Refinement of Reliability by Improving Reliability Estimation Tool

Mohit Surati[1], Bhavesh Tanawala[2] and Prashant Swadas[3]

[1] Department of Computer Engineering, B.V.M Engineering College, Anand, Gujarat, India
mysurati@gmail.com

[2] Department of Computer Engineering, B.V.M Engineering College, Anand, Gujarat, India
Bhavesh.tanawala@bvmengineering.ac.in

[3] Department of Computer Engineering, B.V.M Engineering College, Anand, Gujarat, India
pbswadas@bvmengineering.ac.in

Abstract

In this era of technology, most of the systems are software based system which can be any safety-critical system as well. Therefore it is necessary to monitor the quality of the software on constant basis. This gave upswing to the concept of software reliability as it directly affects the quality of any software product. Reliability is a term which has been defined in early 90's but to due lack of research there were very few standardized tool available which can estimate the software reliability in quantitative terms. CASRE (Computer Aided Software Reliability Estimation) was one of the most standardized tool among them, but still there were room for improvement in the existing CASRE tool. This paper focuses on improving the CASRE.

Keywords: CASRE, Software Reliability, SRGM, Running Arithmetic Average, Laplace Trend Test

1. Introduction

We all are optimistic people, so we will always think positive about anything. The same positive perspective applies in the case of software as well. Generally, all people believe that software is always correct and therefore they blindly trust any software product. But some lethal consequences of software bugs in real-time scenario contradict with the earlier statements.

Each and every industry faces a lot problem due to a software bug. If we take the example of space research and aeronautics industry, the mars climate orbiter crashed in Sep'99 because of a silly mistake i.e. wrong units in the program [11]. In the case of airline industry, Airbus 230 in 1998 face a lot of damage due the cryptic and misleading output displayed by its tracking software [11]. A fatal case of software error took place in US Defence on June 3, 1980 where North American Aerospace Defence Command (NORAD) reported that the U.S. was under missile attack & the report was traced to a faulty computer circuit that generated incorrect signals which cost a humongous damage If the developers of the software responsible for processing these signals had taken into account the possibility that the circuit could fail, the false alert might not have occurred [11]. An analysis of 2002 by the National Institute of Standards and Technology shows that software bugs cost the US economy $59.5 billion every year [10]. How much lethal the consequences of software bug can be? To answer this question we have one real-time example. In 1994 in Scotland, a Chinook helicopter crashed and killed all 29 passengers due to small system error [10].

After observing these consequences, researchers started to think that whether the software is reliable or not? This gave rise to the concept of Software Reliability. According to IEEE Software Reliability can be defined as: "The ability of the system or the component to perform its required functions under stated conditions for a specified period of time" [15]. Afterwards many reliability estimation tools were designed but very few of them were standardized.

CASRE was one of them. The term CASRE stands for Computer-Aided Software Reliability Estimation tool which is a copyrighted by NASA (National Aeronautics and Space Administration) [7]. But currently it only works on WINDOWS XP version. So in this paper we are proposing a improved version of CASRE tool which can

work on any version of WINDOWS which are released after XP.

2. Background Theory

[A]. Software Reliability:

According to ANSI, **"**Software Reliability is defined as the probability of failure-free software operation for a specified period of time in a specified environment" [6]. Therefore we can say that reliability plays a crucial role in enhancing the overall quality of any software product.

[B]. Software Reliability Metrics:

Software reliability metrics are used to quantitatively express the reliability of the software product [17]. Different type of application demands different type of software reliability metrics. Reliability metrics are quantitative indicator for assessing and sanctioning the reliability.

[C]. Software Reliability Growth Model (SRGM):

A software reliability growth model is one of elementary method for assessing software reliability in quantitative terms. These models try to predict the reliability from the tested datasets. For any software to operate in reliable terms, it has to examine constantly by testing and debugging. This effect can be very easily modeled through the use of Software Reliability Growth Models (SRGM).

[D]. CASRE Tool:

CASRE stands for Computer-Aided Software Reliability Estimation Tool. It is a PC-based tool developed in 1993 by the Jet Propulsion Laboratories and copyrighted by NASA (National Aeronautics and Space Administration) [7]. The latest version of CASRE is CASRE 3.0 which available on the website of Open Channel Foundation [7]. (https://www.openchannelsoftware.com/projects/CASRE_3.0)

[E]. Trend Test:

Before mapping any kind of SRGM to the failure data, it is preferable to check whether the given failure data exhibits any reliability growth or not. The process of determining the reliability growth/decline is known as trend test or trend analysis. CASRE tool facilitate the users with two type of trend test: Running Arithmetic Average and Laplace Trend Test.

[F]. Filters:

There are mainly 5 types of filters used in the existing CASRE tool each has its own significance.

(i) Shaping and Scaling filters: It is used to change shape of the data in order to predict the failure.

(ii) Time Unit Changing filter: It is used to change the time unit of the failure data.

(iii) Smoothing filter: It is used to remove any kind of noise from the failure data.

(iv) Severity filter: It is used for forming the subsets of failure data based on its severity.

(v) Round filter: It is used to "round off" the failure data to the nearest whole number

3. Research Methodology

Here we have proposed the improved version of existing CASRE (Computer-Aided Software Reliability Estimation) tool which we have named as ISRET (Improved Software Reliability Estimation Tool). ISRET can work on any version of WINDOWS which is released after XP. Fig. shows the algorithmic representation of the proposed work.

Working steps of ISRET tool:

1) Check if the failure data is available. If it is available start the ISRET and open the failure data; otherwise create a set of failure data and apply it to ISRET.

2) Apply Bartlett Window filter to the failure data.

3) Apply trend test to determine if reliability growth exhibits or not. ISRET facilitate the users with two types of trend test: Running Arithmetic Average and Laplace Trend Test.

4) After applying trend test if the failure data exhibits any reliability growth apply any of the following SRGM: Exponential Model or Log-Logistic Model.

5) If the failure data does not show any reliability growth terminate the process and stop the tool as there is no trend possible.

6) After applying any of the appropriate SRGM view and analyze the model outputs and compare the results.

7) Plot the results in both Time-Between-Failure (TBF) and Failure Count (FC) format and analyze the reliability.

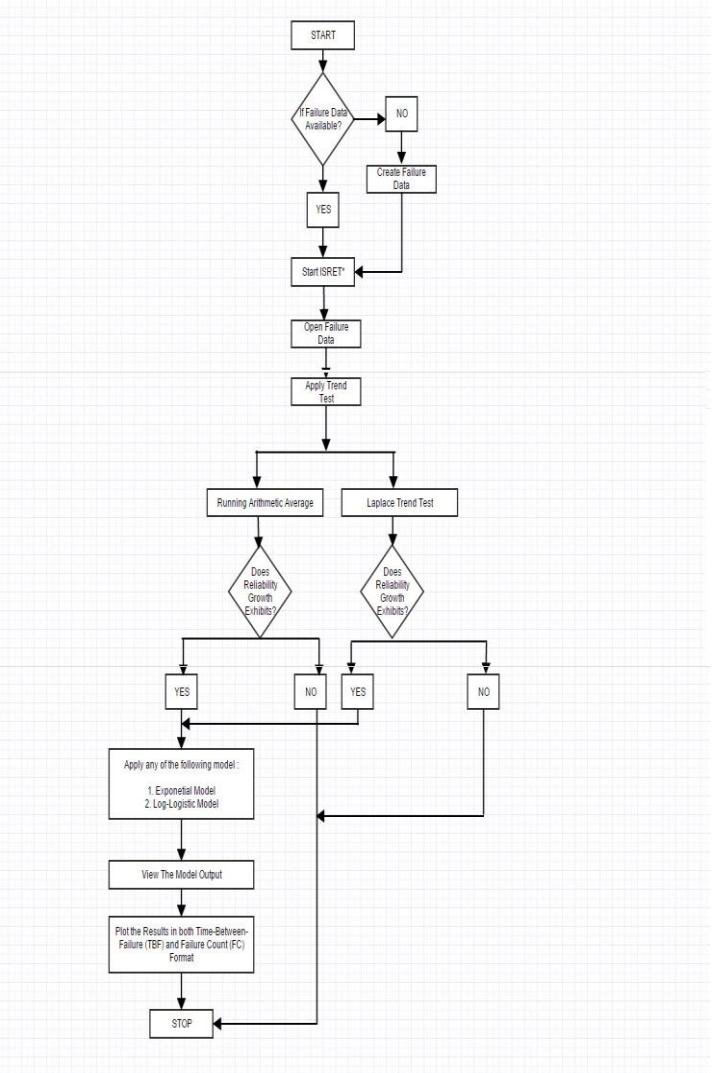

Fig. 1 Proposed System Flowchart.

4. Results

As per our proposed system we are going to develop an improved system of CASRE tool which we have called as ISRET (Improved Software Reliability Estimation Tool). We have implemented steps of the proposed system in MATLAB for time-between-failure format. For this purposed we have taken one standard dataset and collected the results which are as follows:

STEP 1: Original Data Generation and Plotting

- The original data is plotted in terms of Failure Number v/s Time-Between-Failure.

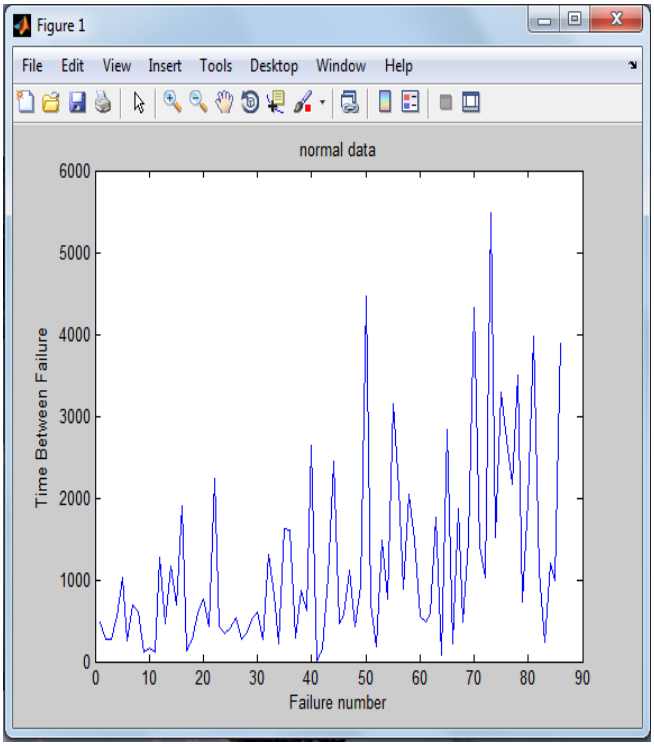

Fig.1: Original Data

STEP 2: Applying Modified Bartlett-Hann Window Filter

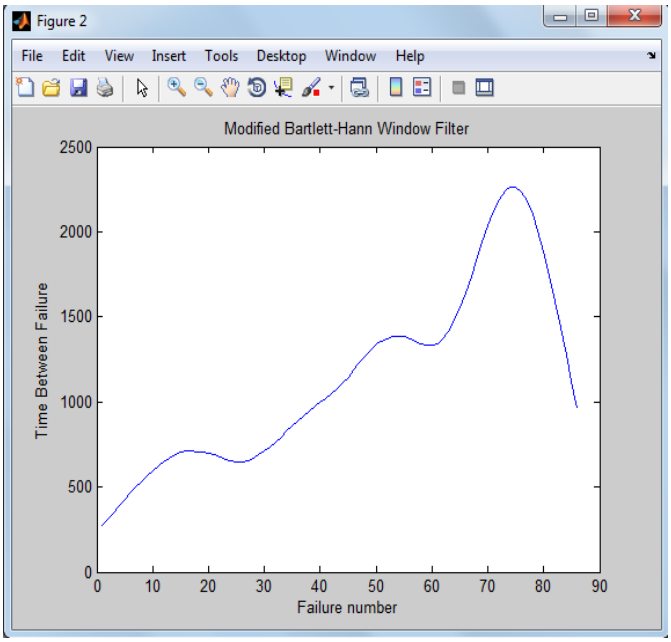

Fig.2: Filtered Data

- The filtered data is plotted in terms of Failure Number v/s Time-Between-Failure. The modified Bartlett-Hann window filter is applied on the original data to remove the noise from the failure and smooth the original data so that it is easy to predict the reliability.

STEP 3: Applying Laplace Trend Test to the Filtered Data

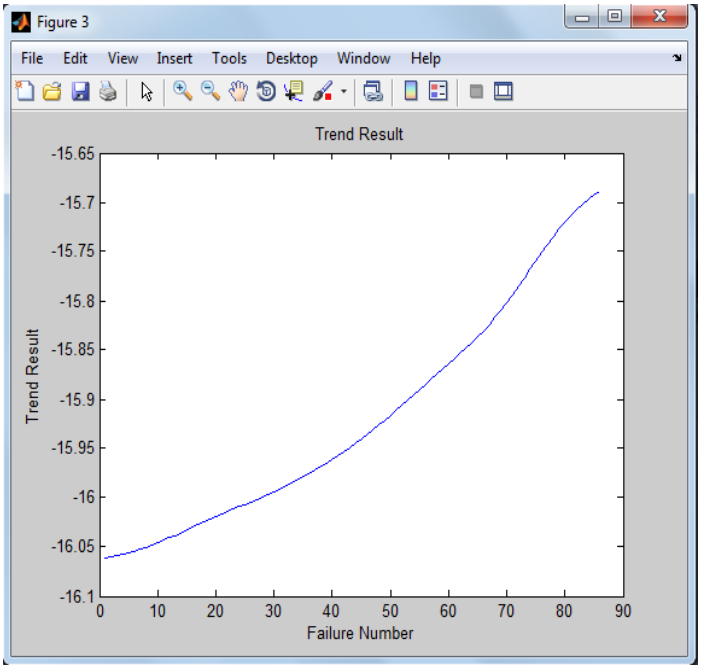

Fig.3: Trend Test Result

- The Laplace Trend Test is the most standard trend test for predicting the behavior of the reliability i.e. whether the reliability is increasing or decreasing or showing no trend. For this purpose we apply trend test on filtered data and plot the results it in terms of Trend Result v/s Failure data.

5. Conclusion and Future Work

After analysing all the research work related to CASRE (Computer-Aided Software Reliability Estimation) tool, we have proposed an improved version of CASRE tool which we named as ISRET (Improved Software Reliability Estimation Tool). After analysing the result comparison we

can conclude that the reliability estimates obtained through ISRET are much more accurate and efficient than CASRE tool. Another advantage of ISRET is that it can work on any version of WINDOWS released after XP (service pack 2).

Acknowledgments

It is a matter of great pleasure for me to get this opportunity of expressing my deep and sincere gratitude to those who has always helped me throughout my work. I heartily thank my guides Mr. Bhavesh A. Tanawala & Mr. P.B.Swadas for providing me their valuable guidance and constant support in my work. They guide me with their excellent knowledge in the area of "Software Engineering." I am also thankful to my friend Mr. Dhavalkumar H. Joshi for his constant cooperation, help and guidance throughout this journey. I would also like to thank all my colleagues who have helped me in some way or the other. I would like to thank my parents & family member for their love and support to build my moral during the work.

References

[1] Michel R. Lyu and Allen Nikora, "CASRE-A Computer Aided Software Reliability Estimation Tool", *IEEE Computer. Soc. Press [1992] Fifth International Workshop on Computer-Aided Software Engineering* - Montreal, Que., Canada..

[2] Chris. F Keremer, "An Agenda For Research In The Managerial Evaluation Of Computer Aided Software Reliability Estimation Tool", *IEEE Computer. Soc. Press [1989] Twenty-Second Annual Hawaii International Conference on System Sciences,* Volume II- Software Tr.

[3] HE Yan, "NHPP Software Reliability Growth Model Incorporating Fault Detection & Debugging", *IEEE 2013 IEEE 4th International Conference on Software Engineering and Service Science (ICSESS)* - Beijing, China (2013.05.23).

[4] Joydip Dhar, Anamika, Seema Ingle and Yaminee Sheshker, "Software Reliability Growth Model with Logistic-Exponential TEF in Imperfect Debugging Environment", *IEEE International Conference on Recent Advances and Innovations in Engineering, (*ICRAIE-2014), May 09-11, 2014, Jaipur, India.

[5] Yan Luo, Torsten Bergander and A.Ben Hamza., "Software Reliability Growth Modelling using a weighted Laplace Test Statistics", *IEEE 31st Annual International Computer Software and Applications Conference* - Vol. 2 - (COMPSAC 2007) - Beijing, China.

[6] Jiantao Pan, (1999 Spring). *Software Reliability*[Online]. Available:http://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/

[7] B.H.Far, *Software Reliability & Quality*[Online]. Available:http://people.ucalgary.ca/~far/Lectures/SENG521/PDF/SENG521-12_handout.pdf

[8] NASA KSC and Tim.C.Adams, *THE LAPLACE TEST*[Online].Available:http://kscddms.ksc.nasa.gov/Reliability/Documents/Laplace_Test.pdf

[9] Robert N.Charette (2005 September). *Why Software Fails*[Online].Available:http://spectrum.ieee.org/computing/software/why software-fails

[10] (2009 September 19). *10 historical software bugs with extreme consequences*[Online]. Available:http://royal.pingdom.com/2009/03/19/10-historical-software-bugs-with-extreme-consequences/

[11] *software horror stories*[Online]. Available:http://www.cs.tau.ac.il/~nachumd/horror.html

[12] Edwared E. Ogheneovo, "Software Dysfunction: Why Do Software Fail?", *Journal of Computer and Communications,* Vol.-2,pg. 25-35, April 2014.

[13] Dr.Linda Rosenberg, Ted Hammer and Jack Shaw, "SOFTWARE METRICS AND RELIABILITY "

[14] Vinay Tiwari and Dr. R.K.Pandey, "Open Source Software and Reliability Metrics," *International Journal Of Advanced Research in Computer and Communication Engineering,*Vol.1,Issue 10,December 2012.

[15] Gurpreet Kaur and Kailash Bahl, "Software Reliability, Metrics and Reliability Improvements Using Agile Process," *IJISET – International Journal Of Innovative Science, Engineering and Technology,* Vol.1,Issue 3,May 2014.

[16] Pankaj Dalal and Dr. D.S.Rao, "Software Reliability With Random Test Cases," *International Journal Of Advanced Research In Computer Science and Software Engineering*, Vol.4, Issue 11, November 2014.

[17] CSE IIT Kharagpur. *Software Reliability and Quality Management*[Online].Available:http://www.nptel.ac.in/courses/Webcoursecontents/IIT%20Kharagpur/Soft%20Engg/pdf/m13L32.pdf

[18] *System Specification* [Online].Available:http://www.cs.umd.edu/~mvz/cmsc435-s09/pdf/slides6.pdf

[19] Allen P. Nikora (2000 March 23). *Computer-Aided Software Reliability Estimation Tool (CASRE) user's guide,* version 3.0 [Online]. Available:http://csis.pace.edu/~ogotel/teaching/CS777CASREUserGuide.pdf

[20] *Computer-Aided Software Reliability Estimation* [Online]. Available:http://www.ece.uvic.ca/~shsaad/seng426/resources/Lab%20Slides/Lab7-SENG09.pdf

[21] Roger. R. Pressman, "Software Quality Assurance," in Software Engineering:, A Practitioner's Approach 5th ed. New York, USA: McGraw-Hill , pp. 212–213.

[22] *Open Channel Foundation CASRE 3.0* [Online]. Available:https://www.openchannelsoftware.com/projects/CASRE_3.0

[23] Prajoy Podder, Tanvir Zamar Khan, Mamdudul Haque Khan and M.Muktadir Rahman, "Comparison Performance Analysis of Hamming, Hanning and Blackman Window", *International Journal of Computer Applications(0975-8887)*, Volume 96-No.18, June 2014.

[24] (2015 May 04). *Understanding FFTs and Windowing* [Online]Available:http://www.ni.com/white-paper/4844/en/

[25] Tian Wenbo, Yu Jamming, Ma Xiaojin and Li Ji, "Power System Harmonic Detection Based on Bartlett-Hann Windowed FFT Interpolation", *Power and Energy Engineering Conference (APPEEC)*,2012 Asia-Pacific.