IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 5, May 2016.
www.ijiset.com
ISSN 2319 – 5967

Software Reliability Measurement Using Software Reliability Growth Model.

Sonia Rikhi
(PG student in cse department at PIET
soniarikhi1988@gmail.com

Kailash Bahl
Kailash.bahl@gmail.com

Abstract— Reliability is the important aspect of software. Software Quality is important factor when we design a software..The software quality depends on different factors such as software reliability, efficiency, cost etc. Software Reliability determines the probability of failure-free software operation for a specified period of time in a particular environment. Software Reliability can be categorized into 3 parts: modeling, measurement & improvement. In this paper, we have described the software reliability as the measure of software quality. Reliable software is able to accomplish the better achievement and allows software to work properly in a specified environment. [1, 2]

Keywords: Software reliability, Software reliability growth model, MTTR, MTTF, MTBF.

INTRODUCTION

Reliability of Software is the possibility that software will work properly in a described environment and for a described time. Using the following formula, the probability of failure is computed by testing a sample of all input which is available.

Probability = Number of failing cases / Total number of cases under consideration [3,4]

Software:

Software is a set of instructions and code written by programmers in any of separate special computer languages. Software is divided into two main groups:

(1) System software: controls the basic operation of a computer which is already exist in the machine. Like BIOS and Operating System.

(2) Application software: It perform the particular tasks as per the user need, such as accounting, communicating, data processing, word processing.[5,6]

.

Software reliability:

It is amount of time that the software is available for use as dedicated sub attributes such as maturity, fault tolerance, recoverability etc. It is also determine as probability of failure free software task for particular period of time in specified environment. A simple degree of reliability is mean time between failure (MTBF).It is calculated as MTBF=MTTF+MTTR here, MTTF=mean time failure MTTR=mean time to repair. In addition to reliability measure, we must grow up the degree of availability. It is possibility that a program is operating according to requirements at a given point in time and is defined as, Availability =(MTTF /(MTTF+MTTR))*100%.

ISSN 2319 – 5967

Software reliability is divided into three activities:
1. Error prevention
2. Fault detection and removal.
3. Measurements to maximize reliability, [7, 8]

Introduction to Software Reliability Growth Models:

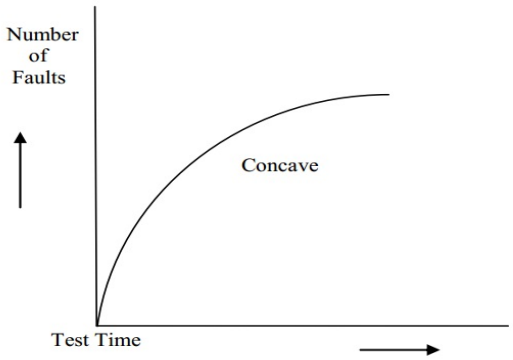
In critical business applications, the vital component is software reliability .In software reliability ,growing is very difficult because there is interrelationship among all the software modules as of existing software. This is also very difficult to find out whether the software is being expressed reliable or not. The users or customer feedback i.e. problem reports, complaints or compliments prove the reliability of any software product.

Software reliability models are classification into two types, which help to predict software reliability by executing test cases. The first group of models is called defect density models. These type of models use loop, lines of code, input or output and external references to find out the number of faults in the software . The second groups of models are called “software reliability growth models”.

The two categories of Software reliability growth models are: concave and S-shaped. In both models the rate of change is decreases as the number of error detection increases.

The fault is detected or repaired, the fault detection increases and the rate of change decreases.

1) Concave



2)S-Shaped

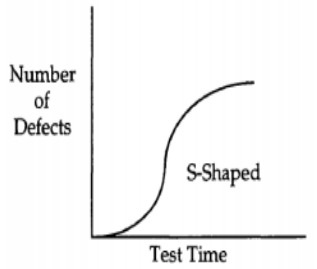


Figure 2. S-shaped model[4]

According to type of failure process software reliability growth models are categorized as follows:

I. Times between failure models:

II. Failure count model:

III. Fault seeding model:

IV. Input domain based model

I selected two models.[9,10]

TBF Models: In these types of models; the process under study is the time between

failures. It is assumed that the time between (i-1) th and ith defeat is a variable, following a distribution whose parameters calculate on the number of defect remaining in the program during this interval. Guess of these parameters are accessed from the checked values of TBFs and then the parameters of SWR are access from the fitted models.

FC Models: In Models, the random variable of interest is the number of errors appears during particular time intervals. It is made-up that failure counts follow a accepted stochastic process. The time can be calendar time or CPU time Parameters of the failure rate can be computed from the found values of failure counts and then the SWR parameters are taken from the appropriate expression. [11, 12]

CONCLUSION

Software reliability is a vital research space and software reliability is fundamental part of software quality .There are many models use in software reliability but I choose two models so I want to compare between in these two models, Times between failure models and Failure count model for check the reliability .[13,14]

References:

(1)http://www.ijiset.com/v1s3/IJISET_V1_I3_24.pdf

2)http://oai.cwi.nl/oai/asset/18249/18249B.pdf

(3)https://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/

(4)https://en.wikipedia.org/wiki/Reliability_engineering

(5)http://www.businessdictionary.com/definition/software.html

(6)https://en.wikipedia.org/wiki/Software

(8)http://ahvaz.ist.unomaha.edu/srg/sofware-reliability-quantitative-techrep.pdf

(9)http://www.ijcaonline.org/volume10/number5/pxc3871990.pdf

(10)http://www.ece.uvic.ca/~itraore/seng426-07/notes/qual07-8.pdf

(11)https://www.cs.drexel.edu/~jhk39/teaching/cs576su06/week1Readings/goel.pdf

(12)http://www.ijarcsse.com/docs/papers/10_October2012/Volume_2_issue_10_October2012/V2I10-0043.pdf

(13)http://www.ijiset.com/v1s3/IJISET_V1_I3_24.pdf

(14)http://cisjournal.org/journalofcomputing/archive/vol3no9/vol3no9_7.pdf