International Journal of Soft Computing and Software Engineering (JSCSE)
Vol.6, No.3, 2016

Published online: May 25, 2016

e-ISSN: 2251-7545

Performance Evaluation of Software Projects using Criteria Importance through Inter-criteria Correlation Technique

*1Madan Kumaraswamy, 2R. Ramaswamy

*1, *Fellow, National Institute of Industrial Engineering, Mumbai, India*
2, *Professor, National Institute of Industrial Engineering, Mumbai, India*
Email: 1madan101@yahoo.com, 2ramaswamy2008@gmail.com

Abstract. **Software project management problems are highly complex, multi-dimensional, and not always objective in nature. Performance evaluation of software projects is a multi-criteria decision. This research paper highlights the applicability of a multi criteria decision making approach to understand software project management decision problems. A multi criteria decision model based approach to decision making helps project managers to identify significant decision criteria that transmits maximum information to influence the decision, criteria that have the highest conflicting information, rank the projects on their performance basis the correlated project criteria. A field study on performance evaluation of software development projects based on three project criteria; project complexity, project team size, and actual effort to complete the project is used to illustrate the multi criteria project performance evaluation problem. The research paper corroborates the findings in literature that project complexity is the criteria that transmits larger information and therefore higher is its importance in project evaluation decision making. The project performance scores based on criteria weights is then used to rank the projects based on their relative performance on the correlated criteria.**

Keywords: *Multi-Criteria Decision Making, Software Project Performance, Complexity*

* Corresponding Author:

Madan Kumaraswamy,

Fellow, National Institute of Industrial Engineering (NITIE)
Vihar Lake, Mumbai, India

Email: madan101@yahoo.com Tel:+91-9867583699

1. Introduction

Multi Criteria Decision Making [MCDM] problems are traditionally solved through the concept of establishing the relative importance of attributes that influence the choice of the decision alternatives (Zeleny, 1982). The information importance of criteria is measured using the deviation of the quantified criteria and arriving at objective weights like in Analytic Hierarchy Process (AHP) (Saaty, 1988). CRiteria Importance Through Inter-criteria Correlation (CRITIC) is a method of ranking alternatives based on multi criteria evaluation, especially when the criteria are correlated and the decision maker is not able to give relative preferences (Diakoulaki et al., 1995). This has significant impact on the research problem of software project evaluation models where the model parameters are highly correlated. CRITIC is applied when the criteria are correlated and is an effective method to

e-ISSN: 2251-7545

assess the convergent validity of decision variables that are related (Diakoulaki et al., 1995). The applicability of CRITIC method is established in the area of multi firm comparison for economic and financial analysis of companies to determine which firm ranks higher in-terms of relative performance (Diakoulaki et al., 1995; Fernando et al., 2010; Vasu Jain, 2013). CRITIC uses the research gap in other MCDM approaches where the quantified criteria are not different from each other in terms of their importance in decision making. Diakoulaki et al. (1995) added the concept of conflict to multi criteria decision making. Every decision maker faces a conflict in choosing the criteria and then the best alternative. It is this conflict that is measured to determine whether these inter-related decision criteria are indeed divergent and contribute to the performance evaluation in ranking and decision making.

2. Software Performance Metric and Measurement

A metric is a number or scale of measurement derived from an identifiable attribute of a product, process, or resource. Software metric models are theoretical or data driven models describing a dependent variable (effort, cost, schedule, defects) as a function of independent variables (size, resources, time, complexity) (Fenton, 1991). "The continuous application of measurement-based techniques to the software development process and products to supply meaningful and timely management information, together with the use of those techniques to improve that process and products is called software metrics" (Goodman, 1993). The International Function Point Group (IFPUG) defines software measurement as "The definition, collection, consolidation, analysis, and reporting of quantitative and qualitative measures within the software development and maintenance area" (IFPUG 2002).

Software metrics is the definition, collection, consolidation, analysis, and reporting of qualitative and quantitative measurement-based techniques to support management decisions in software development process and products (Gilb, 1977; Goodman 1993; IFPUG, 2002). The core metrics in software development are resource metrics, product metrics, quality metrics, and productivity metrics. Gopal et al. (2002) measured success of metrics programs in multiple software firms with usage of metrics in decision-making and its influence on organization performance as dependent variables. The usage of metrics in management decision making was linearly dependent on increase in organizational performance, frequency of metric collection, sophistication of metrics, data collection techniques used, metrics quality, analysis, usage of automated tools, communication, and feedback. Measuring software project performance is not a one dimensional metric that can be derived from a select objective criteria. A subjective criteria like software project complexity requires multiple factors and attributes to evolve itself into a metric. Change in user requirements, manpower attrition, low team coordination, geographically separated teams, low skill sets, new set of technology used, lack of reusability of the existing code, Inability of project managers to lead from the front and establish clear goals, number of interfaces of the application developed, all contribute to the software project complexity metric (Scacchi, 1995; Wagner and Ruhe, 2008; Jiang and Comstock, 2007; Tockey 1996; Nwelih and Amadin, 2008; Blackburn et al., 2002; Yan et al. 2012). The other project performance evaluation measure is software development effort metric used to measure the units of input (effort) required to develop and deliver a software development project usually measured in person months or person days ((Boehm, 1981; Boehm, 2000).

3. Analytic Hierarchy Processing (AHP)

The fundamentals of decision making approach revolve around the ability of a decision maker to judge the relative importance of different decision parameters to achieve the end goal of an informed decision that has the best outcome. Analytical Hierarchical Processing (AHP) uses this fundamental relative comparison and rating method to help in weighing the decision attributes. AHP uses the technique of first identifying the attributes (criteria) that aggregate into the decision alternatives (Saaty, 1980). AHP uses the deductive logic where the decision problem is designed as a hierarchy to develop a score to rank each decision alternative. The ranking of alternatives is based on the relative preferences at all levels in the hierarchy. The first step in AHP based decision making is to construct a hierarchy of the problem statement. The second step is to do a pair-wise comparison of the each decision alternative on each of the decision criteria or attribute. The outcome of each pairwise comparison is a singular, reciprocal, positive, and square matrix. The geometric mean of each alternative in the pairwise comparison is then calculated to arrive at the attribute weights for each alternative. Finally, an eigen vector matrix for the alternatives is constructed which is the relative performance matrix for the alternatives to be selected. The product of the performance matrix of alternatives and the criteria weights gives the overall performance score of each alternative based on the decision criteria. The alternative which has the highest overall score is the decision alternative chosen. This approach is thus useful to decide which decision to choose based on the mutually exclusive and non-related decision attributes. This however is also a pitfall in AHP, when the decision alternatives are themselves tightly coupled that the pair-wise comparisons themselves are not consistent in getting a clear decision. The pairwise judgments more often tend to be untrustworthy and randomly inconsistent when there are many decision attributes to be compared. This has been the major drawback of AHP when the consistency of comparison is not reliable when there are multiple attributes and alternatives. It is in this context that an alternative to AHP method for multi criteria decision is required without compromising the benefits of relative comparison weights and performance scoring.

4. Criteria Importance Through Inter-criteria Correlation

The Criteria Importance Through Inter-criteria Correlation (CRITIC) is a methodology often used to assess performance ratings basis different criteria or variables using their correlations (Diakoulaki et al., 1995). Decision support models like Analytic Hierarchy Processing (AHP) are popular methods to solve multi-criteria problems but are effective when criteria are independent. AHP is a composite structured methodology by which a decision problem is designed as a hierarchy to develop a score to rank each decision alternative based on preferences at all levels in the hierarchy (Saaty, 2008). The advantages of traditional AHP model in Multi criteria decision making [MCDM] is that it is a relative comparison of alternatives based on pair wise comparison and weighting of the decision attributes. Research studies also highlight the disadvantages of application of pairwise comparison of independent attributes which are relative as inaccurate if the decision criteria overlap, and decision alternatives change when one of the decision criteria is replaced by another. This result in reversal of ranking of the decision alternative from high to low or vice-versa if a non-significant decision alternative is introduced (Stam and Silva, 1997).

CRITIC method uses correlation analysis to determine conflicts between decision criteria. The major element of the performance matrix (U) where the columns C (C_1,C_2, …,C_m) are the criteria and rows are the alternatives P (P_1,P_2, …Pn) with entries (P_{ij}) being indicators of alternatives across criteria. The first step is to transform the matrix of alternatives and evaluation criteria into a normalized score matrix. The performance matrix is normalized to derive a score matrix with relative scores of alternatives. For each of the criteria value, identify the maximum value within the criteria (Max(p_{11}:p_{n1}) and minimum value within the criteria (Min(p_{11}:p_{n1})). For each of the criteria values the score is determined. The second step is to determine the standard deviation (σ) of scores for each criteria. With the normalized matrix R and criterion C_j (j=1,2, …,m), by examining the j^{th} criterion in isolation we generate a vector r_j, denoting the scores of all n alternatives.

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} \cdots & r_{1m} \\ r_{21} & r_{22} \cdots & r_{2m} \\ \cdots & \cdots \quad \cdots & \cdots \\ r_{n1} & r_{n2} \cdots & r_{nm} \end{pmatrix}$$

Figure 1. Normalised matrix (R)

e-ISSN: 2251-7545

Each vector r_j is characterized by the standard deviation which denotes the conflict or contrast intensity of the corresponding criterion. The third step is to derive a correlation matrix from the score matrix. A symmetric matrix is constructed, with dimension m x m and a generic element x_{jk}, which is the linear coefficient between vectors r_j and r_k. The more divergence in scores of the alternatives in criteria j and k, the lower the value of x_{jk}. The fourth step is to determine the conflict of each decision criteria and then the weights of each criteria. The sum shown below denotes the conflict measure created by criterion j with respect to the decision (Figure 2).

$$\text{Conflict} = \sum_{k=1}^{m} (1 - X_{jk})$$

Figure 2. Conflict

The final performance score for each alternative The amount of information Pj, given by the jth criterion in terms of conflict and contrast is determined by the multiplicative aggregation formula (Figure 3).

$$P_j = \sigma_j \sum_{k=1}^{m} (1 - X_{jk})$$

Figure 3. Performance score

The higher the value of P_j, more is the information from the criteria and the relative importance is higher for the decision making process. The objective weights result from the normalization of values as per the below equation (Figure 4).

$$w_j = P_j / \sum_{k=1}^{m} P_j$$

Figure 4. Performance weightage

5. Ranking and Evaluation of Software Project Performance

For the set of correlated project performance data, if the project manager wants to evaluate project performance for similar sized projects based on the three factors of actual effort, complexity, and size of the team the CRITIC method is best suited. The method is used to evaluate the 24 projects in the field study. The criterion used for evaluation of past project performance are actual efforts, project complexity, and team size. Lederer and Prasad (1993) in their questionnaire based study cite the importance given by practitioners on complexity of the software developed as the most important factor above all other factors. CRITIC as a multi criteria decision technique is used to identify the criterion which passes the maximum information to the decision of choosing the project that performance well based on the collective correlated criteria correlation. The visual representation of the multi criteria decision making flow is given in Figure 5.

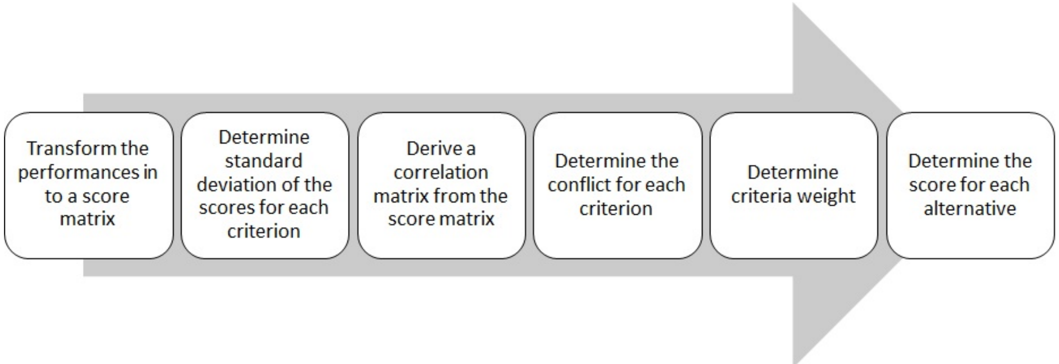

Figure 5. CRITIC decision making flow

Function point [FP] metric measures the various functions of a software application from the user point of view. Function Point Analysis (FPA) is the end user perspective of software size based on the count of data functions and count of transaction functions or elementary processes in the system (Xia et al., 2008; Jorgensen and Shepperd, 2007). The size characteristics of the project are measured in terms of Function Points [FP]. The function points range from 10-200 FP with one project having 333 FP. Hence, all projects in this study can be categorized as Small or Medium sized projects with varying complexity. The categorization of project complexity of is based on two factors, number of external interfaces that the application or enhancement to the application requires (Interface Complexity), whether the application developed is a new product or a change to the existing application (Development Complexity). While there are multiple factors impacting project complexity cited in research literature, the scope of this research is to use complexity factors that are used within the organization in this study and also the factor used should be objectively measurable unlike other factors like team productivity, people complexity, skill sets available. The complexity of the projects were categorized into High-3 (4 projects), Medium-2 (9 projects), and Low-1 (11 projects) complexity. The team sizes for the projects varied from a minimum of 4 people to a maximum of 15 people in a project.

6. Findings and Results

Complexity followed by team size is the criteria that influences the project performance more than the actual effort put in to the project. The effort required for the project can vary but is not an influencing criteria as much as complexity. Complexity thus is the criteria that transmits larger information and higher is its importance in decision making. The findings further establishes what has been cited in earlier literature that complexity and team size are the two most important project attributes that determine the productivity (Lederer and Prasad, 1993; Hill, 2010). The performance score basis the entire criterion taken together is higher for project number 2 which had a higher complexity in the dataset and hence also a larger team size than other projects (Table 3). The criteria that transmits maximum conflict and contrast to the relative performance ranking is actual effort. The relative performance scores of projects 2, 4, 23, and 16 also can be used as an indicator of best performing projects if the three criteria were used as the basis of performance decision. The importance of using CRITIC in this case study is best explained by the fact that the performance scores, though not in much divergence from each other helps the project manager to understand the relative performance of similar projects. CRITIC helps the project manager not only to identify the best project but more importantly to identify the most importance criteria that influences the overall decision. CRITIC is thus a very effective technique in comparison of project risks based on different risk attributes of the project which enables the project manager to weigh the risks and find which attributes conflict. The applicability of CRITIC is high in areas of software product vendor selection based on product delivery and service criteria, and software project risk and impact assessments.

Table 1. Project Criteria Matrix for Performance Evaluation

Project	**Actual Efforts (Person Days)**	**Complexity**	**Team Size**
1	85	2	10
2	136	3	15
3	54	2	8
4	127	3	15
5	90	3	12
6	60	2	10
7	79	1	6
8	61	1	5
9	42	1	7
10	45	1	9
11	91	2	14
12	45	1	12
13	18	1	4
14	25	1	4
15	32	1	4
16	660	2	12
17	35	1	4
18	38	2	6
19	57.6	2	6
20	35	1	4
21	48	1	5
22	232	2	8
23	322	3	11
24	66	2	8
MAX	**660**	**3**	**15**
MIN	**18**	**1**	**4**

Table 2. Correlation Matrix for Project Evaluation Criteria

	Actual Efforts(Person Days)	**Complexity**	**Team Size**
Actual Efforts(Person Days)	1	0.394707384	0.419185174
Complexity	0.394707384	1	0.743460963
Team Size	0.419185174	0.743460963	1

Table 3. Project Evaluation Performance Score and Ranking

Project	**Actual Efforts (Person Days)**	**Complexity**	**Team Size**	**Performance**
2	0.054410567	0.37795408	0.326016	0.758380192
4	0.050260609	0.37795408	0.326016	0.754230234
23	0.140176377	0.37795408	0.207464	0.725594893
16	0.296030375	0.18897704	0.237102	0.722109631
5	0.033199668	0.37795408	0.237102	0.648255962
11	0.033660775	0.18897704	0.296378	0.519015585
22	0.098676792	0.18897704	0.118551	0.406204938
1	0.030894136	0.18897704	0.177827	0.397697837
6	0.019366473	0.18897704	0.177827	0.386170174
24	0.022133112	0.18897704	0.118551	0.329661259
3	0.016599834	0.18897704	0.118551	0.324127981
19	0.018259818	0.18897704	0.059276	0.26651241
18	0.00922213	0.18897704	0.059276	0.257474722
12	0.012449876	0	0.237102	0.249552093
10	0.012449876	0	0.148189	0.160638762
9	0.011066556	0	0.088913	0.099979888
7	0.028127497	0	0.059276	0.087403051
8	0.01982758	0	0.029638	0.049465357
21	0.013833195	0	0.029638	0.043470972
17	0.007838811	0	0	0.007838811
20	0.007838811	0	0	0.007838811
15	0.006455491	0	0	0.006455491
14	0.003227746	0	0	0.003227746
13	0	0	0	0
Std Dev	0.213587736	0.37530181	0.333191	
Conflict	1.186107442	0.86183165	0.837354	
Pj	0.253338003	0.32344698	0.278999	
Weight	0.296030375	0.37795408	0.326016	

7. Summary and Conclusions

The findings of the research presented in this paper establishes software performance measurement as a multi criteria decision problem which captures interaction of multiple criteria and their contrasts in ranking software projects based on criteria weights. The information transmitted by multi-dimensional criteria like software project complexity is high with relatively lesser conflict than the actual effort required to complete a project. The results also signify that software effort is not always the correct measure of determining software project success. A lesser effort spend on completion of a highly complex project does not always mean that the project performed better than another project which took more time and effort for completion. The weights of software project complexity and project team size were higher than actual effort expended for completion of the project. The research can be extended to include more project performance criteria like project cost, number of defects that are correlated criteria that determine project performance. The complexity of the selected software project management problem is evident from the performance score of the project being dominated by the complexity type which itself is open to the subjective interpretation of the project manager. The weights and scores arrived through the methodology is an attempt to interpret the complex attributes of a software project is the most objective, and empirically valid technique.

References

Boehm, B. W., Software Engineering Economics, Prentice Hall, NJ. 1981.

Boehm et al., Software Cost Estimation with Cocomo II, Prentice Hall, NJ. 2000.

Blackburn, J. D., Lapre, M. A., and Van Wassenhove, L. N., “Brookes Law Revisited: Improving Software Productivity by Managing Complexity”, Vanderbilt University, Working paper, 2002-85.

Diakoulaki D., Mavrotas G., Papayannakis L., “Determining objective weights in multiple criteria problems: The critic method”. Computers and Operations Research, vol. 22 no.7, pp.763-770, 1995.

Fenton, N., Software Metric: A Rigorous approach. London: Chapman and Hall., 1991

Gilb, T., Software Metrics, Winthrop Publishers, Inc., Cambridge, Massachusetts, 1977.

Goodman, P., "Implementing Software Metrics Programmes: A Project-Based Approach," In Proceedings of Eurometrics '92, pp. 147 – 151, April 1992.

Gopal, A., Krishnan, M. S., Mukhopadyay, T., and Goldenson, R. D., “Measurement Programs in Software development: Determinants of Success”, IEEE Transactions on Software Engineering, vol. 28 no. 9, pp.863-875, 2002.

Hall, T., and Fenton, N., “Implementing Effective Software Metrics Programs”. IEEE Software, vo. 14, no. 2, pp. 55-66, 1997.

Hill, P., Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort & Duration, McGraw-Hill Osborne, ISBN-10: 0071717919, 2010.

IFPUG, IT Measurement- Practical Advice from the experts, Addison-Wes;ey Information Technology Series, Pearson Education Asia, India., 2002.

Jiang, Z., and Comstock, C., “The Factors Significant to Software Development Productivity”, World Academy of Science, Engineering and Technology, Volume 25, January 2007.

Lederer, A.L., and Prasad, J., “Nine Management Guidelines for Better Cost Estimating”, Communications of the ACM, vol. 35, no. 2, pp. 51 – 59, 1992.

Nwelih, E., and Amadin, I.F., “Modeling Software Reuse in Traditional Productivity Model”, Asian Journal of Information Technology, vol. 7, no. 11, pp. 484-488, 2008.

Saaty, T.L., “Decision making with the analytic hierarchy process”, Int. J. Services Sciences, vol. 1, no. 1, pp.83–98, 2008.

Scacchi, W., “Understanding Software Productivity”. Article in software Engineering and Knowledge Engineering: Trends for the Next Decade edited by D.Hurley, Vol. 4, World Scientific Press, 1995.

Stam, A., and Silva, A. P. D., “Stochastic Judgments in the AHP: The Measurement of Rank Reversal Probabilities”, Decision Sciences, 28: 655–688, 1997.

Tockey, S., “The Effect of Team Size on Team Productivity and Project Cost”, Lecture notes of Software Project Management, CSSE-515, Seattle University, 1996.

Wagner, S., and Ruhe, M., “A Systematic Review of Productivity Factors in Software Development”, Proceedings of 2nd International Workshop on Software Productivity Analysis and Cost Estimation (SPACE 2008), State Key Laboratory of Computer Science, Institute of Software, 2008.

Xia W., Capretz L.F., Ho D. and Ahmed, F., “A New Calibration for Function Point Complexity Weights, Information and Software Technology, Volume 50, Issue 7-8, pp. 670-683, DOI: 10.1016/j.infsof.2007.07.004, Elsevier, June 2008.

Zeleny, M., Multiple Criteria Decision Making, McGraw-Hill, New York, USA, 1982.

Parvez Mahmood Khan, M.M.Sufyan Beg ,"Measuring Cost of Quality(CoQ)- on SDLC projects is indispensible for effective Software Quality Assurance", International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 2, No. 9, pp. 1-15, 2012.

Yan Chun-man, Guo Bao-long, Wu Xian-xiang ,"Empirical Study of the Inertia Weight Particle Swarm Optimization with Constraint Factor ", International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 2, No. 2, pp. 1-8, 2012.

Jorgensen, M., and Shepperd, M., “A systematic review of software development cost estimation studies”, IEEE Transactions on Software Engineering, 33.1,pp. 33-53, 2007.

Vasu Jain,"Prediction of Movie Success using Sentiment Analysis of Tweets", International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, pp. 308-313, 2013.