

IMPORTANCE OF ENVIRONMENTAL FACTORS AFFECTING SOFTWARE RELIABILITY

A. LOGANATHAN AND R. JEROMIA MUTHURAJ*

ABSTRACT. This paper presents the findings of empirical results from 25 participants from system engineers, programmers, testers, software development managers, and other people involved in software development. In 2000 Xuemei Zhang, Hoang Pham described Thirty-two factors which are affecting the software reliability. In this study another 2 factors are added which are affecting the software reliability and the study identifies the factors which have significant impact on software reliability.Relative weight, Correlation analysis and principal component analysis are used to identify the significant factors. The findings may have important implications for further research and it may be useful for software development.

1. Introduction

Software plays an important role all over the world. All human beings are using software in their daily life. So the importance of software is very high. And the reliability of software is also unavoidable. Software Reliability is defined as the probability of failure-free software operation for a specified period of time in a specified environment. Software reliability evaluation is playing an important role in software reliability engineering. The role of statistics is also very important in reliability estimation for software. There are many hardware reliability approaches but Software Reliability Modeling (SRM) work started in the early '70s with the inventive works of Jelinski and Moranda, Shooman and Coutinho. After that many works were done related to software reliability. Many software reliability models were constructed in parametric and non-parametric approaches. Some parametric models are Jelinski and Moranda De-Eutrophication Model (1972), Schick and Wolver ton Model, Goel and Okumoto Imperfect Debugging Model, Littlewood -Verrall Bayesian Model (1973), Goel-Okumoto Nonhomogeneous Poisson Process Model, Shooman Exponential Model, and etc. Some Non Parametric models are A Non-Parametric Order Statistics Software Reliability Model (1998), State Transition Model for Predicting Software Reliability (2007), and etc. The experts say that there are more than 225 software reliability models. But there is not even a single model that can be used in all situations. A model may work well for a set of certain software, but it may be completely off track for other kinds of problems. In 2000 Xuemei Zhang, Hoang Pham introduced a new way of estimating

²⁰⁰⁰ Mathematics Subject Classification. Primary 62N05; Secondary 62H25.

Key words and phrases. Software reliability, Relative weight method, Principal Component Analysis, Kendalls correlation .

^{*} Corresponding Author.

A. LOGANATHAN AND R. JEROMIA MUTHURAJ

software reliability with some environmental factors which are affecting the software. They have defined 32 environmental factors and they conducted a study to find the significant factors which are affecting the reliability of the software. The questioner has 32 variables with 8 scale measurements that 0 to 7. If the measure is 0 then the factor is not significant and if the measurement is 7 the factor is highly significant. They received the information from 23 software development practitioners from various software development companies. They use some statistical tools Relative weight method, Factor analysis, Correlation analysis and ANOVA. Finally the significant factors are discussed. In 2015 Mengmeng Zhu, Xuemei Zhang, and Hoang Pham revisit the 32 environmental factors and analyze their impact on software development and reliability based on a current survey to software development practitioners. In this study, the questioner is formed and performed a survey to get a quantitative and qualitative data from managers, software engineers, designers, programmers and testers who participate in the software development practice. Thirty four factors are involved in every phase of the software development process and the information about the background of survey participants are considered in this research. The Investigative analyses are used to identify the most significant factors with respect to software reliability and study the correlation of these factors. Statistical tools such as relative weighted method, Kendalls correlation analysis are applied to analyze these factors.

2. The objectives of this paper

The main objective of this work is to reinvestigate the factors influencing change. Software reliability environmental factors are proposed before two decades (2000 by Xuemei Zhang, Hoang Pham). The usage of internet is high now and the usage of the software is also more. Secondary objective is the buyer (customers) convenience how to get good software? Is a question. This study may give answer for that question.

3. Statistical methodologies

This study utilized several methods to analyze the ranking data. Relative weight method, Principal component analysis (PCA) and Kendall's correlation analysis.

3.1. Relative weight method. The relative weight method was used to obtain the final ranking for the factors. Let r_{ij} be the original ranking of the i_{th} factor on the j_{th} survey. As discussed in 2000(H. Pham). First normalize these r_{ij} such that

$$w_{ij} = \frac{r_{ij}}{\sum_{i=1}^{n} r_{ij}}$$

Where n is the number of factors on the j_{th} survey

Therefore $\sum_{i=1}^{n} w_{ij} = 1$ for all j. Then average these w_{ijis} to obtain the final weight for the i_{th} factor such that

$$w_i^* = \frac{\sum_{j=1}^l w_{ij}}{l}$$

Where l is the number of surveys used in this method. Based on these relative weights, we obtain the final weight for each factor.

IMPORTANCE OF ENVIRONMENTAL FACTORS AFFECTING SOFTWARE RELIABILITY

3.2. Kendall's correlation analysis. Kendall's Tau correlation coefficient is calculated from a sample of N data pairs (X, Y) a variable U as the ranks of X and a variable V as the ranks of Y. Kendall's Tau is then calculated from U and V using

$$\hat{\tau} = \frac{2(n_C - n_D)}{\sqrt{N(N-1) - T_X} - \sqrt{N(N-1) - T_Y}}$$
$$T_X = \sum_{i=1}^{S_X} (t_{(X)i}^2 - t_{(X)i})$$
$$T_Y = \sum_{i=1}^{S_Y} (t_{(Y)i}^2 - t_{(Y)i})$$

The parameter n_C is the total number of concordant pairs and n_D is the total number of discordant pairs.

3.3. Principal component analysis. The goals of PCA are to

- Extract the most important information from the data table.
- Compress the size of the data set by keeping only this important information.
- Simplify the description of the data set.
- Analyze the structure of the observations and the variables.

4. Data description

In this study the questioner is formatted with 34 variables and the factors which is affecting the software reliability. The respondents have to respond for each variable with the scale measurement 0 to 7. The participants are software development managers, system engineers, programmers, testers and administrators. Totally 25 numbers of participants are responded.



FIGURE 1. Data description

5. Finding & Results

5.1. Relative Weight Method Ranking. Results by the relative weight method are given in Table 1. The column named Normalized Priorities gives the contribution of each factor. For example, Programmer skill contributes approximately 4.06% (its relative weight 0.040609). Higher priority value indicates higher ranking. Since lower class rank implies decrease in magnitude of relative importance, software developers should then pay more attention to the factors with high ranks. Then final priority information can be used to guide the software development process of different applications. The table gives the list of all the environmental factors which are important for software development process and to work with the software. According to The scores which are given by the experts the significant factors are arranged in ascending order.

Rank	Factor Number	Factor name	Normalized priorities
1	F16	Programmer skill	0.040609
2	F8	Frequency of program specification change	0.040552
3	F23	Testing effort	0.039559
4	F22	Testing environment	0.038552
5	F1	Program complexity	0.038076
6	F25	Testing methodologies	0.037368
7	F6	Percentage of reused modules	0.036619
8	F26	Testing coverage	0.036325
9	F27	Testing tools	0.035769
10	F11	Requirements analysis	0.035574
11	F17	Programmer organization	0.034895
12	F24	Testing resource allocation	0.033364
13	F4	Amount of programming effort	0.032157
14	F10	Design methodology	0.031899
15	F20	Domain knowledge	0.031685
16	F13	Programmers experience	0.031536
17	F7	Programming language	0.031156
18	F3	Difficulty of programming	0.031105
19	F5	Level of programming technologies	0.030483
20	F9	Volume of program design documents	0.030157
21	F2	Program categories	0.029964
22	F18	Development team size	0.029749
23	F15	Development management	0.02923
24	F12	Relationship of detailed design to requirement	0.027203
25	F14	Work standards	0.024024
26	F19	Program workload (stress)	0.023285
27	F33	System software	0.022551
28	F28	Documentation	0.021772
29	F34	Random access memory	0.021184
30	F21	Human nature	0.02004
31	F29	Processors	0.014883
32	F30	Storage devices	0.013609
33	F31	Input/output devices	0.012697
34	F32	Telecommunication devices	0.012369

TABLE 1. Relative Weight Method Ranking

5.2. Kendall's correlation analysis. The purpose of performing correlation analysis is to observe the relation between variables and find out the strength and direction of this relationship. Having the knowledge of the correlation of the environmental factors (table 2)will provide a better understanding for software

IMPORTANCE OF ENVIRONMENTAL FACTORS AFFECTING SOFTWARE RELIABILITY

developers on resource allocation and testing efficiency during the software development and testing. Here in this study Kendall's correlation is used to see the correlated factors. Table 2 gives the result how the factors are correlated with other factors.

1 Program complexity Programmer skill 0.519 2 Program categories Difficulty of programming 0.591 3 Difficulty of programming Program categories 0.591 3 Difficulty of programming Program categories 0.591 4 Amount of programming effort Percentage of reused modules 0.438 5 Level of programming technologies Program more skill 0.439 6 Percentage of reused modules Amount of programming effort 0.459 7 Program design documents 0.591 10 Design methodology Testing anvorkload 0.438 11 Requirements analysis Level of programming effort 0.466 12 Relationship of detailed design to requirement Testing anvormment 0.754 13 Programmer skill Programmer skill 0.651 14 Work standards Programmer skill 0.512 15 Development management Testing methodologies 0.552 16 Programmer skill Program workload <th>Factor Number</th> <th>Factors</th> <th>Correlated factors</th> <th>Correlation</th>	Factor Number	Factors	Correlated factors	Correlation
1 Program categories Difficulty of programming 0.591 3 Difficulty of programming 0.591 Program categories 0.591 4 Amount of programming effort Program categories 0.466 5 Level of programming technologies Program workload 0.433 6 Percentage of reused modules Amount of program specification change 0.699 7 Programming language Difficulty of programming technologies 0.699 9 Volume of program specification change 0.489 9 Volume of programming technologies 0.699 10 Design methodology Programming technologies 0.699 11 Requirements analysis Domain knowledge 0.587 12 Relationship of detailed design to requirement Work standards 0.511 13 Programmer skill 0.654 0.511 14 Work standards Programmer skill 0.654 15 Development management Testing methodologies 0.521 16 Programmere skill Program		D 1	Programmer skill	0.519
2 Program categories Difficulty of programming 0.591 3 Difficulty of programming Program categories 0.591 4 Amount of programming effort Program categories 0.591 5 Level of programming technologies Program workload 0.433 6 Percentage of reused modules Amount of program mescification change 0.699 7 Programming language Difficulty of programming technologies 0.59 7 Program congram design documents Level of programming technologies 0.59 9 Volume of program design documents Level of programming technologies 0.59 10 Design methodology Testing resource allocation 0.64 11 Requirements analysis Domain knowledge 0.51 12 Relationship of detailed design to requirement Work standards 0.51 14 Work standards Program conglexity 0.51 15 Development management Program conglexity 0.513 16 Program workload Programmer sepretince 0.654	1	Program complexity	Design methodology	0.489
3 Difficulty of programming Programming nanguage 0.591 4 Amount of programming effort Programming nanguage 0.466 5 Level of programming technologies Program workload 0.459 6 Percentage of reused modules Amount of program specification change 0.699 7 Programming factorization change Level of program specification change 0.466 8 Prequency of program specification change 0.469 9 Volume of program modification change Level of programming technologies 0.591 10 Design methodology Programmeng technologies 0.587 11 Requirements analysis Domain knowledge 0.587 12 Relationship of detailed design to requirement Work standards 0.663 14 Work standards Program categorization 0.782 16 Programmer skill Program complexity 0.511 17 Programmer skill Programmers kaperience 0.6521 16 Program workload Amount of programmere skill 0.522 <	2	Program categories	Difficulty of programming	0.591
3 Difficulty of programming Programming language 0.466 4 Amount of programming effort Work standards 0.509 5 Level of programming technologies Program workload 0.438 6 Percentage of reused modules Amount of program specification change 0.699 7 Program specification change 0.466 8 Prequency of program specification change Level of programming technologies 0.59 9 Volume of program specification change Level of programming technologies 0.59 10 Design methodology Program complexity 0.466 11 Requirements analysis Domain knowledge 0.574 12 Relationship of detailed design to requirement Work standards 0.541 13 Programmer sexperience Programmer siziation 0.782 14 Work standards Programmer skill 0.654 15 Development team size Programmer sexperience 0.782 16 Programmer organization Programmer gramization 0.456 19			Program categories	0.591
Manual of programming effort Work standards 0.509 4 Amount of programming effort Percentage of reused modules 0.459 5 Level of programming technologies Program workload 0.433 5 Level of program specification change 0.699 6 Percentage of reused modules Amount of programming flort 0.459 7 Programming language Difficulty of programming technologies 0.699 9 Volume of program design documents Level of programming technologies 0.699 10 Design methodology Program complexity 0.466 11 Requirements analysis Domain knowledge 0.587 12 Relationship of detailed design to requirement Work standards 0.644 13 Programmer skill Programmer organization 0.683 14 Work standards Program complexity 0.511 15 Development management Testing methodologies 0.521 16 Program morganization Program more Experience 0.663 19 Program worgladad <td>3</td> <td>Difficulty of programming</td> <td>Programming language</td> <td>0.466</td>	3	Difficulty of programming	Programming language	0.466
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$			Work standards	0.509
4 Amount of programming elort Program workload 0.438 5 Level of programming technologies Program specification change 0.699 6 Percentage of reused modules Amount of programming effort 0.439 7 Programming language Difficulty of programming 0.466 8 Prequency of program specification change Level of programming technologies 0.699 9 Volume of program specification change Level of programming technologies 0.591 10 Design methodology Program complexity 0.489 11 Requirements analysis Domain knowledge 0.574 12 Relationship of detailed design to requirement Testing resource allocation 0.664 14 Work standards Programmer organization 0.782 15 Development management Testing resource allocation 0.436 16 Programmer skill Programmers Experience 0.782 17 Programmer organization Program workload 0.436 19 Program workload Program workload 0.4561 </td <td>4</td> <td>Amount of an amount in a offered</td> <td>Percentage of reused modules</td> <td>0.459</td>	4	Amount of an amount in a offered	Percentage of reused modules	0.459
	4	Amount of programming enort	Program workload	0.438
5 Level of programming technologies Volume of program design documents 0.59 6 Percentage of reused modules Amount of programming effort 0.469 7 Programming language Difficulty of programming effort 0.469 8 Frequency of program specification change Level of programming technologies 0.59 9 Volume of program design documents Level of programming technologies 0.59 10 Design methodology Program complexity 0.489 11 Requirements analysis Domain knowledge 0.574 12 Relationship of detailed design to requirement Testing resource allocation 0.64 13 Programmers Experience Programmer organization 0.782 14 Work standards Difficulty of programming 0.509 15 Development management Testing methodologies 0.552 16 Programmer skill Programmers Experience 0.654 17 Program workload Amount of program mogranization 0.456 19 Program workload Program mer organization	E	I and of any many in a tasky also inc	Frequency of program specification change	0.699
6Percentage of reused modulesAmount of programming effort0.4597Programming languageDifficulty of programming0.4668Frequency of program design documentsLevel of programming technologies0.6999Volume of program design documentsLevel of program complexity0.48910Design methodologyProgram complexity0.48911Requirements analysisDomain knowledge0.58712Relationship of detailed design to requirementTesting environment0.78113Programmers ExperienceProgrammer organization0.78214Work standardsDifficulty of programming0.50115Development managementTesting methodologies0.53216Programmer skillProgram complexity0.51917Programmer organization0.78218Development team sizeTesting coverage0.65420Domain knowledgeRequirements analysis0.53721Human natureProgram mer organization0.45623Testing environmentDecumentation0.31124Testing environmentDesign methodology0.75425Testing environmentDesign methodology0.75426Testing coverage0.65427Testing environmentDesign methodology0.75428DocumentationHuman nature0.31129ProcessorsHuman nature0.31129ProcessorsHuman n	5	Level of programming technologies	Volume of program design documents	0.59
7Programming languageDifficulty of programming0.4668Frequency of program design documentsLevel of programming technologies0.6999Volume of program design documentsLevel of programming technologies0.5910Design methodologyProgramming technologies0.58911Requirements analysisDomain knowledge0.58712Relationship of detailed design to requirementTesting resource allocation0.6413Programmers ExperienceProgrammer organization0.78214Work standardsRelationship of detailed design to requirement anagementProgrammer organization0.54115Development managementProgram complexity0.51916Programmer organizationProgrammers Experience0.65217Programmer organizationProgrammers Experience0.65219Program workloadProgrammer organization0.48620Domain knowledgeRequirements analysis0.58721Human natureProcessors0.65122Testing environmentDesign methodologies0.55223Testing environmentDesign effort0.65424Testing coverage0.5520.57425Testing doverageTesting coverage0.65426Testing coverageTesting effort0.65427Testing coverageTesting coverage0.55228DocumentationHuman nature0.65129Processors<	6	Percentage of reused modules	Amount of programming effort	0.459
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	7	Programming language	Difficulty of programming	0.466
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	8	Frequency of program specification change	Level of programming technologies	0.699
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	9	Volume of program design documents	Level of programming technologies	0.59
10Design methodologyTesting environment 0.754 11Requirements analysisDomain knowledge 0.587 12Relationship of detailed design to requirementTesting resource allocation 0.64 13Programmers ExperienceTesting resource allocation 0.64 14Work standardsDifficulty of programming 0.509 15Development managementTesting methodologies 0.552 16Programmer skillProgrammers Experience 0.6541 17Programmer organizationProgrammers Experience 0.6541 18Development team sizeTesting morkload 0.456 19Program workloadProgrammers Experience 0.6531 20Domain knowledgeRequirements analysis 0.587 21Human natureProgramming effort 0.458 22Testing environmentDecumentation 0.311 22Testing environmentDecumentation 0.311 23Testing environmentDesign methodologies 0.552 24Testing environmentDesign methodologies 0.551 25Testing methodologiesTesting coverage 0.654 26Testing coverageDocument anagement 0.552 27Testing environmentDesign methodologies 0.511 28DocumentationHuman nature 0.651 29ProcessorsHeationship of detailed design to requirement 0.654 29ProcessorsHuman nature 0.651 <	10	D. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	Program complexity	0.489
11Requirements analysisDomain knowledge 0.587 12Relationship of detailed design to requirementTesting resource allocation 0.64 13Programmer ExperienceProgrammer organization 0.782 14Work standardsProgrammer skill 0.6541 15Development managementTesting methodologies 0.552 16Programmer organization 0.6454 17Programmer organization 0.6541 18Development team sizeProgrammores Experience 0.6541 19Programmer organizationProgrammer Skill 0.552 20Domain knowledgeRelationship of detailed design to equirement 0.456 21Human natureProgram workload 0.456 22Testing environmentDesign methodology 0.754 23Testing environmentDesign methodology 0.754 24Testing environmentDesign methodology 0.754 25Testing coverageDevelopment team sizeDevelopment management 0.6523 26Testing coverageDevelopment management 0.552 27Testing coverageDevelopment management 0.654 28DocumentationRelationship of detailed design to requirement 0.651 29Presting coverageTesting effort 0.654 30Storage devicesTelecommunication devices 0.544 31Input/output devicesHuman nature 0.654 32Telecommunication devicesSt	10	Design methodology	Testing environment	0.754
12Relationship of detailed design to requirementTesting resource allocation0.6413Programmers ExperienceProgrammer organization0.78214Work standardsProgrammer skill0.54115Development managementTesting methodologies0.50916Programmer skillProgram complexity0.51917Programmer organizationProgram complexity0.51918Development team sizeTesting coverage0.52319Program workloadProgrammer organization0.45619Program workloadProgrammer organization0.45620Domain knowledgeRequirements analysis0.58721Human natureProcessors0.65422Testing environmentDesign methodology0.77423Testing coverage0.65424Testing coverage0.65425Testing coverage0.65426Testing coverage0.65427Testing coverage0.52328DocumentationHuman nature0.65429Processors0.65420Testing coverage0.65421Testing coverage0.65422Testing coverage0.65423Testing environmentDevelopment team size0.52326Testing coverageTesting fort0.65427Testing coverage0.6540.51128DocumentationHuman nature0.65129 <td>11</td> <td>Requirements analysis</td> <td>Domain knowledge</td> <td>0.587</td>	11	Requirements analysis	Domain knowledge	0.587
12Relationship of detailed design to requirementWork standards 0.541 13Programmers ExperienceProgrammer organization 0.782 14Work standardsDifficulty of programming 0.654 15Development managementRelationship of detailed design to requirement 0.541 16Programmer skillProgram complexity 0.519 16Programmer organizationProgrammers Experience 0.654 17Programmer organizationProgrammers Experience 0.654 18Development team sizeTesting coverage 0.523 19Program workloadAmount of programming effort 0.436 20Domain knowledgeRequirements analysis 0.587 21Human natureProcessors 0.654 23Testing effortTesting coverage 0.654 24Testing resource allocationRelationship of detailed design to requirement 0.654 25Testing methodologiesTesting coverage 0.654 26Testing coverageTesting effort 0.654 27Testing toolsTesting effort 0.654 28DocumentationHuman nature 0.651 29ProcessorsHuman nature 0.651 29ProcessorsHuman nature 0.654 30Storage devicesTelecommunication devices 0.544 31Input/output devicesHuman nature 0.654 32Telecommunication devicesStorage devices 0.544	10		Testing resource allocation	0.64
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	12	Relationship of detailed design to requirement	Work standards	0.541
13Programmers ExperienceProgrammer skill0.65414Work standardsDifficulty of programming0.50915Development managementTesting methodologies0.51916Programmer skillProgram complexity0.51917Programmer organizationProgrammers Experience0.65418Development team sizeTesting coverage0.52319Program workloadAmount of programming effort0.43620Domain knowledgeRequirements analysis0.58721Human natureInput/output devices0.45822Testing environmentDesign methodology0.75423Testing resource allocationRelationship of detailed design to requirement0.64425Testing coverage0.6510.51126Testing coverageTesting tools0.51127Testing toolsTesting methodologies0.51128DocumentationHuman nature0.65429ProcessorsHuman nature0.65421Human detailed design to requirement0.64523Testing resource allocationRelationship of detailed design to requirement0.65426Testing coverageTesting tools0.51127Testing toolsTesting methodologies0.51128DocumentationHuman nature0.65129ProcessorsHuman nature0.65430Storage devicesTelecommunication devices0.544 <t< td=""><td>10</td><td></td><td>Programmer organization</td><td>0.782</td></t<>	10		Programmer organization	0.782
14Work standardsDifficulty of programming0.50915Development managementRelationship of detailed design to requirement0.54116Programmer skillProgram complexity0.51917Programmer organizationProgrammers Experience0.65417Programmer organizationProgrammers Experience0.78219Program workloadO.4560.45619Program workloadProgrammer organization0.45620Domain knowledgeRequirements analysis0.58721Human natureInput/output devices0.65422Testing environmentDesign methodology0.75423Testing resource allocationRelationship of detailed design to requirement0.64124Testing resource allocationDevelopment team size0.52326Testing coverage0.6540.51126Testing coverage0.65427Testing coverage0.52328Documentation0.51129ProcessorsHuman nature0.65430Storage devicesTelecommunication devices0.51132Telecommunication devicesTelecommunication devices0.544	13	Programmers Experience	Programmer skill	0.654
14Work standardsRelationship of detailed design to requirement0.54115Development managementTesting methodologies0.55216Programmer skillProgram complexity0.51917Programmer organizationProgrammers Experience0.65418Development team sizeTesting coverage0.52319Program workloadAmount of programming effort0.45620Domain knowledgeRequirements analysis0.58721Human natureInput/output devices0.65422Testing environmentDesign methodology0.75423Testing effortCosta0.64424Testing effortTesting coverage0.65425Testing methodologiesDevelopment management0.55226Testing nethodologiesTesting coverage0.65427Testing coverageTesting coverage0.55228DocumentationHuman nature0.65429ProcessorsHuman nature0.65429ProcessorsHuman nature0.65429ProcessorsHuman nature0.65129ProcessorsHuman nature0.65430Storage devicesTelecommunication devices0.54431Input/output devicesHuman nature0.65432Telecommunication devicesStorage devices0.544	14	TI 7 1 4 1 1	Difficulty of programming	0.509
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	14	Work standards	Relationship of detailed design to requirement	0.541
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	15	Development management	Testing methodologies	0.552
16Programmer skillProgrammers Experience0.65417Programmer organizationProgrammers Experience0.78218Development team sizeTesting coverage0.52319Program workloadProgrammer organization0.45620Domain knowledgeRequirements analysis0.58721Human natureInput/output devices0.45822Testing environmentDevelopment team size0.65423Testing environmentDevelopment management0.65124Testing resource allocationRelationship of detailed design to requirement0.64125Testing coverage0.5230.55126Testing coverage0.6540.55227Testing coverage0.5520.51128Documentation0.3110.52329ProcessorsHuman nature0.65129ProcessorsHuman nature0.65130Storage devicesTelecommunication devices0.54431Input/output devicesHuman nature0.45832Telecommunication devices0.544			Program complexity	0.519
17Programmer organizationProgrammers Experience0.78218Development team sizeProgram workload0.45618Development team sizeTesting coverage0.52319Program workloadProgrammer organization0.45620Domain knowledgeRequirements analysis0.58721Human natureInput/output devices0.45822Testing environmentDesign methodology0.75423Testing effortTesting coverage0.65424Testing resource allocationRelationship of detailed design to requirement0.65125Testing coverageDevelopment team size0.52326Testing coverageDevelopment team size0.52327Testing toolsTesting effort0.65428DocumentationHuman nature0.31129ProcessorsHuman nature0.65130Storage devicesTelecommunication devices0.54431Input/output devicesHuman nature0.45832Telecommunication devicesStorage devices0.544	16	Programmer skill	Programmers Experience	0.654
$ \begin{array}{ c c c c c c } \hline Programmer organization & Program workload & 0.456 \\ \hline Program workload & Development team size & Testing coverage & 0.523 \\ \hline 19 & Program workload & Programmer organization & 0.456 \\ \hline 19 & Program workload & Amount of programming effort & 0.438 \\ \hline 20 & Domain knowledge & Requirements analysis & 0.587 \\ \hline 21 & Human nature & Input/output devices & 0.458 \\ \hline 22 & Testing environment & Design methodology & 0.754 \\ \hline 23 & Testing effort & Testing coverage & 0.654 \\ \hline 24 & Testing resource allocation & Relationship of detailed design to requirement & 0.648 \\ \hline 25 & Testing methodologies & Development team size & 0.523 \\ \hline 26 & Testing coverage & Development team size & 0.523 \\ \hline 28 & Documentation & Human nature & 0.654 \\ \hline 29 & Processors & Human nature & 0.651 \\ \hline 30 & Storage devices & Telecommunication devices & 0.458 \\ \hline 31 & Input/output devices & Storage devices & 0.458 \\ \hline 32 & Telecommunication devices & Storage devices & 0.544 \\ \hline \end{array}$			Programmers Experience	0.782
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	17	Programmer organization	Program workload	0.456
19Program workloadProgrammer organization0.45620Domain knowledgeRequirements analysis0.58720Domain knowledgeRequirements analysis0.58721Human natureInput/output devices0.6521Human natureInput/output devices0.6522Testing environmentDesign methodology0.75423Testing resource allocationRelationship of detailed design to requirement0.6425Testing nethodologiesTesting coverage0.65126Testing coverageDevelopment team size0.52327Testing toolsTesting effort0.65428DocumentationHuman nature0.31129ProcessorsHuman nature0.31129ProcessorsTelecommunication devices0.54431Input/output devicesHuman nature0.45832Telecommunication devicesStorage devices0.544	18	Development team size	Testing coverage	0.523
$ \begin{array}{ c c c c c c } \hline Program workload & Amount of programming effort & 0.438 \\ \hline \hline \\ 20 & Domain knowledge & Requirements analysis & 0.587 \\ \hline \\ 21 & Human nature & Input/output devices & 0.458 \\ \hline \\ 22 & Testing environment & Design methodology & 0.754 \\ \hline \\ 23 & Testing effort & Testing coverage & 0.654 \\ \hline \\ 24 & Testing resource allocation & Relationship of detailed design to requirement & 0.64 \\ \hline \\ 25 & Testing methodologies & Development management & 0.552 \\ \hline \\ 26 & Testing coverage & Development team size & 0.523 \\ \hline \\ 27 & Testing tools & Testing effort & 0.654 \\ \hline \\ 28 & Documentation & Human nature & 0.311 \\ \hline \\ 29 & Processors & Human nature & 0.65 \\ \hline \\ 30 & Storage devices & Telecommunication devices & 0.544 \\ \hline \\ 31 & Input/output devices & Storage devices & 0.544 \\ \hline \end{array}$	10		Programmer organization	0.456
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	19	Program workload	Amount of programming effort	0.438
21Human natureProcessors0.6521Human natureInput/output devices0.45822Testing environmentDesign methodology0.75423Testing effortTesting coverage0.65424Testing resource allocationRelationship of detailed design to requirement0.6425Testing methodologiesDevelopment management0.55226Testing coverageDevelopment team size0.51127Testing toolsTesting effort0.65428DocumentationHuman nature0.31129ProcessorsHuman nature0.6530Storage devicesTelecommunication devices0.54431Input/output devicesStorage devices0.544	20	Domain knowledge	Requirements analysis	0.587
21Human natureInput/output devices0.45822Testing environmentDocumentation0.31123Testing effortDesign methodology0.75424Testing resource allocationRelationship of detailed design to requirement0.6425Testing methodologiesDevelopment management0.55226Testing coverageDevelopment team size0.53327Testing toolsTesting effort0.65428DocumentationHuman nature0.31129ProcessorsHuman nature0.6530Storage devicesTelecommunication devices0.54431Input/output devicesStorage devices0.458			Processors	0.65
$\begin{tabular}{ c c c c c c } \hline & & & & & & & & & & & & & & & & & & $	21	Human nature	Input/output devices	0.458
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$			Documentation	0.311
23Testing effortTesting coverage0.65424Testing resource allocationRelationship of detailed design to requirement0.6425Testing methodologiesDevelopment management0.55226Testing coverageDevelopment team size0.51126Testing coverageDevelopment team size0.53327Testing toolsTesting methodologies0.51128DocumentationHuman nature0.31129ProcessorsHuman nature0.6530Storage devicesTelecommunication devices0.54431Input/output devicesStorage devices0.458	22	Testing environment	Design methodology	0.754
24Testing resource allocationRelationship of detailed design to requirement0.6425Testing methodologiesDevelopment management0.55226Testing coverageDevelopment team size0.52327Testing toolsTesting effort0.65428DocumentationHuman nature0.31129ProcessorsHuman nature0.6530Storage devicesTelecommunication devices0.54431Input/output devicesStorage devices0.458	23	Testing effort	Testing coverage	0.654
25 Testing methodologies Development management 0.552 26 Testing coverage Testing tools 0.511 26 Testing coverage Development team size 0.523 27 Testing tools Testing effort 0.654 28 Documentation Human nature 0.311 29 Processors Human nature 0.65 30 Storage devices Telecommunication devices 0.544 31 Input/output devices Human nature 0.458 32 Telecommunication devices 0.544	24	Testing resource allocation	Relationship of detailed design to requirement	0.64
25 Testing methodologies Testing tools 0.511 26 Testing coverage Development team size 0.523 27 Testing tools Testing methodologies 0.511 28 Documentation Human nature 0.311 29 Processors Human nature 0.65 30 Storage devices Telecommunication devices 0.544 31 Input/output devices Storage devices 0.458	25		Development management	0.552
26Testing coverageDevelopment team size0.52327Testing toolsTesting effort0.65428DocumentationHuman nature0.31129ProcessorsHuman nature0.65430Storage devicesTelecommunication devices0.54431Input/output devicesHuman nature0.45832Telecommunication devices0.544	25	Testing methodologies	Testing tools	0.511
26Testing coverageTesting effort0.65427Testing toolsTesting methodologies0.51128DocumentationHuman nature0.31129ProcessorsHuman nature0.6530Storage devicesTelecommunication devices0.54431Input/output devicesHuman nature0.45832Telecommunication devices0.544			Development team size	0.523
27Testing toolsTesting methodologies0.51128DocumentationHuman nature0.31129ProcessorsHuman nature0.6530Storage devicesTelecommunication devices0.54431Input/output devicesHuman nature0.45832Telecommunication devicesStorage devices0.544	26	Testing coverage	Testing effort	0.654
28 Documentation Human nature 0.311 29 Processors Human nature 0.65 30 Storage devices Telecommunication devices 0.544 31 Input/output devices Human nature 0.458 32 Telecommunication devices Storage devices 0.544	27	Testing tools	Testing methodologies	0.511
29 Processors Human nature 0.65 30 Storage devices Telecommunication devices 0.544 31 Input/output devices Human nature 0.458 32 Telecommunication devices Storage devices 0.554	28	Documentation	Human nature	0.311
30 Storage devices Telecommunication devices 0.544 31 Input/output devices Human nature 0.458 32 Telecommunication devices Storage devices 0.544	29	Processors	Human nature	0.65
31 Input/output devices Human nature 0.458 32 Telecommunication devices Storage devices 0.544	30	Storage devices	Telecommunication devices	0.544
32 Telecommunication devices Storage devices 0.544	31	Input/output devices	Human nature	0.458
	32	Telecommunication devices	Storage devices	0.544

TABLE 2. correlation analysis

5.3. Principal Component Analysis. Principle component analysis (PCA) is a statistical analysis that can be used to reduce the dimensionality of a data set consisting of a large number of interrelated variables. The idea is that we can use smaller dimension set of principle components to capture the characteristics of the larger data set and provide a concise yet critical principle components for software developers. In this study for all the 34 factors PCA procedure is applied. 100% of variation is explained by 34 factors. The PCA is explaining how the variation is explained. If we convinced with 65% of variation then we can take only top 6

A. LOGANATHAN AND R. JEROMIA MUTHURAJ

IADLE J. I UA RESUL	TABLE	3.	PCA	Result
---------------------	-------	----	-----	--------

Comp	omponent Total Variance Cumulative % of Variance		F1	F2	F3	F4	F5	F6	F7					
1		6.869	20.202	2	20.202			067	.356	.692	.281	.124	.473	089
2		3.702	10.888	3	31.091			204	.066	.134	148	342	039	.453
3	3 3.324 9.778 40.868					152	.420	.026	.478	.236	258	.008		
4		2.968	8.729		4	9.597		068	012	.144	.161	499	474	255
5		2.257	6.638		5	6.235		276	331	077	578	.153	009	.017
6		2.099	6.174		62.409				.018	.323	.247	.208	328	.479
7		2.037	5.991		68.401			.071	134	156	.057	.214	.349	.069
8		1.693	4.980		73.381				.310	.433	224	331	.014	.108
9		1.424	4.189		77.569				092	.064	.065	.113	.301	141
10)	1.321	3.885		8	1.454		003	278	.033	.212	.206	.266	.024
1	1	1.111	3.269		8	4.723		.179	382	.083	.083	.256	116	.113
1:	2	.984	2.895		8	7.617		110	162	174	178	053	068	.386
1:	3	.882	2.595		9	0.213		.070	141	.202	.057	.230	.174	049
F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22
.467	829	160	.430	141	125	145	.335	.140	660	159	464	.600	030	.133
.226	320	.344	122	190	315	393	.095	349	422	318	365	.539	.022	.481
098	.253	.140	.524	561	481	140	339	.255	052	008	.028	421	415	.306
017	.265	371	.528	.445	.214	.210	046	.149	200	.258	.386	.066	.560	.226
042	.641	.367	.463	.369	364	.350	.189	211	.178	325	.061	.095	.011	157
259	.224	.171	.094	.223	.290	078	.389	.131	121	209	341	404	.305	123
.153	.330	.323	100	020	.221	338	250	173	.495	.345	022	.101	.229	297
.242	.023	.213	131	.067	033	.555	306	.275	.403	073	164	091	051	.210
021	064	169	284	026	446	.076	.093	117	066	126	218	357	.313	.356
.066	.209	002	127	171	.107	.084	458	.153	237	260	.009	.265	.204	030
.126	014	283	.016	287	085	.185	.135	.070	.270	147	.297	115	332	102
.218	069	.254	.013	170	008	.069	.294	.060	160	.054	.127	083	.070	.041
053	.008	.157	.044	.004	.132	.188	.182	.048	234	.302	.039	.194	029	.356
1	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32	F33	F34		
4		1												

F23	F24	F25	F26	F27	F28	F29	F30	F31	F32	F33	F34
371	446	395	605	696	282	349	.710	.688	.830	.551	.043
.487	.226	010	061	.646	.530	.228	.170	027	.132	.713	114
523	.222	318	.097	.466	295	128	030	.054	244	.396	.655
031	.162	284	.277	.167	.478	.307	.351	.259	.256	044	.196
.122	.033	.245	012	079	078	175	.208	.049	242	.264	.212
.104	.081	.243	063	.139	.069	122	234	.007	.042	.112	.077
214	393	147	.147	.344	.397	.001	032	033	.348	.147	.284
123	.018	.069	044	.062	049	103	042	205	.394	023	073
.154	179	.028	.372	090	.114	243	.163	.189	.127	183	.302
097	.557	.365	058	062	006	.170	.021	.039	.091	144	.141
.124	.038	.005	.379	.163	.081	.171	.215	037	.214	.170	325
105	.086	215	.146	102	063	.200	189	146	.134	327	.239
224	152	.040	.002	.073	.002	099	.199	263	147	.021	112

components from the above table for the further analysis. Likewise we can choose or take components according to our convenience. About 20% of the variation can be explained by the first component, 10% of the variation can be explained by the second component, and so on.

6. Conclusion

The main objective of this study is to find the significant factors of software development process which is affecting the reliability of the software. In this study two factors are added with the existing factors. 34 factors are considered for the analysis by relative weight method the significant factors are listed. Correlated factor are studied and listed. Principal component analysis is made for all the 34 factors. It may be useful for software developers and all those who related with software. If they implement this result they may get good and reliable software.

References

- 1. Barghout, M. and Littlewood, B.: A non-parametric order statistics software reliability model, *Journal of Software Testing, Verification and Reliability* **8(3)** (1998) 113-132.
- Goel, A. L.: A software error detection model with applications, in Proc. ACM Annu. Tech. Conf., ACM, Washington, DC (1978) 496–500
- Goel, A. L.: Software Reliability Models: Assumptions, Limitations and Applicability, *IEEE Transactions of Software Engg.*, SE-11(12) (1985) 1411–1423.
- Goel, A. L. and Okumoto, K.: An analysis of recurrent software failures in a real-time control system,, in Proc. ACM Annu. Tech. Conf., ACM, Washington, DC, (1978) 496–500.
- Jelinski, Z. and Moranda, P.: In Statistical Computer Performance Evaluation, in: W. Freiberger, Ed. New York: Academic press (1972) 465–484.
- Littlewood, B. and Verrall, J. L.: A Bayesian reliability growth model for computer software, *Appl. Statist.*, **22** (1973) 332–346.
- Patwa, S. and Malviya, K.: A Survey on Factors Affecting Testing Techniques in Object Oriented Software, International Journal of Applied research on Information Technology and Computing, 5 2014 78–85.
- 8. Pham H.: System Software Reliability, Springer Series in Reliability Engineering.
- 9. Teng, X. and Pham, H.: A new methodology for predicting software reliability in the random field environments, *IEEE Transactions on Reliability* **55(3)** (2006) 458–468.
- Zhang, X. and Pham, H.: An analysis of factors affecting software reliability, *Journal of Systems and Software* 50(1), (2000) 43–56.
- Zhang, X. Shin, M. Y. and Pham, H.: Exploratory analysis of environmental factors for enhancing the software reliability assessment, *Journal of Systems and Software* 57 (2001) 73–78.
- Zhang, X., Zhu M Y and Pham, H: A comparison analysis of environmental factors affecting software reliability, *Journal of Systems and Software* 109 (2015) 150–160.

A. LOGANATHAN: DEPARTMENT OF STATISTICS, MANONMANIAM SUNDARANAR UNIVERSITY, TIRUNELVELI, TAMILNADU - 627012, INDIA *E-mail address: aln_*24@*rediffmail.com*

R. JEROMIA MUTHURAJ: DEPARTMENT OF STATISTICS, MANONMANIAM SUNDARANAR UNIVER-SITY, TIRUNELVELI, TAMILNADU - 627012, INDIA *E-mail address: jeromstat@gmail.com*

E-mail address. Jeromistal@gmail.com