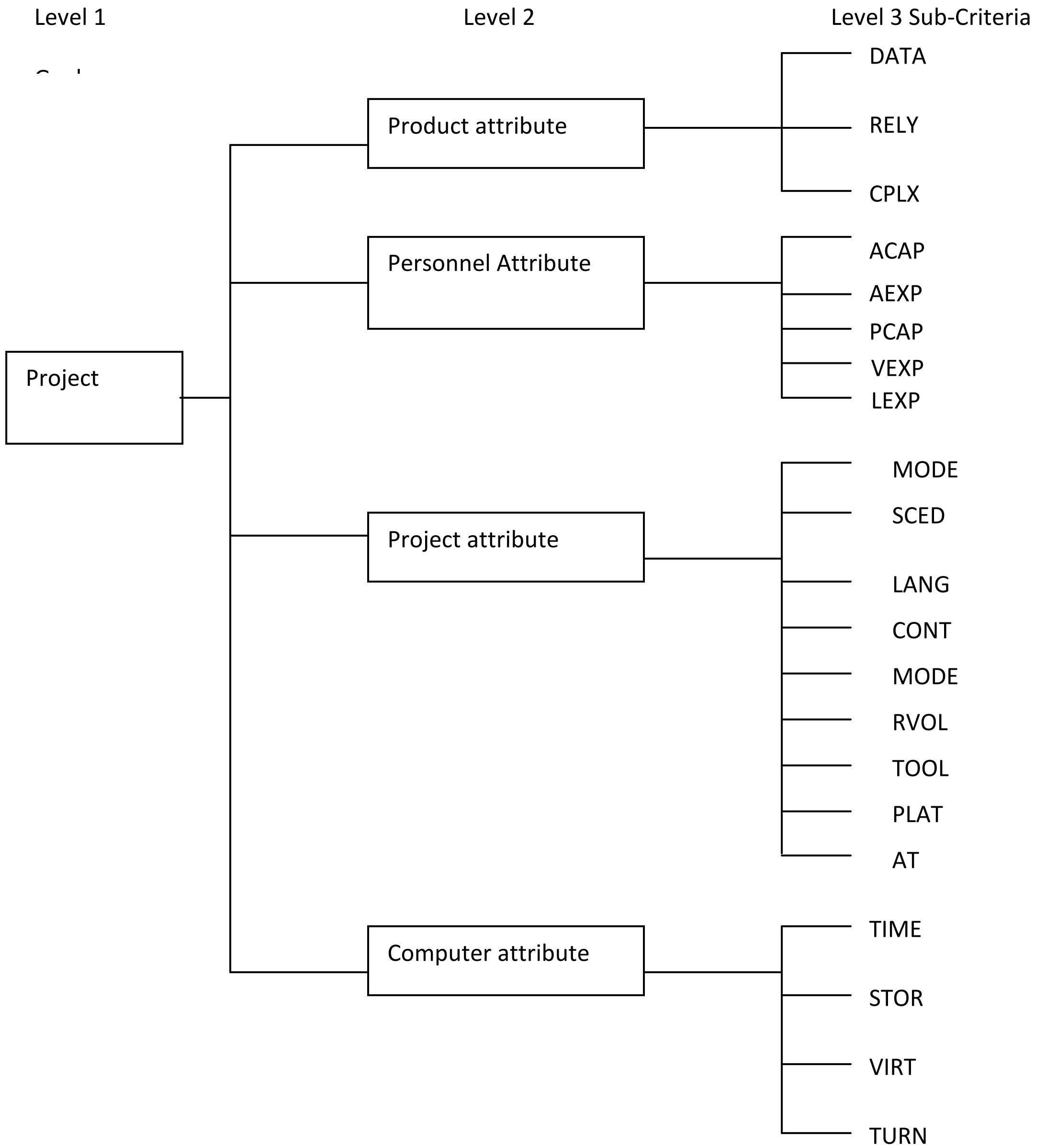

Figure 1: AHP Model for Prioritizing Programmer Productivity

Step 3: A pairwise comparison (PWC) matrix is constructed for the relative importance of these factors. This isgenerated by getting expert judgments on factor importance. The scale of absolute values in Figure 2 is used to express judgment in making paired comparison.

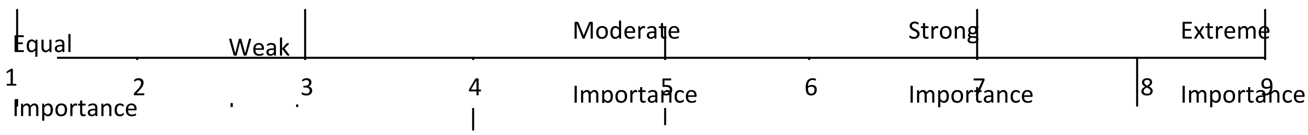

Figure 2: Scale of Absolute values

If for example two variables X and Y are being compared and both are of equal importance, then the rating of the comparison is 1. If X is likely more important than Y, then rating of the comparison is 3. The values 2,4,6,8 represent intermediate judgments used to facilitate compromise between slightly deferring judgments.

Associated with these values are their reciprocals to show the converse of the relative importance. For example if the relative importance of X to Y is 7, then the relative importance of Y to X is 1/7. Table 1 presents a transcript of the pairwise comparison matrix for the programmer productivity measure.

Step 4: The eigenvalues and eigenvectors are then derived as follows:

Obtain the column sum for each column

Divide the elements of the pairwise comparison matrix by their corresponding column sums to obtain the eigenvalues.

Obtain the relative weight of each factor with respect to its controlling factor (eigenvector) by taking the average of the elements the corresponding rows.

Table 2 presents the eigenvalues and eigenvector of the PWC matrix. A computer program is written using Visual Basic to aid in computing the eigenvalues and eigenvector once the PWC matrix has been determined.

The eignvector is used to derive a new selection factor index, which is given by applying equation (1).

$$SFI = 0.145A+0.124B+0.109C+0.79D+\ldots+0.013U \quad \ldots\ldots\ldots\ldots \quad (1)$$

Step5: Assess each available decision alternative (programmer productivity) based on the variables. Table 3 shows the alternative decision-rating matrix.

Table 4 show the qualitative scaling mechanism used to convert the linguistic judgment to numbering base and the results are listed in Table 5.

Table 4: Qualitative Scaling Mechanism

High = 1	Moderate = 0.65	Low = 0.33	None = 0

Multiply the quantified assessments by the eigenvector of the respective variables and find the sum as a basis for ranking the selection. Thus the final rating is given by applying equation (2)

$$W_i = \sum_{i,j=1}^{n} R_{ij}E_j \quad \ldots\ldots\ldots\ldots \quad (2)$$

where

W_i = weighted rating of decisions alternative i

R_{ij} = rating of decision alternative i on variable j

E_j = eigenvector of variable j

3.0 Results And Discussion

Table 5 shows that programmer (P8) emerged the first in ranking (SFSI = 0.8704). In other words, programmer 8 must be considered first in the selection process. For a best programmers team selection, the first three programmers ranking 1,2,3 (P8, P4, P1) would be considered.

Some areas in which the AHP has been successfully employed include resource allocation, forecasting, total quality management, business process re-engineering, quality function deployment, and the balanced scorecard [9, 10].

Table 5: Final Evaluation and Ranking

Decision alternative	Rating	Rank
P1	0.6202	3
P2	0.6182	4
P3	0.4815	8
P4	0.7983	2
P5	0.5711	5
P6	0.4605	6
P7	0.5138	7
P8	0.8704	1
P9	0.4354	10
P10	0.45235	6

4.0 Conclusion

Without prioritizing programmer productivity, the problem of improving the productivity of software development project which is the greatest challenge the computing industry faces in this decade will be far from solved. The main purpose of this paper is the introduction of a method based on AHP to solve the problem of prioritizing of alternatives in the Software Programmer Productivity process. It allows software Project Managers to compare factors systematically.

The Analytic Hierarchy Process is a proven, valuable and versatile decision support tool that software engineers, software project managers and specialists should consider using to improve the analysis, organization and implementation of important decision. When properly designed and applied, AHP methodology can help to elicit the relevant criteria, assess each criteria relative importance to the decision, quantify the inevitable inherently conflicting trade-offs, and structure and document the evaluation of alternatives. The resulting scores allow relative comparison of the alternatives.

The proposed method will benefit software developers in choosing a better team for their development activities. It will prioritize the process and improve the chances of producing a successful software product.

TABLE 1: PAIRWISE COMPARISON MATRIX

Variable	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
A	1	3	3	5	5	5	3	7	7	7	9	9	5	5	5	7	7	7	5	3	3
B	1/3	1	3	3	5	5	3	7	7	7	7	7	7	7	7	7	5	5	5	5	5
C	1/3	1/3	1	7	5	5	3	7	7	7	7	7	7	5	7	7	5	3	3	3	3
D	1/5	1/3	1/7	1	5	5	7	5	5	5	5	3	3	3	3	3	3	3	3	3	3
E	1/5	1/5	1/5	1/5	1	3	5	5	5	5	7	3	3	3	3	3	3	3	5	3	3
F	1/5	1/5	1/5	1/5	1/3	1	3	5	5	5	7	7	3	3	3	3	3	3	3	3	3
G	1/3	1/3	1/3	1/7	1/5	1/3	1	3	5	3	3	3	9	3	3	3	3	3	3	3	3
H	1/7	1/7	1/7	1/5	1/5	1/5	1/3	1	3	7	7	3	3	3	7	5	5	3	3	3	3
I	1/7	1/7	1/7	1/5	1/5	1/5	1/5	1/3	1	5	7	3	3	7	7	5	5	3	3	3	3
J	1/7	1/7	/17	1/5	1/5	1/5	1/3	1/3	1/5	1	9	9	3	3	5	5	5	3	3	3	3
K	1/9	1/7	1/7	1/5	1/7	1/7	1/3	1/7	1/7	1/9	1	5	3	3	3	3	3	3	3	3	3
L	1/9	1/7	1/7	1/5	1/3	1/7	1/3	1/7	1/3	1/9	1/5	1	3	3	3	5	3	5	5	3	3
M	1/5	1/7	1/3	1/3	1/3	1/3	1/9	1/3	1/3	1/3	1/3	1/3	1	3	3	3	3	3	3	3	3
N	1/5	1/7	1/5	1/3	1/3	1/3	1/3	1/3	1/7	1/3	1/3	1/3	1/3	1	3	5	3	3	3	3	3
O	1/5	1/7	1/7	1/3	1/3	1/3	1/3	1/7	1/7	1/5	1/3	1/3	1/3	1/3	1	5	3	3	3	3	3
P	1/7	1/7	1/7	1/3	1/3	1/3	1/3	1/5	1/5	1/5	1/3	1/5	1/3	1/5	1/5	1	3	3	3	3	3
Q	1/7	1/5	1/5	1/3	1/3	1/3	1/3	1/5	1/5	1/5	1/3	1/3	1/3	1/3	1/3	1/3	1	3	3	3	3
R	1/7	1/5	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/5	1/3	1/3	1/3	1/3	1/3	1	3	3	3
S	1/5	1/5	1/3	1/3	1/5	1/3	1/3	1/3	1/3	1/3	1/3	1/5	1/3	1/3	1/3	1/3	1/3	1/3	1	3	3
T	1/3	1/5	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1	3
U	1/3	1/5	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1

TABLE 2 EIGENVALUE / EIGENVECTOR

VARIABLES	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	EIGENVECTOR
A	0.194	0.390	0.274	0.243	0.196	0.177	0.102	0.161	0.146	0.138	0.125	0.131	0.097	0.092	0.077	0.098	0.109	0.115	0.079	0.05	0.048	0.145
B	0.065	0.130	0.274	0.146	0.196	0.177	0.102	0.161	0.146	0.138	0.097	0.102	0.136	0.129	0.108	0.098	0.078	0.082	0.079	0.083	0.048	0.124
C	0.065	0.043	0.091	0.341	0.196	0.177	0.102	0.161	0.146	0.138	0.097	0.102	0.058	0.092	0.108	0.098	0.078	0.049	0.047	0.05	0.048	0.109
D	0.039	0.043	0.013	0.049	0.196	0.177	0.239	0.115	0.104	0.098	0.097	0.073	0.058	0.055	0.046	0.042	0.047	0.049	0.047	0.05	0.048	0.079
E	0.039	0.026	0.018	0.01	0.039	0.106	0.171	0.115	0.104	0.098	0.097	0.044	0.058	0.055	0.046	0.042	0.047	0.049	0.079	0.05	0.048	0.064
F	0.039	0.026	0.018	0.01	0.013	0.035	0.102	0.115	0.104	0.098	0.097	0.102	0.058	0.055	0.046	0.042	0.047	0.049	0.047	0.05	0.048	0.057
G	0.065	0.026	0.03	0.007	0.008	0.012	0.034	0.069	0.104	0.007	0.042	0.044	0.174	0.055	0.046	0.042	0.047	0.049	0.047	0.05	0.048	0.049
H	0.028	0.043	0.013	0.01	0.008	0.007	0.011	0.029	0.063	0.007	0.097	0.102	0.058	0.055	0.108	0.07	0.078	0.049	0.047	0.05	0.048	0.043
I	0.028	0.019	0.013	0.01	0.008	0.007	0.007	0.008	0.021	0.098	0.097	0.044	0.058	0.129	0.108	0.07	0.078	0.049	0.047	0.05	0.048	0.048
J	0.028	0.019	0.013	0.01	0.008	0.007	0.011	0.008	0.004	0.02	0.125	0.131	0.058	0.055	0.077	0.07	0.078	0.049	0.047	0.05	0.048	0.044
K	0.022	0.019	0.013	0.01	0.006	0.005	0.011	0.003	0.003	0.002	0.097	0.073	0.058	0.055	0.046	0.042	0.047	0.049	0.047	0.05	0.048	0.03
L	0.022	0.019	0.013	0.01	0.013	0.005	0.011	0.003	0.007	0.002	0.014	0.015	0.058	0.055	0.046	0.07	0.047	0.082	0.047	0.05	0.048	0.031
M	0.039	0.019	0.03	0.016	0.013	0.012	0.004	0.008	0.007	0.007	0.003	0.005	0.019	0.055	0.046	0.042	0.047	0.049	0.047	0.05	0.048	0.027
N	0.039	0.019	0.018	0.016	0.013	0.012	0.011	0.008	0.003	0.007	0.005	0.005	0.006	0.019	0.046	0.07	0.047	0.049	0.047	0.05	0.048	0.026
O	0.039	0.019	0.013	0.016	0.013	0.012	0.011	0.003	0.003	0.004	0.005	0.005	0.006	0.006	0.015	0.07	0.047	0.049	0.047	0.05	0.048	0.023
P	0.028	0.019	0.013	0.016	0.013	0.012	0.011	0.005	0.004	0.004	0.005	0.003	0.006	0.004	0.003	0.014	0.015	0.049	0.047	0.05	0.048	0.019
Q	0.028	0.026	0.018	0.016	0.013	0.012	0.011	0.005	0.004	0.004	0.005	0.005	0.006	0.006	0.005	0.005	0.005	0.049	0.047	0.05	0.048	0.018
R	0.028	0.026	0.03	0.016	0.013	0.012	0.011	0.008	0.007	0.007	0.005	0.003	0.006	0.006	0.005	0.005	0.005	0.016	0.047	0.05	0.048	0.017
S	0.039	0.026	0.03	0.016	0.008	0.012	0.011	0.008	0.007	0.007	0.005	0.003	0.006	0.006	0.005	0.005	0.005	0.006	0.016	0.05	0.048	0.015
T	0.065	0.026	0.03	0.016	0.013	0.012	0.011	0.008	0.007	0.007	0.005	0.005	0.006	0.006	0.005	0.005	0.005	0.006	0.005	0.017	0.048	0.015
U	0.065	0.026	0.03	0.016	0.013	0.012	0.011	0.008	0.007	0.007	0.005	0.005	0.006	0.006	0.005	0.005	0.005	0.006	0.005	0.006	0.016	0.013
TOTAL	**5.15**	**7.686**	**10.942**	**20.54**	**25.473**	**28.216**	**29.307**	**43.493**	**48.027**	**50.820**	**72.197**	**68.598**	**51.664**	**54.198**	**64.865**	**71.665**	**64.332**	**60.999**	**63.666**	**60.333**	**63.00**	

TABLE 3: THE ALTERNATIVE DECISION RATING MATRIX

Decision Variable / Decision Alternative	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
P1	High	High	Low	Mode	High	None	Mode	High	High	Low	None	Mode	None	Mode	None	None	None	High	None	None	High
P2	Low	High	Mode	Mode	High	Mode	None	High	High	Mode	Low	High	None	Low	Mode	High	None	Mode	None	High	None
P3	Mode	Mode	None	Mode	Low	None	Low	High	High	High	None	High	None	None	High	Mode	None	Mode	Low	Low	None
P4	High	High	High	High	High	None	Mode	High	High	High	None	Mode	Low	None	Mode	High	Mode	High	Mode	Mode	None
P5	Mode	Mode	None	Low	High	None	High	High	High	High	Mode	High	Low	None	High	Mode	None	Mode	Mode	Mode	None
P6	High	High	None	Mode	Mode	None	None	Low	Low	High	None	Low	None	None	Low	None	None	Mode	None	None	None
P7	High	Mode	None	Low	Mode	Low	Low	High	Mode	Low	None	Mode	Low	Mode	Mode	Mode	None	High	Low	Mode	None
P8	High	High	High	Mode	High	Mode	High	High	High	High	Mode	Mode	Low	High	Mode	Mode	Low	High	Mode	Mode	High
P9	Low	Mode	None	High	Low	None	High	Mode	Mode	Mode	None	Low	Mode	None	None	None	None	High	Mode	Mode	High
P10	High	Mode	None	High	Mode	None	Low	Mode	Mode	Mode	Mode	Mode	Mode	None	Mode	Low	Low	Mode	None	None	None
Eigenvector	0.145	0.124	0.109	0.079	0.064	0.057	0.049	0.043	0.048	0.044	0.030	0.031	0.027	0.026	0.023	0.019	0.018	0.017	0.015	0.015	0.013

References

[1] Aina, A.T. (1992) "The Challenge of Sustainable Development in Nigeria." PP 55 – 106. Publisher: Nest. ISBN (10): 9783120344 and ISBN (13): 9789783120341

[2] Taylor, W.F. (1991) "The Principles of Scientific Management." New York: (Harper Bros.): 5 – 29. Available on line: www.fordham.edu/halsall/mod/1911taylor.hatml

[3] Chauncey, E.W. (2005) "Categories of Return on Investment and Their Practical Implications." Cost-Justifying Usability: An update of an Internet Age, pp 215 – 250. Edition: 2, illustrated, revised. Publisher: Morgan Kaufmann, ISBN 0120958112, 9780120958115.

[4] Saaty, T.L. (1980) "The Analytic Hierarchy Process", McGrwa Hill, New York. Citeseer.ist.psu.edu/context/45323/0

[5] Saaty, T.L. (1994) "Fundamentals of Decision making and priority Theory with the Analytic Hierarchy Process. RWS Publications: urg. Superdecicions/~saaty/ISAHP2005/PAPERS/…

[6] Saaty, T.L. (1996) "The Analytic Hierarchy Process". Pittsburgh. PA: RWS Publications. fb.cityu.edu.hk/ms/objs/content/…/MS6219Performancemeasurement.pdf

[7] Ahmad, N, Berg, D and Simons, G. (2006) "The Integration of Analytical Hierarchy Process and Data Envelopment Analysis in a Multi-criteria Decision Making Problem" International Journal of information Technology & Decision making vol. 5 No. 2 pp. 263 – 276. Available on line: www.eng.rpi.edu/dses/csre/people.cfm

[8] Kitchenham, B. (1992) "Empirical Studies of Assumptions that Underlie Software Cost-Estimation Models." Information and Software Technology, Volume 34, Issue 4, PP: 211 – 218. ISSN: 0950 -5849. Publisher: Butterworth – Heinemann Newton, MA, USA.

[9] Saaty, T.L. (1991) "Prediction, Projection and Forecasting: Applications of the Analytic Hierarchy Process in Economics, Finance, Politics, Games and Sports" ISBN0-7923-9104-7, Kluwer Academic.

[10] Kitchenham, B. and Kansala K. (1993) "Inter-item Correlations among Function Points." International Conference on Software Engineeringarchive.Proceedings of the 15th international conference on Software Engineering table of contents. Baltimore, Maryland, United States. PP: 477 – 480. ISBN:0-89791-588-7. Publisher: IEEE Computer Society press Los Alamitos, CA, USA.