Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Use of Software Reliability Growth model to Estimate the Reliability of Web Applications

Jyoti Tamak
Department of Computer Science and Engineering
University Institute of Engineering & Technology
Kurukshetra University,Kurukshetra,
Haryana, India

Abstract-The reliability of web applications is more difficult to measure and improve because the large system has highly distributed nature. Hardware faults can be easily predicted rather than the software faults. The customer required more security and accuracy in web applications. So the reliability of web applications is important to consider over different kind of network. The reliability of web applications is more complex rather than the other software systems. In this paper, we will use the Goel-Okumoto Software Reliability Growth models (SRGM) to detect the number of faults in a specified time and estimate its reliability in regard of web applications. The rate of change is calculated by executing the test cases for actual defects per day. The Goel-Okumoto uses exponential distribution to predict the number of faults in web applications. We will assess the software reliability of web applications by using software reliability growth models.

Keywords:- Fault detection, software reliability growth model (SRGM), non-homogeneous Poisson process(NHPP),residual defects, confidence interval .

I. Introduction to Software Reliability Growth Models

Software reliability is an important component in critical business applications. Developing reliable software is very difficult because there is interdependence among all the software modules as of existing software. This is also very difficult to find out whether the software being delivered is reliable or not. The users or customer feedback i.e. problem reports, system outages, complaints or compliments indicate the reliability of any software product. There are two types of software reliability models which help to predict software reliability by executing test cases. The first types of models are called "defect density" models. These models use loop, lines of code, input or output and external references to find out the number of faults in the software product. The second types of models are called "software reliability growth models”. These models used to correlate defect detection data statistically with known functions such as an exponential function. If the correlation is good, the known function can be used to forecast future behavior [1]. Software reliability is defined as the probability of failure free software operation for a specified period of time in a specified environment. Software reliability is accepted as the major factor rather then the other factors like as usability, functionality and maintainability etc. When we discuss about web based software reliability, we have to consider many technologies. A collection of servlets, html pages, classes and other resources are used and run on different platforms by different users. Each component has its own failure modes and sources of delay and unreliability. The failure of any one system or service in the path between server and user will in effect cause failure of the entire application as far as the user is concerned. The important characteristics of a website are high reliability, high availability, high security and more interactivity. All these characters are independent on the web application [3]. Reliability growth models are based upon the assumption that the reliability of a program is based on the number of faults. Statistical techniques are used by such models to observe the failures during software testing and operation to measure the product’s reliability. If we have N installations of the software, and a total of F failures are reportable at time T, the failure rate of the software can be computed as $\lambda = F / (N*T)$.

When the product is released, there are more chances to find out the number of defects in the software. The different users use the software due to which it becomes easier to calculate the reliability of software. In the most software reliability growth models a parameter is used to relate the total number of faults contained in a set of code. If this parameter and the current number of faults discovered, we know how many faults remain in the code (see Figure 1).Knowing the number of residual defects helps us decide whether or not the software is ready to release and how much more testing is required if we decide the software is not ready to release. It gives us an approximation of the number of failures that our customers will come across during the operation of software. This approximation helps us to plan the suitable levels of support that will be required for fault correction and also verify the cost of underneath software.

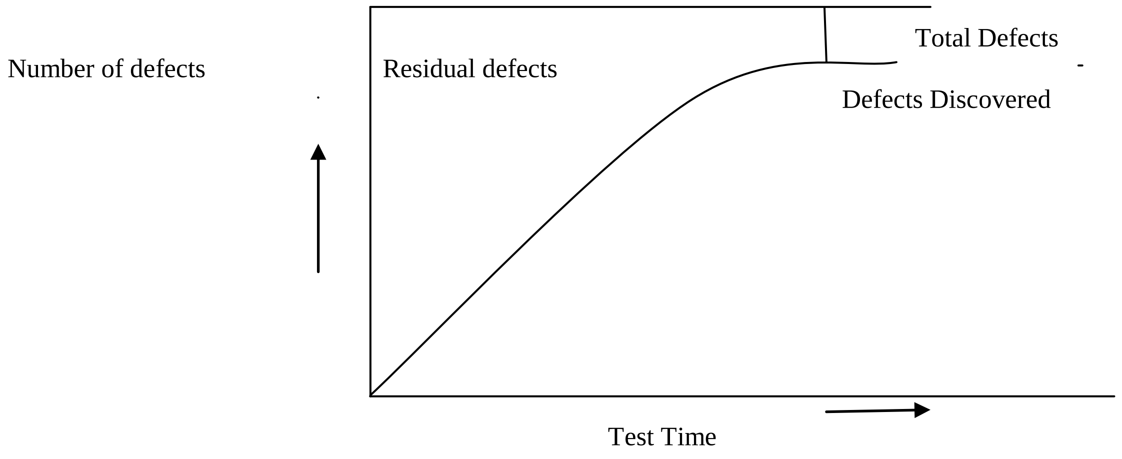


Fig 1. Residual Defects

Types of Reliability Growth Models

Software reliability growth models have two types of models: concave and S-shaped. Both models have the same asymptotic behavior. In both models, the rate of change is decreases as the number of faults detection increases. Both models are shown in Figure. It is assumed that the rate of change is proportional to the number of faults in the software. Each time a fault is detected or repaired, the fault detection increases and the rate of change decreases. The concave model firmly follows this prototype. The early testing is not as efficient as later testing assumed in the S-shaped model, so there is a ramp-up period during which the defect detection rate increases. Let us suppose if the early Quality Assurance test uncover the faults in other products that prevent QA from finding defects in the product being tested [1]

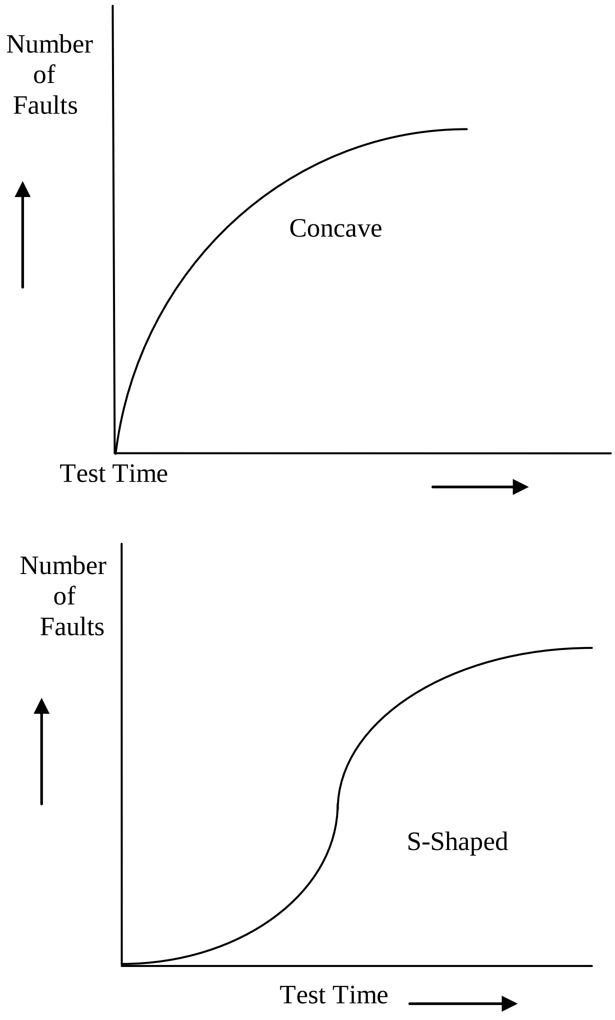
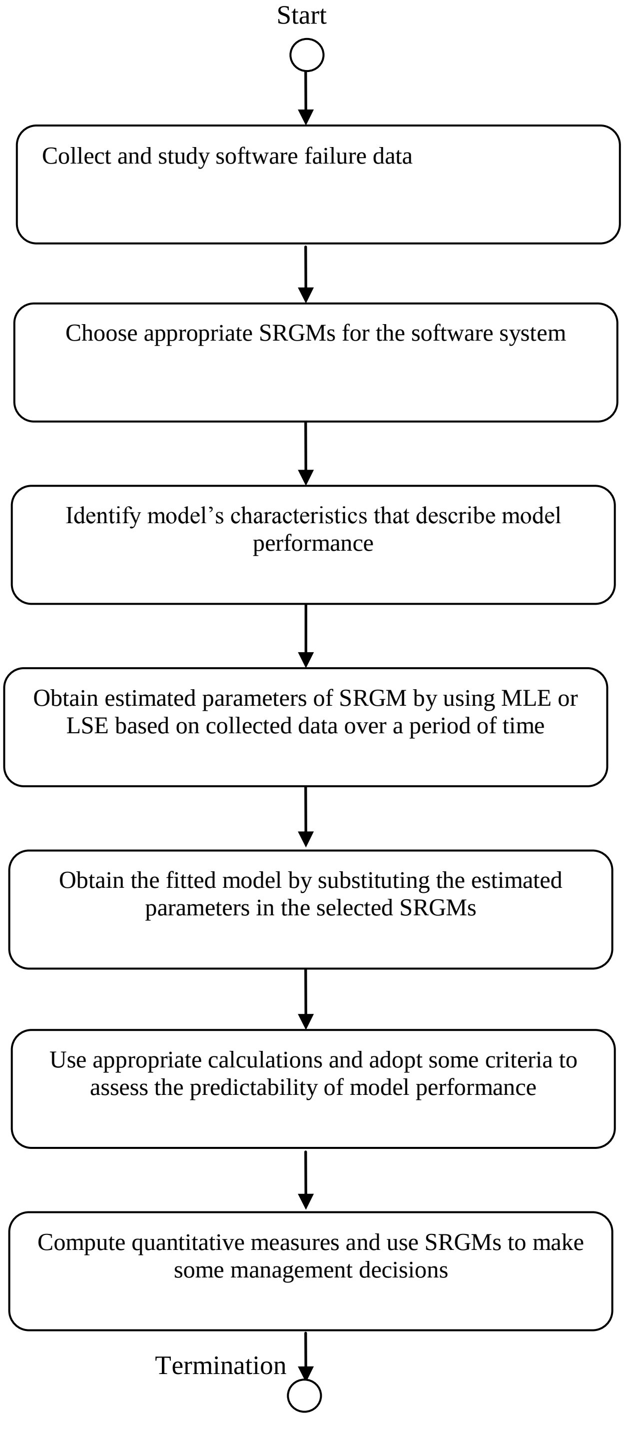


Fig 2: Concave and S-Shaped Models

II. Procedure of Using SRGM



III. Methodology

In various Software Reliability Growth Models, two important factors affect the reliability. The first factor is the number of actual defects and the second is the rate of change. These both are used to represent the reliability of software or web applications. After the detection of faults it becomes easier to decide whether the software is suitable for customer to use or not and how much more testing resources are required. It becomes easier to estimate the number of defects that will be encountered in the future. The rate of change is used to measure the effectiveness of fault detection by executing the test cases. The rate of change is assumed constant in maximum cases. The most researchers think that all defects have equal probability of being detected during the testing process due to which the rate of change is constant. But in realty, the rate of change is based on the program size, skill of test teams and software testability. The rate of change can be increase, decrease or remain constant [4].

Software Reliability Growth Model (SRGM) is used to correlate defect detection data with estimated residual defects and time.

In the analysis, test cases are executed to find out the actual defects per day till date. The actual defects are plotted against time. The curve should be flattering out as time progresses. The curve shows the growth of reliability. If the curve is not flattering out or in zigzag shape, it means the software is not reliable. In this paper we use reliability model based on Goel-Okumoto and Confidence interval (CI). This reliability model is based on parameters such as defects; Amount of Testing (in days) and Statistical Techniques (G-O and CI).The amount of testing can be measured in calendar time as well as number of tests run. In this example we will use calendar time and number of days as a parameter [2].

The following steps are used to detect the faults:-

Step 1:- Collect data as per table shown in Software Reliability Growth Model. Compute the actual defects and rate of change accurately. Find out the values of total actual defects (a), test case efficiency or rate of fault detection (b) and current time (t).

TABLE 1

Days	**Actual faults**	**Test executed**	**Rate of change**
1	140	200	.70
2	125	200	.62
3	108	175	.61
4	102	175	.58
5	85	150	.56
6	80	150	.53
7	65	125	.52
8	60	125	.48
9	46	100	.46
10	40	100	.40
11	28	75	.37
12	25	75	.33
	904	1650	.51

Total number of defects (a) = 904

Rate of change or test case efficiency = 0.51

Step 2:- Compute the faults by using Goel-Okumoto model.

$$\mu(t) = a(1 - e^{-bt}) \quad , a>0, b>0, \text{ where}$$

$\mu(t) =$ current actual defects, using Goel-Okumoto Model.

a = initially a is taken as total defect detection till date

b = the rate at which the defect-rate decreases or the defects are detected

Goel & Okumoto is one of the most popular NHPP models in the field of software reliability engineering. Goel and Okumoto assume the mean value and intensity functions as Exponential SRM:

Mean Value Function (MVF):-

$$\mu(t) = a(1 - e^{-bt})$$

$$\mu(t) = 902$$

Failure Intensity Function:-

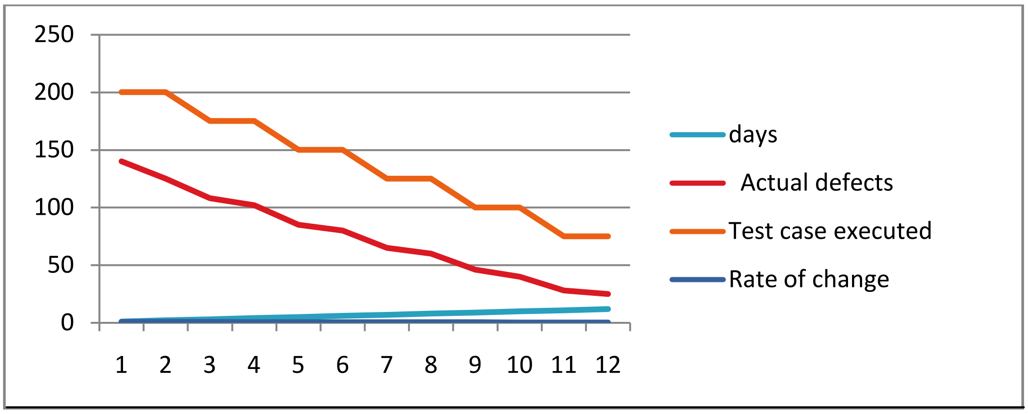
$$\lambda(t) = \mu'(t) = ab\, e^{-bt}$$

$$\lambda(t) = 0.92$$

Failure intensity function is used to estimate the quality of software.

Graph:-

The table values are expressed in the form of graph. The horizontal line is used to represent the time and vertical line represents the number of defects. The blue line represents the number of days and purple line represents the rate of change. Similarly, red line represents the actual defects per day while the green line represents the execution of test cases in the graph.



Step 3:- Compute Confidence interval (CI), to get estimated defects with reasonable interval

Confidence Interval Table:-

TABLE 2

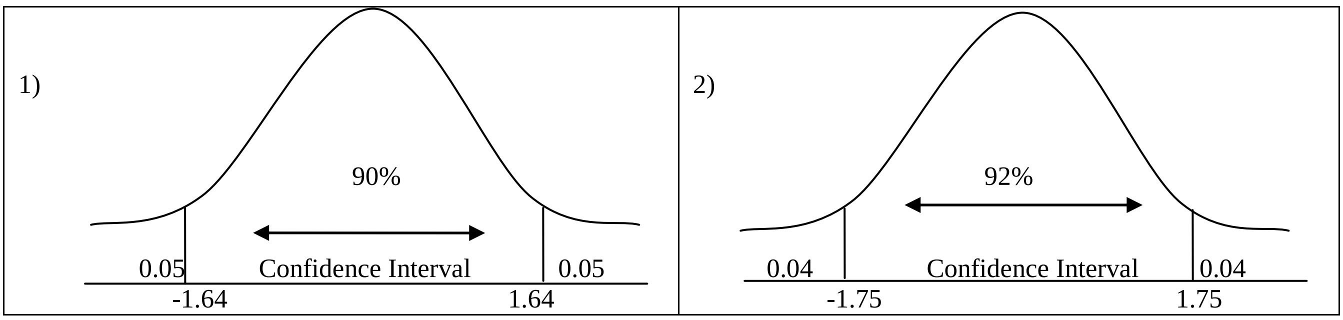
α	**1-α**	**Zα/2**
0.10	90	1.64
0.09	91	1.70
0.08	92	1.75
0.07	93	1.81
0.06	94	1.88
0.05	95	1.96
0.04	96	2.05
0.03	97	2.17
0.02	98	2.33
0.01	99	2.57

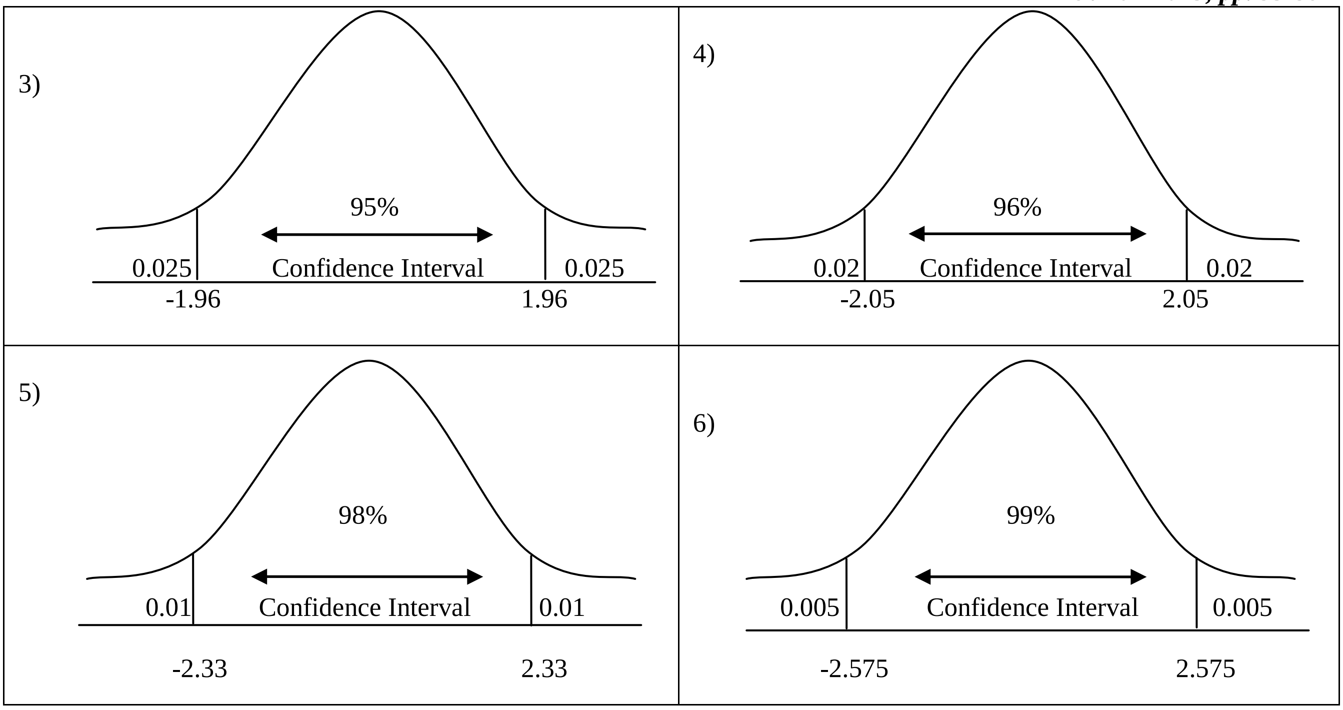
The confidence interval is estimated by using normal distribution with standard deviation. Where, α represents total area and α/2 represents half of the normal distribution. The confidence interval for 95% is equal to 1.96, represented by Zα/2. Confidence Interval (CI) gives probable range of values from the set of sample data. The confidence level is the possibility coupled with a confidence interval, which is expressed as percentage (%).The confidence level tells you how sure you can be. The confidence interval is symmetrically located around the normally distributed estimation. The larger the confidence level is, the narrower the confidence interval [5].Wider confidence interval, gives more possibility to estimate the residual defects prediction in the right range. Hence 95% confidence level is the most frequent and reasonable value than 99% or 90%.

Steps to find out the confidence interval [6]:-

1. α : Subtract the given CI from 1.
   α = 1- 0.90 = .10
2. $\alpha/2$: Divide the given CI by 2
   $\alpha/2$= 0.10/2 = 0.05
3. Then look up that area in the Z-table.
   0.90+0.05 = 0.95
4. The closest Z-value to an area of .05 is 1.645. Similarly, all values are calculated from 91 to 99%.

Confidence interval curves





Let us use 95% confidence level to predict faults specified by:-

$$CI = \mu(t) + Z\alpha/2 * SQRT(\mu(t))$$

Predict defects at (t) day's formula:-

$$CIres = \mu(t) + Z\alpha/2 * SQRT(\mu(t))$$

Predict defects at 13th day:-

$$CIres = 902+1.96*30.03 = 961$$

Reliability:- Reliability per day

$$\lambda = .92$$
$$t = 1 \text{ day}$$
$$R(t) = e^{-\lambda t} = 0.40$$

Here we calculate the reliability of one day. From the result it has proven, as the number of faults increases per day the reliability will be decreases.

IV. Conclusion & Future Work

There are number of software reliability growth models used to detect faults in web applications. But every model has some benefits as well as limitations also. In this paper, we use Goel-Okumoto model to predict the defects by executing the number of test cases per day for a specified time. The random data is collected to calculate the mean value function and failure intensity function by using exponential distribution. After the calculation of MVF and failure intensity function, the reliability is calculated. The Goel-Okumoto model is simple and easier to use in comparison of other reliability growth models. This model provides good results in case of fault detection.

The future work involves the use of other reliability growth models to predict the reliability of web applications and to find out the best one model which provides the efficient results in case of fault detection and reliability.

Acknowledgment

I would like to express my special thanks and gratitude to my guide Mr. Karambir who gave me the golden opportunity to write this paper on the topic Use of Software Reliability Growth model to Estimate the Reliability of Web Applications which also helped me in carrying out an intensive research through which I came across so many new concepts and terminologies. I am really thankful to him.

References

[1] Alan Wood "Software Reliability Growth Models" in 1996

[2] Sunil Savanur **"**Use Software Reliability Growth Model (SRGM) for Residual Defects Estimation" in January 1, 2013 .

[3] D.Swamydoss, Dr. Kadhar Nawaz "Enhanced Version of Growth Model in Web Based Software Reliability Engineering" in JGRCS Volume 2, No. 12, December 2011

[4] Sy-Yen Kuo, Chin-Yu Huang, and Michael R. Lyu "Framework for Modeling Software Reliability, Using Various Testing-Efforts and Fault-Detection Rates" in IEEE Transactions on Reliability, Vol. 50, NO. 3, Sept 2001

[5] Frankfort-Nachmias, C. & Leon-Guerrero,(2006).

[6] Stephanie "How to find a confidence interval" in Oct 2009.

[7] Chin-Yu Huang, *Member, IEEE*, and Chu-Ti Lin "Software Reliability Analysis by Considering Fault Dependency and Debugging Time Lag" in IEEE Transactions on Reliability, Vol. 55, No. 3, September 2006.

[8] Y. P. Wu, Q. P.Hu, M. Xie**,** S. H. Ng "Modeling and Analysis of Software Fault Detection and Correction Process by Considering Time Dependency" in IEEE Transactions on Reliability, Vol. 56, No. 4, Dec 2007.

[9] Chin-Yu Huang, and Wei-Chin Huang "Software Reliability Analysis and Measurement Using Finite and Infinite Server Queuing Models" IEEE Transactions on Reliability, Vol. 57, No. 1, March 2008.

[10] Chin-Yu Huang, and Michael R. Lyu "Estimation and Analysis of Some Generalized Multiple Change-Point Software Reliability Models" IEEE Transactions on Reliability, Vol. 60, No. 2, June 2011

