

Investigating the effects of factors on Software Development

Dr.Seetharam.K
Professor,
Dept. of CSE/IT,MeRITS,Udayagiri
Andrapradesh State

Chandrakanth G Pujari,
Asst.Professor,
Dept. of M.C.A. Dr.AIT Bangalore,
Research scholar, Sathyabama University
Chennai.

ABSTRACT

The current methods to measure the reliability of software are usually focused on large server based products. In these approaches, the product reliability is traditionally measured in terms of catastrophic failures, as the raw data is generally collected manually through service organizations. This filter out data on many types of operational failures. In this paper we discuss some of the key factors in determining reliability of such software products. Paper then discusses systems being used for measuring reliability of commercial software.

Qualitative/quantitative measurement of software reliability on following five factors. S1=Installation requirements, S2= staff analysis skills, S3=Staff application knowledge, S4=staff tool skill and S5= staff team skill. In this paper four different cases are carried out by means of principle component analysis. First analysis with size as predominant factor, Second analysis with effort as predominant factor, third analysis with duration as predominant factor, finally including all the three associated in the list of seven factors with software reliability performance. The analysis of variables is to identify the dimension that is latent. This can be considered in the phenomena of performance correlation. That is to study the effects in the developed principal components analysis approach.

Keywords:

Software reliability, Eigen values, fuzzy logic, software metrics.

1. INTRODUCTION:

The software development process becomes increasingly time consuming and expensive due to the complexity of software systems [12]. In the mean time, the need for the highly reliable software system is ever increasing [3,4].

How to enhance the reliability of the software systems and reduce the cost to an acceptable level becomes the main focus of the software industry. Methods of applying reliability to software development practice is highly desirable (Pham & Zhang, 2000)[14,15]. PCA (Principal Component Analysis) is used to explain the correlation structure of a set of predictor variables[2]. In this paper three different groups with each group

of five elements combinations of variables. The variables are related to software development. Here 50 validated projects data are taken for this analysis are used for software performance measurement[4,7,8,9].

2. ANALYSIS

In this analysis, we performed a survey and obtained empirical qualitative and quantitative data from managers who participate in the software development practice. Eight factors are involved in every phase of the software development process. Here four different groups are considered. The predictor variables are

- 1) Application size, S1, S2, S3, S4 & S5
- 2) Duration, S1, S2, S3, S4 & S5
- 3) Effort, S1, S2, S3, S4 & S5 where
- 4) Size, effort, duration, S1, S2, S3, S4 & S5 where
S1= Installation requirements
S2= staff analysis skills
S3=Staff application knowledge
S4= staff tool skill
S5= staff team skill

Five different levels are identified (see details of factors) for each variable separately by means of fuzzy logic [1,13]. Four analyses are carried out. The variables are standardized using normal distribution principles [6]. Covariance matrix is generated.

2.1 Size, S1, S2, S3, S4 & S5:

Principle components analysis matrix coefficients are given in table A1. Principal component Scores is given in table A2. Rows correspond to observations, columns to components. Hotelling's T-squared statistic for each observation is given in Table A3. The eigen values and proportion of variances explained by component is shown in table A4. Table A5 gives residuals obtained by retaining the principal components by the 10-by-5 data matrix. Rows correspond to observations, columns to variables[5,6].

2.2 Duration, S1, S2, S3, S4, & S5:

Principle components analysis matrix with Duration the main component is in Table B1. Principal component Scores is given in table B2. Rows correspond to observations, columns to components. Table B3 represents Hotelling's T-squared statistic for each observation. Table B4 shows the eigenvalues of the covariance matrix in latent and Percentage of variance. Eigen values show only Duration and S1 can be retained. S2, S3, S4 &

S5 have no role. Table B5 gives Residuals obtained by retaining the principal components by the 10-by-5 data matrix. Rows correspond to observations, columns to components.

2.3 Effort, S1, S2, S3, S4, & S5:

Table C1 gives Principle components analysis matrix with effort as the main component. Principal component Scores is given in table C2. Hotelling's T-squared statistic for each observation in table C3. Table C4 represents eigenvalue of covariance matrix in latent and percentage of variance. Eigen values show only effort and S1 are valid. S2, S3, S4 have no role.. C5 gives residuals of principal components

2.4 Size, Duration, Effort, S1, S2, S3, S4, S5:

Table D1 gives combined effect of S1, S2, S3, S4, S5 with size, duration and effort all in Principle components analysis matrix. Table D2 gives Principal component Scores is in and Hotelling's T square statistic for each observation in table D3. Table D4 represents eigenvalue of covariance matrix in latent and percentage of variance. Eigen values show similar to table A4. D5 deals with principal components Residuals.

3. CONCLUSION

Investigating the effects of factors in each (Four different groups) case is discussed in detail,

- PCA is done using a covariance matrix,
- Sum of the eigen values represents the number of variables entered into the PCA,
- Last component eigen value in tables A4, B4, C4 and D4 is very small in each case.
- The analysis of variables is to identify the dimension that is latent.
- Here retained all five values. Screen plot for these examples is shown.
- Hotelling's T-squared statistic for each observation and residuals from PCA are also given.
- Size, effort, duration and S1 are more predominant factors;
- Duration or efforts has no effect on S2, S3, S4 and S5.

4. DETAILS OF FACTORS:

Training needs for users and variants of platform

S1= Installation requirements

- 1=Very low; No training needs; < 10 users.
- 2= Low; some training; about 10 users; creation of basic data only minor
- 3=Nominal; typical training; 10-50 users. Some conversion of old data.
- 4=High; large scale training for several organizations; <1000 users; extra software for conversions; possible parallel runs ; several platforms
- 5=Very high; > 1000 users; Long expected life time; several user organizations; several different platforms.

S2= staff analysis skills; Analysis skills of project staff at kick off.

- 1 Very low; No experience in requirements analysis or similar projects.

- 2. Low; < 30 % of project staff with analysis and design experience in similar projects.

- 3 Nominal; 30-70% of project staff with analysis experience; one experienced member.

- 4. High; Most members of staff with experience in specification and analysis; Analysis professional in charge.

- 5. Very high; Project staff composed of first –class professionals. Members have strong vision and experience with requirements analysis.

S3= Staff application knowledge: Knowledge of application domain in project team.(supplier and customer)

- 1=very low; team application experience < 6 months on average.
- 2= low; application experience low; some members have experience; 6-12 months on average.

- 3=Nominal; Application experience good; 1-3 years on average.

- 4=High; Application experience good both at supplier and customer sites. 3-6 years on average. Business dynamics known.

- 5=Very high; both supplier and customer know application area well, including the business; > 6 years average experience.

S4= staff tool skill: Experience level of project team (supplier and customer) with development and documentation tools at project kick off;

- 1 = Very low; Team has no experience in necessary tools; team's average experience < 6 months.

- 2 = Low; Tools experience < average; Some members have experience with some tools; 6-12 months on average.

- 3 = Nominal; tools experience good in about half the team; some members know development and documentation tools well; 1-3 years on average

- 4 =High; most team members know tools well; some members can help others; 3-6 years on average.

- 5= Very high; team knows all tools well; support available for specific needs of project; >6 years average experience.

S5= staff team skill: Ability of project team to work effectively according to best project practices:

- 1=Very Low; scattered team; minimal project and management skills

- 2=Low; some members with previous experience on similar projects; not united as a group

- 3=Nominal: most members with experience on similar projects ; commitment on project goals good ; no motivation to utilize real team spirit

- 4=High; group very active and knows how to exploit team effectiveness

- 5=Very high; very anticipatory team; team can solve in an innovative way most personal and team problems; superior spirit

5. REFERENCES:

- [1] Guanrong Chen and etal. *Introduction to fuzzy sets, fuzzy logic and fuzzy control systems*. CRC Press, 2000.
- [2] Daniel T Larose *Data Mining: methods and models*. Wiley interscience 2006.
- [3] John D Musa *Software Reliability Engineering* McGraw Hill 1999
- [4] Conte, S, Dunsmore, H. and Shen, V., *Software Engineering Metrics and models*, Benjamin Cummings, Menlo Park CA, 1986.
- [5] R.Kelley Pace and Ronald Berry, sparse spatial autoregressions, *statistics and probability letters*, vol. 33 No.3,pp 291-297, May 5, 1997Data set available from stat lib <http://lib.stat.cmu.edu/datasets/houses.zip>.

- [6] Richard A Jhonson and Dean Wichern, *Applied multivariate statistical analysis*, Prentice Hall, upper Saddle River , NJ,1998.
- [7] Diane Hope, Corinna Gries, Weixung Zhu, William F .Fagan, Charles L.Redman, Nancy B. Grimm, Amy L. Nelson, Chris Martin, and Ann Kinzig, Socioeconomics drive urban plant diversity, *Proceedings of the National Academy of Sciences* vol. 100, No.15, pp.8788-8792, July 22, 2003
- [8] Carr, D. and R.J.Kizior, “*The case for continued COBOL Education*”, IEEE Software, Volume17, Number2 March/April 2000, pp.33-36.
- [9] Arranga.E., “*In COBOL’s Defense*”, IEEE Software, volume 17, number 2, March/April 2000, pp.70-75.
- [10] Kemerer.C.F. and S.Slaughter, “*An Empirical Approach to Studying Software Evolution* “, IEEE Transactions on Software Engineering, Volume 25, Number 4, July/August 1999, pp.493-509.
- [11]Kemerer.C.F, and S.Slaughter, “*Determinants of Software Maintenance Profiles: An Empirical Investigation*,” Software Maintenance Research and Practice, Volume 9, 1997, pp.235-251.
- [12] Gerg Goth, “*Software-as-a-service: The Spark That Will Change Software Engineering?*” IEEE distribute systems, July 2008.
- [13] Guanrong Chen and etal. *Introduction to fuzzy sets, fuzzy logic and fuzzy control* systems. CRC Press, 2000.
- [14] E.M.Hall Managing risk: Methods for software system Development Addison-Wesley; 1998.
- [15] Xuemei Zhang, Hoang Pham *An analysis of factors affecting software reliability*. The journal of systems and software pp43-56, 2000.