A SSS SCHEME TO INCREASE THE RELIABILITY IN SOFTWARE SYSTEMS

J. Arthy[1], H. Vishnupriya[2], R. G. Suresh Kumar[3]

[1, 2.3] *Assistant Professor, Department of Computer Science and Engineering,*
Rajiv Gandhi college of Engineering and Technology, Puducherry, (India)

ABSTRACT

A failure free operation is necessary for the software systems. Due to the increasing data and the real time constraints, the task failed to achieve the successful completion and for to produce the outcome of the problem. Many methods were proposed where the good outcome is reached but it is tedious in the sense of time and measurements taken throughout the process. To overcome all those things, the propose SSS scheme reduce the manipulation and it thoroughly go through the entire process and by that process it tends the information to the new process. This process produces good results for the taken inputs. In this method, the three components acts as a procedure to increase the reliability. The Start-up analyse the incoming process and the strength-up evaluates the new errors and the stater resolves all the errors while the process enters to the execution phase. SSS increase the reliability and the risk of failures and errors are reduced in the SSS scheme.

Keywords: SSS, Start-up, Strength-up, Stater, System Remodelling

I. INTRODUCTION

Software Reliability is defined as the probability of failure-gratuitous software operation for a designated period of time in a designated environment. Software reliability is different from hardware reliability because time is not a major constraint. It will not modify over time unless transmuted or upgraded transpired frequently. There are many software quality features such as usability, maintainability etc.… Reliability is one of them and it is very hard to procure it because it leads to a high degree of involution when the software application size is sizably voluminous. The involution of an application is inversely cognate to reliability and directly cognate to quality. Good projects are emerging from good management such as time, cost and development. Software reliability consist three components: modeling, quantification and enhancement. Reliability modeling refers the optimized model which is ascertained by a system testing. There are sundry estimation techniques to quantify the reliability, enhancing the reliability is the process of incrementing capability of software during testing and implementation.

There are sundry reasons behind software failures such as errors, interpretation faults, incompetence, testing and other quandaries. Design faults withal affect the reliability of software. The quantification of software reliability thoroughly depends on manipulation and calculation and so physical presage is not possible. There are some worst situations where the error appears without any caveat. For example, the inputs of a program additionally affect the software in the situations like redundancy, interference and overlapping. By analyzing the above

issues the standard testing and immensely colossal testing is examined to amend software reliability. But there are no standard methods other than some logic structures and calculations.

There is not a simple method to quantify software reliability. If the programmer or utilizer doesn't understand the software system or application then it becomes very rigorous to quantify it. Most of the software metrics not have a prevalent definition or methods. There are many metrics taken into consideration such as product, process, fault and failure metrics. These metrics avail the designer to quantify reliability indirectly. There is no standard way of counting the application other than LOC (Lines Of Code) or LOC in thousands (KLOC). Software operating environments are different for every application and so it withal affects the reliability .There are two issues which infringe the software reliability. They are Control dependence and data independence.

The hardware reliability is jaded but the software reliability is conceptual and document. To enhance software reliability first the quantification and amelioration of the metrics are initialized. At the next step the cost, effort, time and other set of involution metrics must be low. There are many authentic time examples of software failure such as fortuitous change of function when fault input to the system, encountered, misinterpretation of requisites etc., All the above discussed issues as certain that the reliability of a software is an capricious one. So there is a desideratum for a good presage method of software reliability. To enhance reliability the following steps to be done (i) standardizing data amassing methods (ii) documenting again and again (iii) inter-rater reliability.

II. RELATED WORK

To perform the reliability operations the existing algorithms are very tedious to manipulate. Upon various approaches the easy and efficient means of increasing the performance is not easily available. The simple approach is not use for complicate operations. In the literature, there are some works for procuring reliability models. The main disadvantage of this approach is that no distinction is made between different tests, and the fact that these different tests cover different possible faults. The main postulation utilized by these models in treating the fault rectification process is that the rate of fault rectification is proportional to the number of faults to be redressed, betokening that the expected cumulative number of redressed faults is proportional to the expected cumulative number of detected faults with constant delay. All the methods have some posits and calculations. Ingunn Myrtveit, Erik Stensrud [5] (2005) develops a research procedure with the study of software prediction models. Kapil Sharma et al. [4] (2010) derived a deterministic model which is evaluated by distance. Ahmed et al. [1] (2010) software application method is verified by several codes and programs. Yousif A. Bastaki [9] (2012) develop a method to increase the reliability which is very user interactive. Mohd. Anjum, Md. Asraful Haque, Nesar Ahmad [7] (2005) proposed a set of twelve comparison criteria and assigned a weight to rank the software reliability growth models. Eduardo Oliveria Costa et al. [3] (2010) obtained a good curve for reliability by time and cost which is implemented by genetic programming. Costa et al. [2] (2010) developed a reliability model for the same approach. Onishi et al. [6] (2007)analysis a failure and Mean Time to improve the reliability. Manfred Broy [8] (2007) discusses the automotive software engineering with the root and issues of automotive industries as processes, methods, tools, models, product structures.

III. SYSTEM MODEL

For the above discussed problems, the SSS (Starter, Strength-up, and Stater) method is introduced. In the SSS scheme there are several checkpoints to recognize and modify the corrupted and mistaken data. This method

performs minute scanning and so the errors and performance degradation existing work is considerably decreased. We will discuss the operation of this powerful SSS tool. The below description of SSS scheme is given below in fig 1.1.

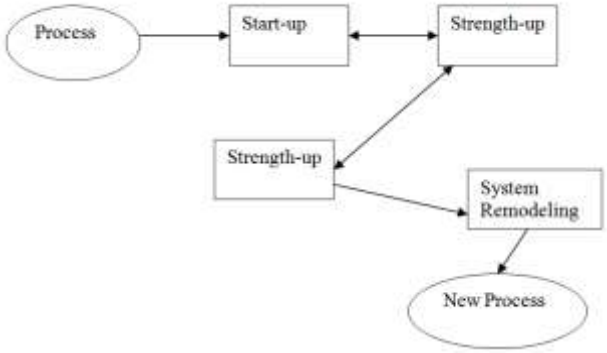

Fig.1: Overview of SSS

The SSS scheme performs the process as the program but actually it is a system. This system is very compatible with all environments. It suits all processor including the 2-way superscalar and 4-way superscalar. The processor contains numerous processes which are directly embedded to the SSS scheme. The below figure 2 depicts the schematic view of the SSS scheme.

The imported process initially enters into a startup stage where the starting work will process. The several checkpoints presents inside a start-up are (i) Load Estimation (ii) Possible Error Estimation (iii) Potent Estimation. The load estimation measures the load (i.e) the content, the running time, LOC etc…. The load estimation measures the process by integrating all the process. After the Load estimation the processor the possible error detection mechanism is the powerful method technology which is implemented by any high level language. This possible error method finds out all the possible errors in the processes. After this method the processor will enters to a potent estimation block. In this block the withstanding capability of all processor are measured for the purpose of error tolerance. After, the start up the process enters into a stater section.

Stater is a flexible one which composed of some mathematical terms and solutions. It acts as a interactive dictionary of our system. The outcome of the stater itself increases the reliability of the process in most of the cases. But the real time scenario changes system by system, case by case and even data by data. We cannot predict accurate outcome of a process in some cases. In order to overcome this we need in depth analysis of a system also a evaluation of individual components. The startup and stater are interconnected and so the operation is fast and quick. After the method of stater the processes enters into a strength-up section.

The strength up is a two way block and it is a special method to find out the uncoverable errors in the tasks. The possible error detection method in a startup complete the process as it is possible. But to enhance the reliability in a task we are including the block strength-up. The add on functionality in a strength up process is Audit maintain. It keep track of all records for the future process also. The major advantage and the powerful functionality in strength up block is a connection between the stater (i.e) the new errors are also stored in the stater and hence the future processor updated errors will detect easily. When the processor will not enter into strength up means then it will run quickly.

After the completion of the SSS scheme the processes are remodeled which is increased functionality and increased reliability one. The future section will discuss the increased reliability with the proven data.

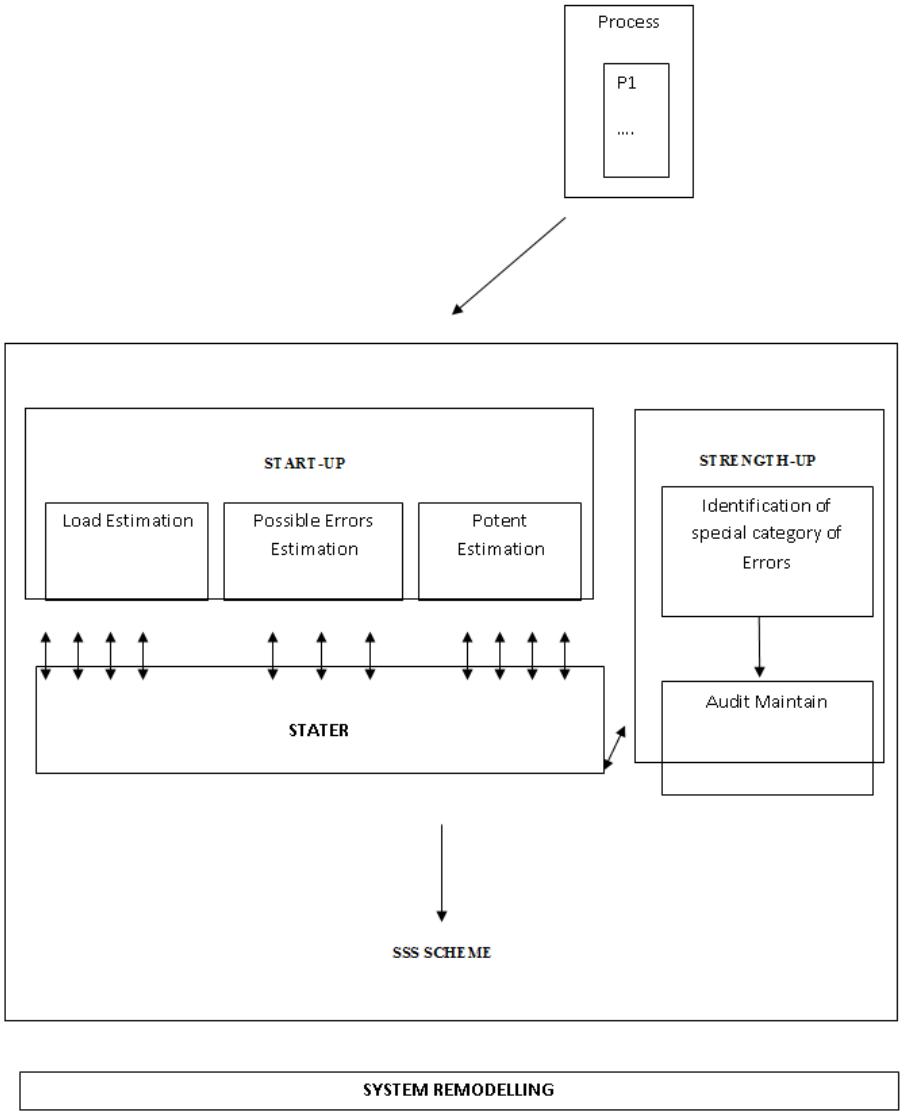

Fig.2: SSS System Model

IV. EXPERIMENTS AND RESULTS

To evaluate the performance initially we are taken several programs. The below table 1.2,3 explains the various phases of SSS in the considered process P1 to P7 and give the summarization of the entire method.. All the processes are simple programs which are executed as projects.

The table 1 shows the possibility of reliability but before the strength up process in which the load, Error and potent are estimated. To enhance the reliability here the new errors are estimated and mentioned in table 2. The table 3 shows the system remodeling which is the final outcome of the SSS scheme an the graph is drawn comparing with SSS and without SSS.

Load Estimation	Error Estimation	Potent Estimation
120	10	<5
140	22.5	<10
162	32	<16
125	15	<5

138	22	<10
142	25	<10
125	15	<5

Table 1: Start-up

Process	Error Estimation	Special Errors(LOC)
P1	10	15
P2	22.5	215,216
P3	32	138
P4	15	20,81
P5	22	15
P6	25	10
P7	15	11,35, 72

Table 2:Strength-up

Process	Error Estimation(Without SSS)	Error Estimation(with SSS)
P1	10	0.001
P2	22.5	0.1
P3	32	0.2
P4	15	0.001
P5	22	1.001
P6	25	0.12
P7	15	0.12

Table 3:System Remodelling

All the data are chosen randomly from various source codes and the reliability is measured inherently without any enhancer process. For the table 1 the corresponding graph is drawn which is given in Fig 3. The graph shows the reliability modeling with the fault profile.

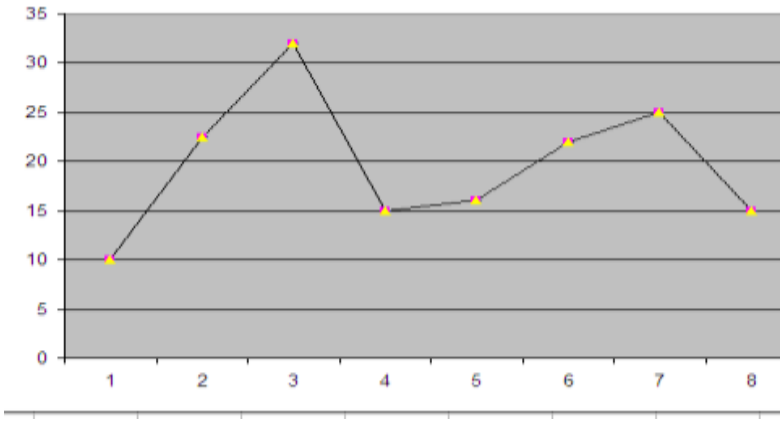

Fig.3: Reliability without SSS

The above graph is a model with reliability enhancer. By performing the necessary enhancer i.e (SSS) the below graph Fig 4 is drawn where the reliability is enhance and it is proved.

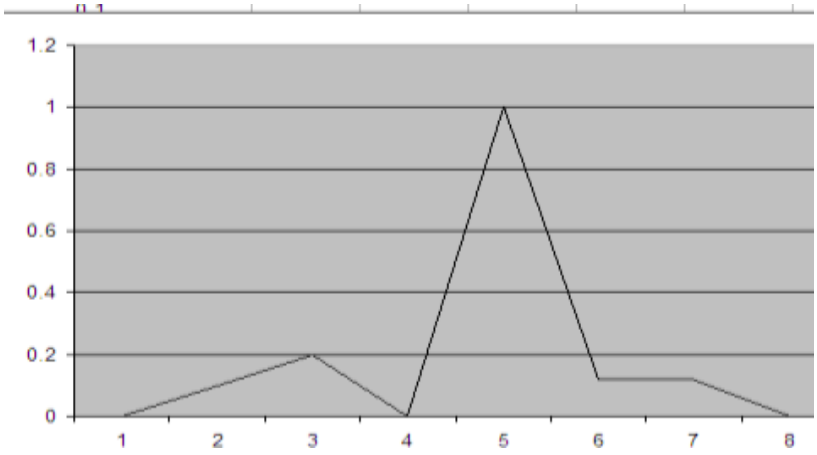

Fig.4: Reliability with SSS

V. CONCLUSION

The SSS Scheme is an optimized model to increase the reliability. SSS model automatically estimates the load, errors and potent. The audit maintain phase of this model regularly update the start-up phase which is increase the reliability to the new models. Increasing the reliability is very tedious and it requires complicate mathematical functions. But in this model, by considering the enormous projects itself detects the errors and repair the process. This method gives a good results and it is proven in the above tables. All the above experiments induce through all the phases of the SSS scheme and the reliability is increased. This method is not affect the output and reacts with the sensitive failure optimistically. There is a possibility to improve this model which will produce no error and achieves a full optimal reliability model. Therefore, SSS is a good reliability model to increase the reliability with respect to cost, time and effort.

REFERENCES

[1] Ahmed Patel, Liu Na, Rodziah Latih, Christopher Wills,Zarina Shukur and Rabia Mulla "A Study of Mashup as a Software Application Development Technique with Examples from an End-User Programming Perspective" Journal of Computer Science 6 (12): 1406-1415, 2010.

[2] Costa, E.O., A.T.R. Pozo and S.R. Vergilio, "A genetic programming approach for software reliability modeling", IEEE Trans. Reliability,2010

[3] Eduardo Oliveira Costa, Aurora Trinidad Ramirez Pozo, and Silvia Regina Vergilio A Genetic Programming Approach for Software Reliability Modeling, IEEE TRANSACTIONS ON RELIABILITY, VOL. 59, NO. 1, MARCH 2010.

[4] Kapil Sharma , Rakesh Garg, Nagpal C. K, Garg R. K, " Selection of Optical software Reliability Growth Models Using a distance Based Approach" IEEE Transactions on Reliability , Vol 59, No. 2, June 2010.

[5] Ingunn Myrtveit, Erik Stensrud, "Reliability and Validity in Comparative Studies of Software Prediction Models, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 2005.

[6] J. Onishi, S. Kimura, R.J.W. James, and Y. Nakagawa, "Solving the Redundancy Allocation Problem with a Mix of Components Using the Improved Surrogate Constraint Method," IEEE Trans. Reliability, vol. 56, no. 1, pp. 94-101, Mar. 2007.

[7] Mohd. Anjum, Md. Asraful Haque, Nesar Ahmad, Analysis and Ranking of Software Reliability, Models Based on Weighted Criteria Value, I.J. Information Technology and Computer Science, 02, 1-14, 2013

[8] Manfred Broy, Ingolf H. Kru¨ger, Alexander Pretschner, and Christian Salzmann, Engineering Automotive Software, Proceedings of the IEEE | Vol. 95, No. 2, February 2007

[9] Yousif A. Bastaki, A Framework for Teaching Programming on the Internet: A Web-Based Simulation Approach. Journal of Computer Science, 8: 410-419, 2012.