© 2017 IJEDR | Volume 5, Issue 3 | ISSN: 2321-9939

# Software Reliability growth Models, their assumptions, reality and usage of Two stage model for predicting software reliability

[1] Dr R.K.Sharma,[2] Dr. Archana Kumar,[3] Sapna Bajaj
[1] Professor,[2] Director,[3] Asst. Professor
[1] CSE Agra College of Engg.and Technology, Agra, India
[2] CSE DITM, Gannaur, India
[3] CSE. B.M.Institute of Engg. and Technology, Sonipat, India

---

**Abstract— In this paper, we will discuss about types of software reliability growth models, and some underlying assumptions that has been used while implementing these SRGMs and their reality in practical life. Here, we are also discussing that how a software reliability model can be more useful so that it can be able to find more and more number of errors and then we have defined some equations which has been used to check whether the available reliability Growth model is useful or not.**

**IndexTerms— Exponential function, Failure Data, Software Reliability, Software reliability growth models, Translation Time**

---

## I. INTRO DUCTION

For critical or important business applications, continuous availability is a requirement. and software reliability is most important component of continuous availability of any application. However, rare kinds of single software defects can cause a system failure .To avoid these failures and to decrease software support costs, software designers needs to deliver reliable software. Development of reliable software is a major problem which software industry is facing since its development. Limitation of resources, infeasible requirements of customer and pressure of delivering software on time are the major factors that decreases the reliability of the software. Developing reliable software is also a tough task when the modules of the software are much dependent on each other. It may also be possible that during delivery of software the software is reliable but after delivery, during the usage of the product by customer ,the software becomes unreliable due to the errors or faults occurred because of using the product in the real time user's environment. Generally software vendors try to remove all the faults before delivery so that it does not affect the product's reliability after the shipping of the product. So for this purpose software reliability growth models(SRGM) has been used. These models are used to predict or estimate the reliability of software at an early stage.
There exist two types of reliability models:

1. First type of reliability models are those which predict the reliability of the product by considering the design and constraint parameters of the software. These are known as defect density models. The parameters on which these models focus are LOC, Nesting in Loops, complexity, input domain , output domain etc.
2. 2. Another type of Reliability models predict the reliability by considering the available test data. These types of models are known as software reliability growth models. These types of models tries to apply the calculated failure data with some available functions eg. Exponential function. If that data is well suited to that specific function, then that function has been further used for predicting the reliability.

There are many model parameters and their available options.
These parameters are :

1.Amount of testing- It is measured by calculating CPU time or By Calculating Total Number of Test Cases.

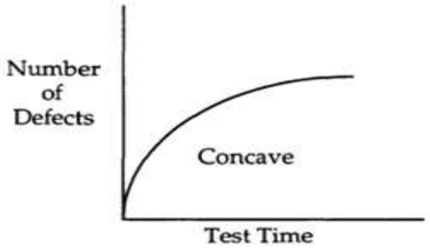
2. Total Defect Data- Calculated by total number of available defects in the software at that time. There are two types of data for calculating the reliability in SRGMs. The first type of data is defined by the time at which defects were occurred in the program and it is known as Test Time Data whereas second type of data is the amount of defects that occurred in the program which is also known as Defect Data. There are some parameters used for measuring test time data. these parameters are: Total Calendar time, Total number of running Test Cases and Total Execution Time of the program.
For calculating Exact Test Time data. Total Number of test cases should be maximum, as more will be the number of test cases; more will be the probability of finding defects.

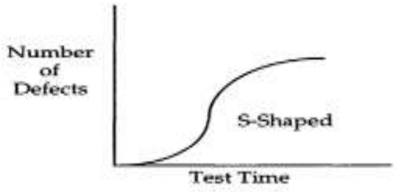
## II. TYPES OF RELIABILITY GROWTH MODELS

SRGMs are grouped into two types of classes:

1. Concave model
2. S- Shaped Model

These types of models have same kind of asymptotic behavior. Here are the designs of Concave and S- Shaped Model:




Concave Models has concave function as they bend downwards continuously while s-Shaped models has s shaped function and they are first convex and then becomes concave.

## III. UNDERLYING ASSUMPTIONS OF AVAILABLE SRGM AND THEIR REALITY

The main problem of software reliability growth models is that they make some assumptions . These assumptions are about test repair and testing. But many of these assumptions are not practically feasible. So here are some of those assumptions and their reality:

**Assumption1**- When defects are discovered they are immediately resolved.
**Reality**: It is not possible to repair all the defects immediately. We can partially remove defects. if an un repaired defects prevents other defects from being found then test time may be changed.

**Assumption2**- When we repair the defect, the software becomes perfect.
**Reality**: When we repair the defects, there are the chances that these defects introduce more new defects so its not always possible that the software becomes perfect after repairing the defect.

**Assumption3**-Only the members of test group can report the defects.
**Reality**: Apart from test group, managers and even end users can also report the defects. If the testing activity can be done parallel by multiple groups then defects can be reported by multiple groups too.

**Assumption 4**- Different unit of times like total number of test cases, total execution time, total development time etc. are considered to be equivalent
.**Reality**: All units of time are not equivalent actually. Total number of test cases depends upon the stress we want to give on the software to find more and more errors . Numbers of test cases can be more or less depending upon the number of errors we want to reveal. Total execution time depends upon the number of errors in the program. More will be the number of errors, more will be the execution time.

**Assumption5**- Failures encountered in the program are independent to each other.
**Reality:** Failures are not Independent because one error can lead to multiple errors which causes the failures of the software.

## IV. NECESSARY CONDITIONS FOR A MODEL TO BE USEFUL AND ACCURATE

By looking at the above assumptions, we can define the necessary conditions for a model to become suitable to all software.
Those necessary conditions are as follows:

1. If a software or product is in testing phase then the model should remain stable during that period and there should be no change in the software during that duration.
2. The model should provide almost exact prediction of the total number of errors that arise in the software during testing phase.

## V. PROBLEM FORMULATION

According to our first condition which states that the model should remain stable during the software testing phase, so if a bulk amount of new code has been added to the software during the testing phase , there is a method that permits to change the data according to the change in the code after addition.

Here the overall curve is a two stage model which is a combination of S-Shaped and Concave –Shaped Model.This Model uses the advantages of both the models.

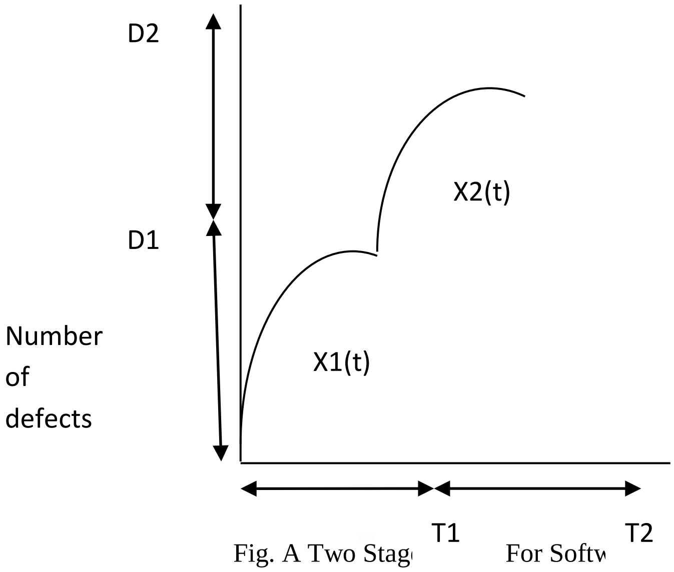



Fig. A Two Stag... For Softw... iability

In this figure D1 are the defects found in T1 time before the addition of new code and (D2-D1) are the defects which are found in (T2-T1) time after the addition of the code. So the main objective is to convert this data to a model X(t) that have been obtained if the new code has been added to the software initially, ie. before the start of testing phase .Suppose $X_1(t)$ represents the defect data before adding the new code and $X_2(t)$ represents the defect data after the addition of the code. Here the model X(t) has been created by changing the failure time from $X_1(t)$ and $X_2(t)$.

We are showing that $X_1(t)$ has been applied to the time period 0-T1 and the model $X_2(t)$ has been applied during the time period (T1-T2).

Here are the steps of the translation:

**Step1:** To find the parameters of $X_1(t)$ for obtaining $X_1(t)=a_1(1-e^{-b1t})$

Here, a1=Expected number of defects present in the code at time 0-T1

b= the rate at which the rate of failure decrease, it can be also called as shape factor.

**Step2:**To find the parameters of $X_2(t)$ for obtaining $X_2(t)= a_2(1-e^{-b2t})$

Now a2= Expected number of defects present in the code during the time (T1-T2).

**Step3**:Now, Calculate the total translation time for the errors which were found in the code before adding the new code. This translation time $Z_1$ can be calculated by using the following equation:

$Z_i= (-1/b2)\ln(1-(a1/a2)(1-e^{-b1ti})$

**Step4**: Calculate the translation time $Z_2$ for the errors which were found in the code after adding the new code. Here, the value of $Z_2$ is:

$Z_2=(-1/b2)\ln(1-D1/a2)$

Where, $D1=a2(1-e^{-b2Z_2})$

**Step5**: Compare the values of $Z_2$ and $Z_1$ and check whether the value of $Z_2$ is less than the value of $Z_1$ Because according to our problem statement , the failure rate after insertion of code should be less than the failure rate before the insertion of the code. If it is less, then our results are correct.

## VI. Conclusion

Based upon the discussion of the above reliability growth models and various assumptions it has been concluded that the two stage model which we have proposed here is better than concave and s shaped model in the software's where we want to insert more data during testing phase as it leads to less number of defects in that situation if compared with the other two models. However, future research needs to be extending the above proposed model or to find some new environment which is more realistic and perfect for representing the software reliability.

## VII. REFERENCES

[1]B.Littlewood,The Littlewood- Verrall Model for Software Reliability Compared With Some Rivals, pp. 251-257.

[2]Najeeb Ullah,Maurizio Morisio,Antonio Vetro,A Comparative Analysis Of Software Reliability Growth Models Using Defect Data of Closed And Open Source Software,2012 IEEE 35th Software Engineering Workshop,pp.187-191.

[3]Rolph Ernst,Co- Design of Embedded System- Status and Trends,1998,IEEE Transactions on Design and Test of Computers,pp.45-52.

[4]Anna Hac,Using a Software Reliability Model To Design A Tele- Communication Software Architecture,IEEE Transactions on Reliability,Vol.40,No.4,Oct 1991,pp.488-93.

[5]H.Joe,N.Reid,On The Software Reliability Models Of Jelinski-Moranda And Littlewood, IEEE Transaction on Reliability,Vol R.34,No.03,1985 August,pp.216-18.

[6]Miroslaw Staron, Rakesh Rana,,Selecting Software Reliability Growth Models And Improving Their Predictive Accuracy Using Hostorical Project Data,ELSEVIER The Journal Of System And Software2014,pp.59-78.