IJCSC International Journal of Computer Science & Communication Vol. 1, No. 2, July-December 2010, pp. 405-408

The Factors Affecting in Software Development Using Factor Analysis Model

Seetharam. K[1] & Chandrakanth G. Pujari[2]

[1]Dept. of CSE/IT, MeRITS Udayagiri, Andra Pradesh State
[2]Dept. of M.C.A., Dr.AIT,Bangalore, Research Scholar Sathyabhama University, Chennai
Email: [1]amoghks@rediffmail.com, [2]varuncp@rediffmail.com

ABSTRACT

Factor Analysis is a statistical technique which can be used to reduce a large number of response variables to a smaller set of uncorrelated variable as well as to interpret these newly created variable. The factor analysis model assumes that this new set drives or controls the value of the variables that are actually being measured. If and How this underlying set can be interpreted is almost always determined subjectively by the researcher rather than objectively by the statistical methods.

This paper presents the findings of empirical research from 50 validated projects chosen from software managers to identify the factors that may impact software reliability. Qualitative/quantitative measurement of software quality related aspects in all stages of software development are desirable. Any measurement in this line forms an element in the set of software quality metrics.Here the seven elements considered are size, effort, duration, S1 (customer participation), S2 (staff availability), S3(standards use) and S4 (methods use). In this paper three different cases are carried out by means of factor analysis. One analysis with size as predominant factor, second analysis with effort as predominant factor, and third analysis with duration as predominant factor with software reliability performance. The analysis of variables is to identify the dimension that are latent. That can be considered in the phenomena of performance correlation. This is to study the effects in using factor analysis approach.

Keyword: Software Reliability, PCA, Factor Analysis

1. INTRODUCTION

Computer software has gradually become an indispensable element in many aspects of our daily lives and an important factor in numerous systems. In software engineering software matrix plays wide and deeper scope[1,3]. Software industry is facing many problems. Industry is in the infancy about the estimation of software and other depending factors[4]. The paper discusses four software factors variation with application *size*, corrective maintenance *effort* and also *duration* as reference variable. Taking S1 (*Customer participation*), S2 (*Staff availability*), S3 (*Standard Use*) and S4 (*Methods use*), as depending variables.

Five different levels are identified for each variable separately by means of fuzzy logic[2,14]. Three analysis are carried out. The variables are standardized using normal distribution principles. Covariance matrix is generated[18].

Three statistical analysis is carried out by using i) Size, s1, s2, s3 and s4 ii) Effort, s1, s2, s3 and s4 iii) duration, s1,s2,s3 and s4 for factor analysis[4,17,18].

Factor analysis is an interdependence technique. This analysis demonstrates its basic objective of grouping highly inter correlated variables into distinct factors. These factors can provide wealth of information about the interrelationships of variables. Factor analysis identified for software management[6,9]. A smaller set of concept to consider in any strategic or tactical marketing plans, while still providing insight into what constituents each general area. That is individual variable defining each factor.

2. ANALYSIS

The measured design matrix were analyzed for their correlation as shown in table T1,T2,T3 for size, effort, duration respectively. The Correlation coefficients are large for size, Customer participation, Staff availability.

T1	*Size*	*S1*	*S2*	*S3*	*S4*
Size	1				
S1	0.552	1			
s2	0.147	0.465	1		
s3	-0.52	-0.087	0.324	1	
s4	-0.77	-0.096	0.096	0.597	1

T2	*Eff*	*S1*	*S2*	*S3*	*S4*
Eff	1				
S1	0.184	1			
S2	0.389	0.465	1		
S3	-0.21	-0.087	0.324	1	
S4	-0.77	-0.096	0.096	0.573	1

T3	*Dur*	*S1*	*S2*	*S3*	*S4*
Dur	1				
S1	0.3661	1			
S2	0.2132	0.4653	1		
S3	-0.361	-0.0867	0.3241	1	
S4	-0.8913	-0.0956	0.0957	0.5973	1

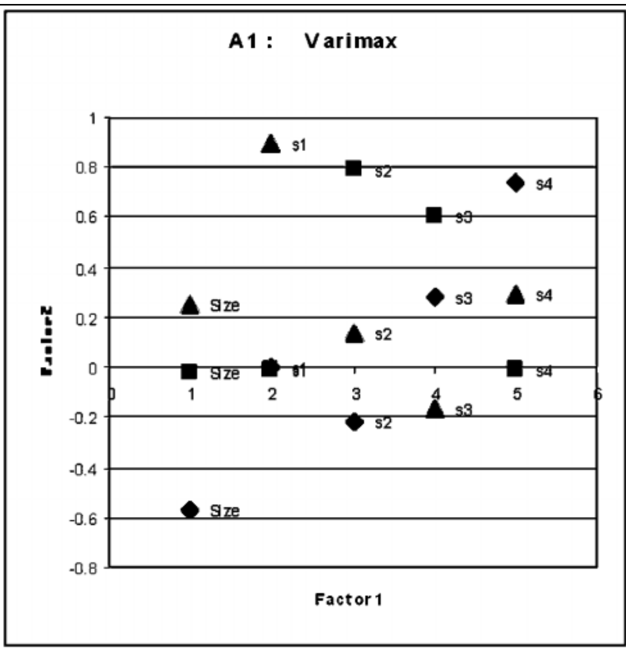
This suggests that the matrix not only measures unique aspects of the design but also somewhat related to each other. This is shown darkened within the data set.

Orthogonal factor rotation method is the most popular rotation method focusing on simplifying the columns in a factor matrix. Generally considered superior to other orthogonal factor rotation methods in achieving a simplified factor structure. Selecting a rotational methods are shown in four tables. These tables are 1) Varimax 2) Equimax 3) Quartimax 4) Parsimax also called as orthogonal methods[7,8,9].

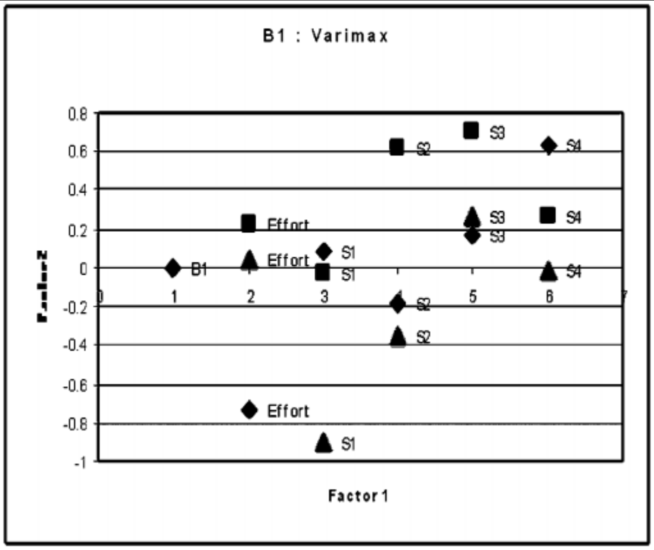
1) *Varimax Rotation:* An orthogonal rotation method that minimizes the number of variables Size, s1, s2, s3, s4 that have high loadings on each factor. It simplifies the interpretation of the factors. Shown in table A1.

Similar interpretation with effort as predominant factor given in table B1, with duration is in table C1.

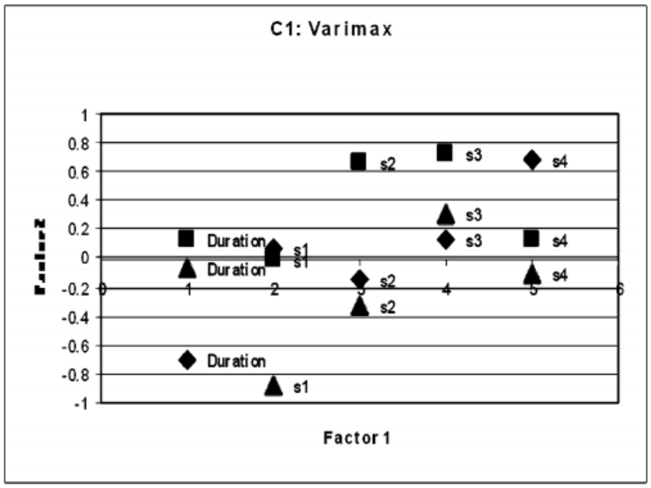
A1	*Varimax*		
Size	–0.5729	–0.0212	0.2526
s1	0.006	–0.0054	0.8965
s2	–0.219	0.7962	0.1409
s3	0.2773	0.6046	–0.1652
s4	0.7395	–0.0074	0.292



B1	*Varimax*		
Effort	–0.7307	0.2222	0.0348
S1	0.0819	-0.034	–0.8991
S2	–0.1908	0.6174	–0.3508
S3	0.1664	0.7065	0.2588
S4	0.6287	0.2631	–0.0174



C1	*Varimax*		
Duration	-0.6999	0.1329	-0.0608
s1	0.0647	-0.013	-0.8908
s2	-0.1406	0.6563	-0.3197
s3	0.1323	0.731	0.3004
s4	0.6846	0.1306	-0.1017



2) *Equimax Rotation:* A rotation method that is a combination of the varimax method, which simplifies the factors, and the quartimax method, which simplifies the variables. The number of variables Size, s1, s2, s3, s4 that load highly on a factor and the number of factors needed to explain a variable are minimized. as shown in table A2. Similar interpretation with effort as predominant factor given in table B2, with duration as predominant factor is in table C2.

A2	*Equmax*		
Size	-0.5702	-0.0222	0.2585
s1	0.0152	-0.0046	0.8964
s2	-0.2192	0.7958	0.1425
s3	0.2743	0.6051	-0.1685
s4	0.7425	-0.0055	0.2844

B2	*Equmax*		
Effort	-0.7314	0.22	0.0325
S1	0.0832	-0.0391	-0.8988
S2	-0.1924	0.6147	-0.3547
S3	0.1638	0.7085	0.2548
S4	0.6279	0.265	-0.0182

C2	*Equmax*		
Duration	-0.6999	0.1326	-0.0614
s1	0.065	-0.0158	-0.8907
s2	-0.1406	0.6553	-0.3218
s3	0.132	0.732	0.2982
s4	0.6846	0.1305	-0.1018

3) *Quartimax Rotation:* A rotation method that minimizes the number of factors needed to explain each variable Size, s1,s2,s3 and s4. It simplifies the interpretation of the observed variables.Given in table A3.

Similar interpretation with effort as predominant factor given in table B3, with duration as predominant factor is in table C3.

A3	*Quartimax*		
Size	-0.5778	-0.0189	0.2415
s1	-0.0113	-0.0066	0.8965
s2	-0.218	0.797	0.1378
s3	0.2832	0.6036	-0.1589
s4	0.7337	-0.0111	0.3063

B3	*Quartimax*		
Effort	-0.7293	0.2262	0.0388
S1	0.0796	-0.025	-0.8996
S2	-0.1879	0.6222	-0.3437
S3	0.1712	0.7027	0.2659
S4	0.6303	0.2595	-0.0161

C3	*Quartimax*		
Duration	-0.6999	0.1336	-0.0595
s1	0.064	-0.0077	-0.8909
s2	-0.1405	0.6583	-0.3157
s3	0.1329	0.7291	0.3047
s4	0.6846	0.1309	-0.1014

4) *Parsimax Rotation:* A rotation method is Special case of the orthomax rotation (default). Specifies families of orthogonal rotations. Gamma specifies the member of the family to use. Varying Gamma changes maximization of the variances of the loadings from columns (Varimax) to rows (Quartimax) shown in table A4.

Similar interpretation with effort as predominant factor given in table B4, with duration as predominant factor is in table C4.

A4	*Parsimax*		
Size	-0.5693	-0.0225	0.2605
s1	0.0184	-0.0043	0.8964
s2	-0.2193	0.7957	0.143
s3	0.2733	0.6052	-0.1696
s4	0.7435	-0.005	0.2818

B4	*Parsimax*		
Effort	-0.7317	0.2193	0.0318
S1	0.0836	-0.0409	-0.8987
S2	-0.1929	0.6137	-0.356
S3	0.1629	0.7092	0.2534
S4	0.6276	0.2656	-0.0185

C4	*Parsimax*		
Duration	-0.6999	0.1324	-0.0616
s1	0.0651	-0.0167	-0.8907
s2	-0.1407	0.6549	0.3225
s3	0.1319	0.7323	0.2974
s4	0.6846	0.1304	-0.1019

3. SUMMARY AND CONCLUSION

This paper has shown that it is possible to identify key factors that have a impact on and relate software development process success. In this analysis we have identified seven factors for the software development process. Correlations of the factors has analyzed. This information will be used when determining which factors should be taken into consideration in formulating software reliability.

The application of factor analysis produces reduction in the data dimensionality through the aggregation of correlated variables into a smaller set of uncorrelated factors.

The high degree of freedom involved in the available choice of methods and rotation methods (Varimax, Equimax, quartimax, parsimax) within factor analysis.

In all four rotations S3 is highly correlated for effort and duration, S1, S2 is correlated for size, S4 is correlated for size, effort, and duration.

3.1. Details of Factors

S1: customer participation: how actively customer took part in development work:

1 = very low; none

2 = low; passive; client defined or approved <30% of all functions

3 = normal; client defined and approved 30-70% of all functions

4 = high; active; client defined and approved all of most important functions, and over 70% of others

5 = very high; client participated very actively, thus most functions were slightly volatile and changes had to be made.

S2: staff availability: availability of software personnel during project:

1 = very low; big problems with key personnel availability; lots of simultaneous customer and maintenance responsibilities; special know-hold required

2 = low; personnel involved in some other simultaneous projects and/or maintenance responsibilities.

3 = normal; key members involve in only one other project.

4 = high; project members involved almost full-time.

5 = very high; qualified personnel available when needed full-time participation.

S3: standards use: level and use if standards:

1 = very low; standards developed during project.

2 = low; some standards, but not familiar ones; more must be developed for some tasks.

3 = nominal; generally known standards applied in environment before; some tailoring needed.

4 = high detailed standards applied in same environment for some time.

5 = very high; stable and detailed to team; use controlled.

S4: method use: level and use of methods:

1 = very low;no modern design methods; mostly meeting; used by individuals.

2 = low; use beginning; traditional concepts employed(structural analysis and design, top-down design etc).

3 = nominal; generally known methods used 4 = High Methods integrated in detail and most activities are covered. Support Existed. Used by everyone.

5 = Very high methods used during entire life cycle.

4. REFERENCES

[1] Conte, S.,Dunsmore, H.and Shen,V, *Software Engineering Metrics and Models*, Benjamin Cummings, Menlo Park CA, 1986.

[2] Guanrong Chen and etal, *Introduction to Fuzzy Sets, Fuzzy Logic and Fuzzy Control* Systems, CRC Press, 2000.

[3] E.M.Hall Managing Risk: *Methods for Software System Development* Addson-Wesley, 1998.

[4] Carr,D.and R.J.Kizior, "The Case for Continued COBOL Education," *IEEE Software*, **17**, Number 2, March/April 2000, pp.33-36.

[5] Daniel T Larose *Data Mining: methods and Models*, WIley Interscience 2006.

[6] John D Musa *Software Reliability* Engineering McGraw Hill 1999.

[7] Pace and Ronald Berry, Sparse Apatial Autoregressions, *Statistics and Probability Letters*, **33**, No 3, pp 291-297, May 5 1997, Data Set Available from StatLib.

[8] Richard A Johnson and Dean Wichern, *Applied Multivariate Statistical Analysis* Prentice Hall, Upper Saddle River, NJ, 1998.

[9] Joseph Hair, Rolph Anderson, Ronald Tatham and William Black, *Multivariate Data Analysis* , 4th Ed, Prentice Hall, Upper Saddle River, NJ, 1995.

[10] U.S.Census, Bureau, Urban Area Criteria for Census 2000, *Federal Register*, **67**, No 51, March 15 2002.

[11] Musa JD, "A Theory of Software Reliability and its Application", *IEEE Trans Software Eng 1975*; Se-1(3): 312-27.

[12] Marcio Rodrigo Braz, "Software Effort Estimation Based on use Cases", *IEEE 30th International Conference Proceedings COMPSAC'06* 2006.

[13] Fergal Mc Caffery, Philip S Taylor Gerry Coleman *Adept: A Unified Assessment Method for Small Software Companies*, Jan//Feb 2007 IEEE Software 24-31.

[14] Luigi Cerulo, Raffaelo Esposito, Maria Tortorella, Luigi Troiano RCOST–Research Centre on Software Technology Italy *Supporting Software Evolution by using Fuzzy Logic* 7th International IEEE Workshop Proceedings on Principles of Software Evolution (IWPSE'04).

[15] Xuemei Zhang, Hoang Pham, "An Analysis of Factors Affecting Software Reliability", *Journal of System and Software*, **50,** 2000, 43-56.

[16] Tsun Chowe, Dac-Buu Cao, "A Survey Study of Critical Success Factors in Agile Software Projects", *Journal of System and Software,* **81**, 2008, 961-971.

[17] Nasib S Gill Haryana, "*Factors Affecting Software Quality Management Revisited", ACM SIGSOFT Software Engineering Notes*, March 2005, **30**, No. 2, PP 1-4.

[18] Daniel T Larose, "Data Mining: Methods and Models", WIley Interscience 2006.