International Journal of Latest Trends in Engineering and Technology
Vol.(10)Issue(1), pp.305-308
DOI: http://dx.doi.org/10.21172/1.101.54
e-ISSN:2278-621X

ANALYSIS OF SOFTWARE RELIABILITY ESTIMATION MODELS

Anu[1]

Abstract-There is no doubt that the reliability ensures the quality, availability and simplicity. There is complex network of open source software and Developing reliable software in today's environment exhibits coordination, control and communication challenge. Software reliability represents the main concern of software quality, and the estimation of this is not easy one to achieve with accuracy. The unavoidable price of reliability is simplicity;itis defined as the probability of failure free process of computer programfor a describedamount of time in a specific domain. In order to achieve the good quality, reliability estimation is most important key factor for this. We concentrate in this paper is to study the existing model for software reliability estimation.

Keyword: Reliability, Usability and Maintainability etc.

1. INTRODUCTION

Lesser the failure rate higher the reliability, software reliability means probability of the uninterrupted operation of software. It depends upon flexibility, complexity, maintainability, reusability. If software is factual and free of error then its candid is 1 or 0.Software veracious is directly proportional to software reiteration.Reusability may be stated as, how to use any element in numerous times outwardly any breakdown or a bitfarther bounds is called software reusability.

Software fidelity= {1- probability of failure}

Software reliability problems can almost always be traced to defects in software programs.

Software reliability depends simultaneously on distinct property. These properties are indexed below

1. Usability
2. Motility
3. Maintainability Aesthetic
4. Reusability

A simple measure of reliability is to measure the mean time between failure (MTBF) and mean time to repair (MTTR).

MTBF= MTTF + MTTR

An alternative measure to dependability is failure in time (FIT) – a numericalcompute of how countless failures a module will have over one million hours of action. Therefore, 1 FIT is correspondent to one failure in each million hours of action.

In adding together to reliability calculate, build up a compute of software accessibility. Software accessibility is the prospect that a process is working according necessities at a given position in time.

$$\text{Availability} = \frac{\text{MTTF}}{\text{MTTF +MTTR}} \times 100$$

Software reliability = {1 – Probability of accessibility}

As software growth has turn intoannecessaryventure for a lot of organizations, software evaluation is ahead an mountingsignificance in efficient software schemesupervision. In follow, software evaluationincludes cost inference, eminencejudgment, risk study, etc. Accurate software inference can supplydominantbacking for software supervision decisions .The foremost challenges are

1) The associationsamong software harvest metrics and causative factors demonstratesturdyintricate nonlinear characteristics.
2) Measurements of software metrics are frequentlyindefinite and indecisive
3) Impenetrability in utilizing both proficient knowledge and statisticalventurefacts in one sculpt.

2. RELATED WORK:

Over the past few years, many open source software have been developed and this is still a continuing effort.As in [1] provides the criteria for software reliability estimation and their systematic study with reliability functions. As in [2] author studied the various reliability models and there theory to applying a particular model in the specified environment. As in [3, 4] author explains the software reliability estimation models, reliability matrices and agile development, usage in reliability estimation. As in [5, 6] author explain the object oriented metrics of reliability estimation and an algebra of reliability calculation. As in [7-11] authors gives an alternative method of fuzzy system to measure the reliability. V. Kumar and D.

[1] Assistant Professor (Resource Person), Department of Computer Science, UIET, MDU, Rohtak, Haryana, India

chhabra [12] designed the Fuzzy Logic Controller for Active Vibration Control of Cantilever Plate with Piezo - Patches as Sensor /Actuator. V. Kumar and A.Kumar[13] optimised the vibration of the plate with the help of Adaptive neuro-fuzzy inference system (ANFIS) controller.

Varun Kumar[14] control the vibration of the plate with the help of fuzzy logic controller. A finite element model of a two-dimensional cantilever plate instrumented with a piezoelectric patches sensor-actuator pair is derived.

Criteria For analysis of software reliability models

1. Capability 3. Quality
2. Simplicity 4. Precision Levels 5. Applicability

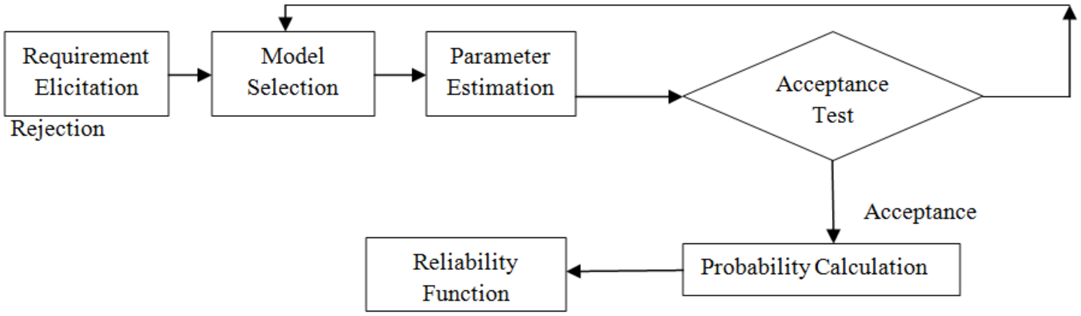

Fig. 1

3. EXISTING MODELS

Software reliability growth fashions are normally designed for use with facts accrued in phrases of the testing instant among disasters, along withitsdistant on such models to facilitate this monitorresolve cognizance. These fashions are on average said in terms of equations; the suggest price feature and the failure depth characteristic. All of the fashions tested have parameters. Regardless of ways the ones models in which to start with formulated, we are able to check with the parameters of these fashions as $\square$ zero and $\square$ 1.

4. EXPONENTIAL MODEL

The maximum extensively worn software dependability increase version is the exponential sculpt. This is a stochastic version base totally on a non-homogeneous poisonsprocedure. The authentic exponential version completeexercise of the onwards wall timepieceat the same time as a failure turned into encountered. A massive refinement modified in it and restated the mold in terms of CPU completingmoment making an allowance for greater accurate predictions. It defined a method for moving between execution time furthermore wall clock time, building it less difficult to craft predictions in phrases of actual global calendars and time limits. Although superior to the sooner models, it's been verified that the exponential version isn't typically the maximum correct SRGM. However, this version stays well-known and extensively used.

As formerly stated, this observe specializes in carrying out based totally models, measured in CPU seconds. For this case, the exponential version takes the shape

$$\square\ (t) = \quad \square_{0[}1\ ,e,^{\square\ 1t]} \tag{1.1}$$

$$\square\ (t) = \quad \square_{0}\square_{1}e,^{\square\ 1t} \tag{1.2}$$

Where$\square_0$ is in use to be the wholeamount of defects present in the software, and $\square_1$ is taken to be the per-fault risk rate of the course.

4.1 Estimation Parameter:

Here equation (1.2) the estimated failure intensity for the exponential mold at time t. We know how tolinarite this equation and fit the in fact failure intensity data to obtain estimates of the model parameters.

$\mathrm{Ln}\ (\square) = \ln\ (\square^{E}{}_{0}\square^{E}{}_{1}),\ \square^{E}{}_{1}t$

To achieveconstraint estimates we catch the innate log of the breakdown intensity observed at every time t and fit a line up to this data. We then obtain constraint estimates by solving for the parameters in terms of the slope and intercept. In this case, the slope is,$\square^{E}{}_{1}$, and the intercept is equal to ln ($\square^{E}{}_{0}\square^{E}{}_{1)}$.

4.2 Logarithmic Model

Similar to the exponential version, it models the breakdown system as a non-homogeneous poisonsapproach. The nearly all full-size distinction among this version and the exponential is so as to the logarithmic edition assumes that failure strength will diminish exponentially with the estimated variety of disasters skilled, even as the exponential model assumes a same cut price in failure depth with each fault exposed and corrected. In this experience it may be regarded as a non-prevent approach of the geometric model.

It is especially simple to apply, despite the fact that no longer as extensively used due to the fact the exponential version. This can be due in component to the problem of acquiring a concrete elucidation of the version's parameters. We present such an interpretation in financial disaster three.
The logarithmic model takes the outline as

$$\square\,(t) = \square_0 \ln(1 + \square_1 t) \tag{1.3}$$

$$\square\,(t) = \frac{\square\ 0\square\ 1}{1 + \square_1 t} \tag{1.4}$$

Farr gives the ceiling likelihood equations as

$$\square_0^L = \frac{n}{\mathrm{Ln}(1 + \square_1^L t_{n)}}$$

$$\frac{1}{\square_1^L}\overset{n}{\square_{i=1}}\frac{1}{1 + \square_1^L t_i} = \frac{nt_n}{(1 + \square_1^l t_n)\ln(1 + \square_1^L t_{n)}}$$

Once more, $\square_0$ is reliant on $\square_1$. The dilemma is to locate an exactestimation for $\square^L{}_1$. We receive the left passface of equation 2.4 and deduct the right hand side. If we have preferred$\square^L{}_1$suitably this equation should equivalent to zero. We make use of the golden-hairedsegmentschedule to find the solution to this equation as described previously.
Short Term Predictive Capability Model: SRE
In this wewillpowerscale the fast term prognosticprecision of software program reliability increase models through looking how close they come to predicting the next found failure. We will time period our mistakes degree Short time period Relative Error, or SRE. To compute this we healthy the version the use of only the first records factors, after which use the prepared parameters to predict what number of mistakes may want to have occurred at the time at the same time as the zero.33 mistakes become clearly located (the 0.33 information point). We take virtually the fee of the difference amongst these values, and divide by way of the variety of errors clearly observed right now. We then repeat the way the use of the primary 3 records points, then the number one four, and so on till we have used the primary n ,1 records points in the set to are expecting what number of errors are expected at time t_n. We average the results over all predictions made.

$$\mathrm{SRE} = \frac{1}{z}\overset{n,1}{\underset{i=2}{\square}}\frac{j\square_r(i + 1)\ ,\square_p(i + 1)j}{\square_r\,(i + 1)} \tag{1.5}$$

Where $\square_r(i + 1)$ is the number of errors detected by time t_{i+1} and $\square_p(i + 1)$ is the number of errors the model predicts would be found by t_{i+1} when the first i data points are used fit the model.
The z term is used to gain the common of all errors predictions. Ideally, z = n , 2, indicating that we acquired a prediction mistakes for all subsets of the statistics along with the number one statistics points as an awful lot as the following to final information factor. Unfortunately, experience indicates that in some instances it could not be viable to healthy the parameters of a version to a records set for all data units or subsets of information sets.

4.3 Long Term Predictive Capability: MRE

Software reliability fashions are often employed to set schedules on the start of the test phase, making their long term predictive capabilities probable extra critical than their short term capabilities. We will measure long time predicative accuracy in a way just like brief term accuracy; using the Mean Relative Error or MRE. The MRE of a version applied to a statistics set may be discovered with the aid of becoming the version parameters to the information the use of simplest the primary data factors. Then the geared up version parameters are used to assume the quantity of errors as a manner to be located via the give up of trying out. The absolute charge of the difference among the anticipated charge and the real amount of faults decided on the cease of trying out is then divided through the use of the variety of faults surely discovered to locate the mistake term. The manner is repeated with the primary three statistics points, the first 4, and so forth. All blunders phrases are averaged over the variety of mistakes predictions made to discover the final MRE.
Formally,

$$\begin{array}{lll} & 1n & \underline{j^{N}0\ ,\square\ \ f\,(^{i)j}} \\ \mathrm{MRE} = & \bar{z}\,\square & N \\ & \ \ i{=}2 & 0 \end{array} \qquad (1.6)$$

WhereverN$_0$ is the total numeral of fault detected at the finish of testing and □ $_{f(}$i) is the number of faults that the model predicts will includeby the closing stages of testing, based on appropriate and adjusting the model based on the initiali data points. Over again, the z term is used to average over all predictions made when some subsets of the data will yield illegal parameters values; for the MRE z = n, 1 is the ideal case.

5. CONCLUSION

The conclusion of the paper is that we can replace the obsolete reliability estimation model with the new models and we can develop better quality software that will be easy to update and consistency can be maintained at peak level. It also authenticate to data sets.

6. REFERENCES

[1] Iannino A., Musa J. D., Okumoto K., and Littlewood B., “Criteria for software reliability model comparisons” ACM SIGSOFT Software Engineering Notes, vol. 8, no. 3, pp. 12-16, 1983;

[2] Lyu M. R. and Nikora A., “Applying Reliability Models More Effectively” IEEE Software Transactions., vol. 9, no. 4, pp. 43-52, 1992;

[3] Whittaker J. A. and Voas J., “Toward a More Reliable Theory of Software Reliability” Computer, vol. 33, no. 12, pp. 36-42, 2000; IEEE (ANSI) Standard 982.2/1988. Software Reliability Terminology.

[4] Raoul Vallona,b,⍰ , Bernardo José da Silva Estácioc , Rafael Prikladnickic , Thomas Grechenigb, “Systematic literature review on agile practices in global software development”,Information and SoftwareTechnologyVolume 96, April 2018, Pages 161-180

[5] Deepak Arora, PoojaKhanna, AlpikaTripathi, Shipra Sharma, SanchikaShukla (2011) “Software Quality Estimation through Object Oriented Design Metrics”, International Journal 100 of Computer Science and Network Security, Volume 11, No.4.

[6] Huang, N., Wang, D., Jia, X., 2008. FAST ABSTRACT: an algebra-based reliability prediction approach for com-posite web services, 19th International Symposium on Software Reliability Engineering, pp. 285–286.

[7] Dimov, Aleksandar, Sasikumar, Punnekkat, 2010. Fuzzy reliability model for component-based software systems, 36th EUROMICRO Conference on Software Engineer-ing and Advanced Applications, pp. 39–46.

[8] Tyagi, K., Sharma, A., 2012. A rule-based approach for estimating the reliability of component-based systems.Adv. Eng. Softw. 54, 24–29

[9] R.Chinnaiyan and Dr.S.Somasundaram,2008, "RELIA-BILITY ASSESSMENT OF COMPONENT BASED SOFTWARE SYSTEMS USING TEST SUITE - A RE-VIEW", Journal of Computer Applications, Vol – 1, No.4, Oct – Dec 2008, 34-37.

[10] Fazal-E-Amin, et al., "A Review of Software Component Reusability Assessment Approaches," Research Journal of Information Technology, vol. 3, pp. 1-10,20 1 1.

[11] Seth, K., Sharma, A., Seth, A., 2010. Minimum spanning tree-based approach for reliability estimation of COTSbased software applications. IUP J. Comput. Sci. 4 (4), 13–21. [29] J. Aguilar, J. Cordero, L. Barba, M. Sanchez, P. Valdiviezo, L. Chamba, Learning analytics tasks as services in smart classroom, UniversalAccessInf.Soc.J.2017.

[12] Varun Kumar and Deepak Chhabra, “Designed the Fuzzy Logic Controller for Active Vibration Control of Cantilever Plate with Piezo -Patches As Sensor /Actuator” International Journal of Emerging Research in Management &Technology, Volume-2, Issue-8,pp.34-44,2013.

[13] Varun Kumar andAmit Kumar ‘Design of Adaptive neuro-fuzzy inference system (ANFIS) Controller for Active Vibration control of Cantilever Plate with Piezo -Patches as sensor /actuator’INTERNATIONAL JOURNAL OF R&D IN ENGINEERING, SCIENCE AND MANAGEMENT vol.1, issue I, AUG.2014.

[14] Varun Kumar, ‘An Emerging Controlling Process of Vibration with help of Fuzzy Logic Controller for a Cantilever Plate’ International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463 Vol. 2 Issue 12, December-2013, pp: (144-151).