Software Reliability: Metrics

Durga Patel
AIM & ACT
Banasthali University
Tonk,Rajasthan

Pallavi
AIM & ACT
Banasthali University
Tonk ,Rajasthan

ABSTRACT

For the critical business application, continuous availability is the requirement, and software reliability is an important component of continuous application availability. Reliability implies probability. Program may contain no error and its reliability is unity. If the program contains error then its reliability is zero. Need of complex system is increased more rapidly. In early 1970s Software also become a matter of concern primarily due to a continuing increase in the cost of Software relative to hardware in both the development and the operational phase of the system. Software is essential in instrument for transforming a discrete set of inputs into a discrete set of output. Since, Software is produced by humans; the finished product is often imperfect. Therefore, Software Reliability is important and measuring technique to detect the failure. There has lot of work done in field of software reliability estimation.

Keywords

Software Reliability (SR), software reliability metrics, software measurement, and complexity metrics

1. INTRODUCTION

According to IEEE, "Reliability is defined as the ability of a system or component to perform its required function under a given period of time and conditions". Software reliability is considered as failure free operation. It covers all the categories such as Analysis and design, Coding, Testing and Hardware systems. According to the study, in July 2013 done by different authors Reliability is defined in three categories that are improvement, modeling and measurement. The Software Assurance Technology Center (SATC) at NASA tried to apply Models and Metrics to improve quality and reliability of software product. According to the definitions given by IEEE and NASA, Software reliability is composed of Error prevention, Fault detection and removal and Measurements to maximize reliability. In this paper, we are going to discuss about the various factors which affect the Software Reliability.

Software reliability

Software reliability is considered as one of most important assessment processes during software development. Software reliability is defined as stability or life of the software system with different properties. Reliability is referring as 'how well software is met up the requirements of the client. Software reliability is a measure in planning and controlling the resources during the development process, so that high quality software can be developed. [1]

Expression of SWR:

The expression of Software reliability is based on probability and is given by probability of success at time t, Ps (t).

$$R(t) = Ps(t) = 1 - Pf(t) = 1 - F(t)$$

$$= 1 - f(x)\, dx$$

Where R (t) =Reliability of software at time t

Ps (t) = Probability of failure

Pf(t) = Probability of failure.

F(t) = Failure distribution function.

f(x) = Failure density function.

Density function is not convenient for study of failure data and we normally consider hazard rate or failure rate Z (t). SWR in terms of hazard rate z (t) is given as

$$R(t) = \exp[- z(x)\, dx]$$

This expression shows that the reliability of software increased with passage of time.

2. LITERATURE REVIEW

According to SATC, they have introduced the metric for measuring the reliability of a software product. Accordingly they have decided to cover those areas where error prevention technique and metrics is applied to impact the reliability i.e.: requirements, coding and testing. They have given their own metrics in the form such as:

Requirement Reliability metrics

Requirement analysis is an important factor while considering any software. Any phrases such as TBD (to be determined) and TBA (to be added) since the lack of this phrases may have a negative impact on the design. The requirements should be in stable form so that the designing and implementation part should proceed. There are mainly three standard formats which are given by: IEEE, DOD and NASA.

The requirements must be complete, structured and easy to apply. Due to the correctness in the requirement many software industry have been aided for the creation and management of software. The software developed by them is: Automated Requirements Measurement (ARM) to scan the text of requirement specification and search for specific words and phrases.

Those words indicate the document's quality as requirements specification and have been applied to 56 NASA specification document. As a result 7 measures were developed: Lines of Text, Imperatives, Continuances, Directives, Weak phrases, incomplete and options. In this the structure is evaluated by identifying the requirements and by hierarchical numbering them.

Design and Code Reliability metrics

Since the design and coding of all the software is not same. If the module is more complex it will be more difficult to understand and has a direct impact on the software's quality. Size is one of the oldest form for software measurement as it is also a quality indicator. This in turn had an impact on software maintainability and on the basis of complexity there are:

1. Statement count: According to the abstract viewpoint, the more detail that an entity posses; the

more difficult it sis to understand. Due to its simplicity it is used worldwide.

2. Halstead Software Science: In this all the operators and operands are taken in to consideration and based on that value :

n1=number of unique operators

n2=number of unique operands

N1=total number of operators

N2=total number of operands

Then program Volume is:

$V = (N1+N2) \log_2 (n1+n2)$

Programming effort is calculated as:

E = V2/V*

Where,

V* is the minimum possible volume for program.

This has certain drawback as the most effort is required to produce the initial part of program.

3. McCabe's Number or Cyclomatic Complexity

McCabe ([Mccab 76]) has defined the complexity of a program to be:

$v = e - n + 2p$

Where

e = number of edges in a program flow graph

n = number of nodes

p = number of connected components

It can be calculated by counting the number of independent paths. This method has been used by SATC.SATC has found the relation between the size and complexity and indicated that the modules with high complexity and large size is less reliable as compared to modules with low size and high complexity. It also has a drawback as for many different complex program the complexity can be same.

Testing Reliability metrics

Testing metrics must take two approaches: the first one include evaluation of test plan and the other one is related to the evaluation of the number of errors in code and rate of finding/fixing them. In the first approach it is functional specific in which multiple cases are generated .In this test case, belong to one system state and test function on the basis of requirement. The objective is that if it pass the test than that required functionality will be added in the product. In this each of the test cases must be tested once and some of them will be tested multiple times as they are involved in different ways in the software. In the event of finding an error SATC has developed musa model to compute nonlinear approximation to cumulative errors. In addition to relate it to manpower efforts modified musa model has been used which uses the integral of a Rayleigh function.

As now a days developing software has become a more complex and time consuming task. The need for highly reliable system is ever increasing. The software development life cycle consist of five phases: analysis, design, coding, testing and operation. At each phase there are many different factors which can affect the quality of the final product. For the complex task there are limitations so factors such as subsystem configuration, working languages, operational profile and so on should be considered. As a result 13 companies participated to identify the 13 factors that may have an impact on software reliability. Their result is based on the 3 hypothesis which are:

a. The significance of impact of 32 factors on Software Reliability Assessment is of the same level.

b. People playing different role in the software development have same opinion on significance of 32 factors.

c. People developing software for different applications have the same opinion on the importance of the 32 factors.

The 32 factors have been collected and have been clustered in 5 groups on the basis of general, Analysis and Design, Coding, Testing and Hardware System .As the factors may seem externally independent but internally they are correlated to each other. So, these are stated as:

Table 1:Classification Of Factors Of Metrics

General	Program complexity, Program category, Difficulty of Programming, Amount of programming effort, Level of programming technologies, Percentage of reused module, Programming language.
Analysis and Design	Frequency of program specification change, Volume of program design documents, Design methodology, Requirements analysis, Relationship of detailed design to requirement, Work standards, Development management
Coding	Programmer skill, Programmer organization, Development team size, Program workload (stress), Domain knowledge, Human nature (mistake and work omission)
Testing	Testing environment (duplication of production), Testing effort, Testing resource allocation, Testing methodologies, Testing coverage, Testing tools, Documentation
Hardware System	Processors, Storage devices, Input/output devices, Telecommunication devices, System software

3. SCOPE AND MOTIVATION

Software Engineering is a young discipline, so its theories, methods, models and technique need to be fully developed and assessed and the 32 factors stated above can be used to give highly reliable software which is more in demand.. Software Engineering measurement is one of the areas in Software Engineering where researchers are active since more than 30 years. Therefore, Software Reliability has become more and more important. Software Reliability is a measuring technique for defect that cause software failure. This is the hot topic for research in the field of Software engineering.

4. OBJECTIVE

The number of metrics have been proposed during the past years for accessing the reliability of a software system. In this research work we access the performance of metrics factor during the Software Development Cycle. In this research paper we have taken all the 32 factors and ranking is given according to their performance.

5. RESULTS AND DISCUSSION

The 32 factors have been statistically analyzed using the SPSS tools in which various parameters have been considered.

GENERAL:

Table 2 : Correlation Matix

		var1	var2	var3	var4	var5	var6	var7
Correlation	var1	1.000	.343	.233	.240	.218	.116	.162
	var2	.343	1.000	-.004	.262	.177	.423	.289
	var3	.233	-.004	1.000	.264	.141	.141	.015
	var4	.240	.262	.264	1.000	.284	.295	.010
	var5	.218	.177	.141	.284	1.000	.492	.168
	var6	.116	.423	.141	.295	.492	1.000	.455
	var7	.162	.289	.015	.010	.168	.455	1.000

Table 3: Kmo & Bartlett's Test

Kaiser-Meyer-Olkin	
Measure of Sampling Adequacy	0.606
Bartlett's Test of Sphericity	
Approx. Chi Square	94.973
Df	21.000
Sig	

Table 4 : Communalities

	Initial	Extraction
Var 1		0.373
Var 2		0.478
Var 3		0.557
Var 4		0.540
Var 5		0.412
Var 6		0.690
Var 7		0.599

Extraction method:Principal Component analysis

Table 5: Total Variance Explained

Component	Initial Eigen Values			Extraction Sums Of Squared Loadings			Rotations sums of squared loadings
	Total	% of variance	Cumulative%	Total	% of variance	Cumulative %	Total
1	2.420	34.568	34.568	2.42	34.568	34.56	2.139
2	1.230	17.568	52.136	1.23	17.568	52.13	1.780
3	0.958	13.689	65.825				
4	0.836	11.938	77.763				
5	0.707	10.104	87.867				
6	0.528	7.547	95.414				
7	0.321	4.586	100.00				

Extraction Method: Principal Component Analysis.

a. When components are correlated ,sum of squared Loadings cannot be added to obtain a total variance.

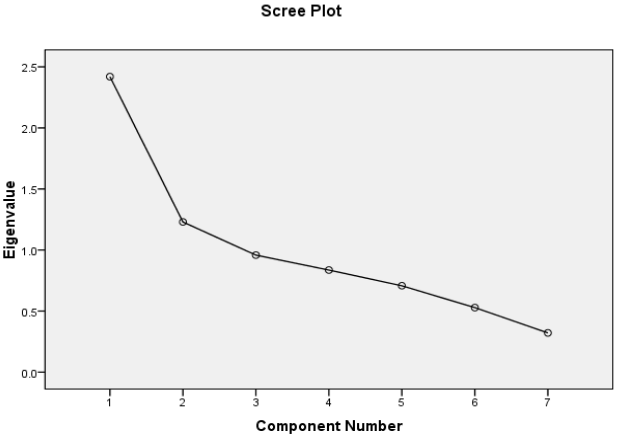

Fig 1: Plotting Of General Metrics Wrt. Eigen Value & Component Number

Table 6:Component Matrix

	Component	
	1	2
Var 6	0.776	
Var 2		
Var 5		
Var 4		
Var 1		
Var3		
Var 7		

Extraction Method:Principal Component Analysis

a. 2 components Extracted

Table 7:Rotation Method

	Component	
	1	2
Var 7	0.794	
Var 6	0.793	
Var 2		
Var 5		
Var 3		0.768
Var 4		0.706
Var 1		

Extraction Method:Principal Component Analysis Rotation Method: Oblimin with Kaiser Normalization.

Table 8:Structure Matrix

	Component	
	1	2
Var 6	0.822	
Var 7	0.735	
Var 2		
Var 5		
Var 4		0.729
Var 3		0.717
Var 1		

Extraction Method:Principal Component Analysis
Rotation Method: Oblimin with Kaiser Normalization

Analysis

Table 9:Correlation Matrix

	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7
Var 1	1.000	0.295	0.093	0.270	0.208	0.213	0.367
Var 2	0.295	1.000	0.184	0.272	0.407	0.607	0.494
Var 3	0.093	0.184	1.000	0.479	0.482	0.335	0.162
Var 4	0.270	0.272	0.479	1.000	0.595	0.131	0.462
Var 5	0.208	0.407	0.482	0.595	1.000	0.322	0.496
Var 6	0.213	0.607	0.335	0.131	0.322	1.000	0.523
Var 7	0.367	0.494	0.162	0.462	0.496	0.523	1.000

Table 10:Kmo & Bartlett's Test

Kaiser-Mayer-Olkin Measure of Sampling Adequacy	0.674
Bartlett's Test of Spherecity	
Approx Chi square	187.520
Df	21.000
Sig	

Table 11 : Communalities

	Initial	Extraction
Var 1	1.000	0.298
Var 2	1.000	0.695
Var 3	1.000	0.628
Var 4	1.000	0.735
Var 5	1.000	0.710
Var 6	1.000	0.651
Var 7	1.000	0.656

Extraction Method: Principal Component Analysis

Table 12 :Total Variance Explained

Component	Initial Eigen Values			Extraction Sums Of Squared Loadings			Rotation Sums of squared Loadings
	Total	% of variance	Cumulative %	Total	% of variance	Cumualtive %	Total

1	3.169	45.265	45.265	3.169	45.265	45.265	2.696
2	1.206	17.224	62.489	1.206	17.224	62.489	2.443
3	0.923	13.183	75.672				
4	0.662	9.454	85.126				
5	0.457	6.527	91.653				
6	0.371	5.299	96.953				
7	0.213	3.047	100.00				

Extraction Method: Principal Component Analysis

a. When components are correlated sums of squared loadings cannot be added to obtain a total variance.

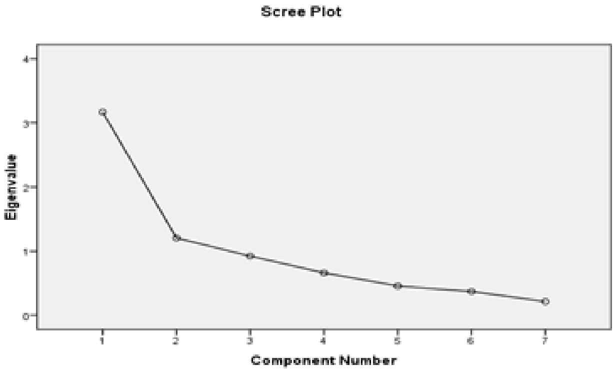

Figure 2:Plotting Of Analysis Factors Wrt To Eigen Values & Component Number

Table 13:Component Matrix

	Component		
	1	2	3
Var 5	0.792		
Var 1	0.772		
Var 6			
Var2		0.794	
Var 3			
Var 4			

Extraction Method:Principal Component Analysis

a. 3 components Extracted

Table 14:Pattern Matrix

	Component		
	1	2	3
Var 6	0.879		
Var 1	0.820		
Var 5			
Var 2		0.917	
Var 4			-0.894
Var 3			

Extraction Method:Principal Component Analysis Rotation Method :Oblimin With Kaiser normalization

a. Rotation covered in 10 iterations

Table 15:Structure Matrix

	Component		
	1	2	3
Var 1	0.844		
Var 6	0.827		
Var 5			
Var 2		0.915	
Var 4			-0.857
Var 3			

Extraction Method:Principal Component Analysis Rotation method:Oblimin with Kaiser normalization

Table 16: Component Correlation Matrix

Component	1	2	3
1	1.000	-0.001	-0.255
2	-0.001	1.000	-0.013
3	-0.255	-0.013	1.000

Extraction Method: Principal Component Analysis Rotation Method : Oblimin With Kaiser normalization

Testing

Table 17: Correlation Matrix

	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6
Var 1	1.000	-0.122	0.289	0.183	0.500	0.491
Var 2	-0.122	1.000	0.193	-0.167	-0.194	-0.124
Var 3	0.289	0.193	1.000	0.343	0.394	0.172
Var 4	0.183	-0.167	0.343	1.000	0.320	0.060
Var 5	0.500	-0.194	0.394	0.320	1.000	0.299
Var 6	0.491	-0.124	0.172	0.60	0.299	1.000

Table 18: Kmo & Bartlett’s Test

Kaiser –Meyer-Olkin Measure of Sampling Adequacy	0.633
Bartlett’s Test of Sphericity	84.579
Approx Chi square	15.000
Df	
Sig	

Table 19:Communalities

	Initial	Extraction
Var 1	1.000	0.722
Var 2	1.000	0.887
Var 3	1.000	0.726
Var 4	1.000	0.773
Var 5	1.000	0.637
Var 6	1.000	0.725

Extraction Method: Principal Component Analysis

Table 20:Total Variance Explained

. Components	Initial Eigen Values			Extraction Sum of squared loadings			Rotation Sums of Squared Loadings
	Total	% of Variance	Cumulative %	Total	% of variance	Cumulative %	Total
1	2.285	38.080	38.080	2.285	38.080	38.080	1.964
2	1.188	19.793	57.873	1.188	19.793	57.873	1.160
3	1.047	17.445	75.319	1.047	17.445	75.319	1.697
4	0.599	9.981	85.300				
5	0.477	7.954	93.245				
6	0.405	6.755	100.00				

Extraction Method: Principal Component Analysis When components are correlated, sum of squared loadings cannot be added to obtain a total variance

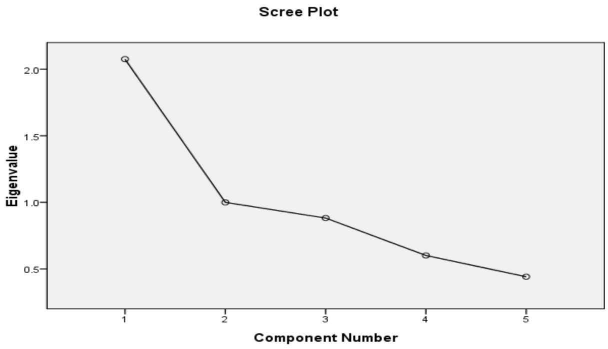

Figure 3:Plotting Of Testing Wrt To Eigen Value And Component Number

Table 21: Component Matrix

	Component
	1
Var 2	0.800
Var 5	0.714
Var 1	0.681
Var 3	0.663
Var 4	

6. ACKNOWLEDGMENT

I would like to take this opportunity to express my gratitude to all those people who have in various ways helped us including the college students. My sincere thanks goes to Dr. Saurabh Mukherjee(Associate Professor,Department of Computer Science, Banasthali Vidhyapeeth) for sharing their wisdom, knowledge and expertise to better understand the issues related to this work.

7. REFERENCES

[1] Tariq Hassain Sheakh, Vijaypal Singh, "Taxonomical Study of Software Reliability Growth Models",International Journal of Scientific Research Publications, Vol.2, Issue 5, pp-1-3 May 2012

[2] Chandrasekhar Rajaraman, Michael R. Lyu ,Reliability and Maintainability Related Software Coupling Metrics in C++ Programs

[3] Xuemei Zhang, Hoang Pham ,An analysis of factors affecting software reliability The Journal of Systems and Software 50 (2000), page no. 43-56.

[4] Norman F.Schneidewind Methodology for validating software metrics, , IEEE transaction on Software Engineering, vol.18, No. 5 May 1992 page no. 410-422.

[5] Dr. Linda Rosenberg Ted Hammer, Jack Shaw.Software metrics and reliability,