Improvement of the Software Reliability Model with Equivalent Failure Times

YONGQIANG ZHANG[1], WEN CHEN[2], SHENGJUAN SUN[3]
[1,3]School of Information and Electricity-Engineering
Hebei University of Engineering
[1,3]Guangming South Street 199#, Handan city, Hebei Province, 056038 P.R.China
[2]Department of Computers, Fuzhou Polytechnic
[2]Fuzhou University Zone, FuZhou city, Fujian Province, 350108 P.R.China

Abstract: - The concept of equivalent failure times is put forward and the derivation of it is given in this paper. Moreover, we demonstrate the rationality of improving software reliability model using equivalent failure times, and show its application method by example.

Key-Words: - Software Reliability Model; Equivalent Failure Times

1 Introduction

How to estimate the reliability of the models with the known software failure data is an important tusk in study of software reliability engineering. Extract more information reflecting the characters of software system from the known failure data helps to understand the software system and improve the software reliability model. So the concept of equivalent failure times is presented to reflect the internal characters of the failure data. In this paper, an example is given about JM model to show the formula of computing equivalent failure times. Moreover, the method of improving the model using equivalent failure times is introduced combining the example.

2 JM Model

2.1 Basic Assumptions [1]

(1) The inherent fault number N_0 in the program is an undetermined constant.

(2) Every fault in the program is independent, and has same chance of causing system failures. Every interval between failures is independent as well.

(3) A fault is corrected instantaneously without introducing new faults into the software.

(4) The fault detection rate remains constant over the intervals between fault occurrences. The rate of fault detection is proportional to the current fault content of the software. During the ith testing interval, the rate of fault is $\lambda(x_i) = \phi(N_o - i + 1)$, where ϕ is proportional constant; x_i is a time variable starting from the i-1th failure in the ith failure interval.

(5) The software is operated in a similar manner as that in which reliability predictions are to be made.

With probability theory the conclusion about JM models can be easily obtained, that is, after the i-1 failure of system, the ith failure time follows negative exponential distribution with a parameter $\phi(N_o - i + 1)$, the density function is:

$$f(t) = \phi(N_o - i + 1)\exp\{-\phi(N_o - i + 1)t\} \tag{1}$$

The reliability function is:

$$\exp\{-\phi(N_o - i + 1)t\} \tag{2}$$

The expected interval between the ith failure and the i-1th failure is:

$$MTBF_i = \frac{1}{\phi(N_o - i + 1)} \tag{3}$$

The undetermined parameter N_o and ϕ can be estimated with maximum likelihood estimate. That is, N_o and ϕ satisfies the following two formulas:

$$\sum_{i=1}^{n} \frac{1}{N_o - i + 1} = \frac{n}{N_o - \frac{1}{t}\sum_{i=1}^{n}(i-1)x_i} \tag{4}$$

$$\phi = \frac{n}{N_o t - \sum_{i=1}^{n}(i-1)x_i} \tag{5}$$

Where, $t=\sum_{i=1}^{n} x_i$.

2.2 FC-shaped JM model

In JM model mentioned above, demand one fault in one testing interval. If n_i faults are found in the ith test, FC-shaped JM model can be applied, and the corresponding formula computing the parameter N_o, ϕ is amended as:

$$\sum_{i=1}^{m}\frac{n_i}{N_0-M_{i-1}}=\frac{(\sum_{i=1}^{m} n_i)(\sum_{i=1}^{m} x_i)}{\sum_{i=1}^{m}(N_0-M_{i-1})x_i} \quad (6)$$

$$\phi=\sum_{i=1}^{m} n_i \Big/ \sum_{i=1}^{m}(N_O-M_{i-1})x_i \quad (7)$$

Where n_i is the actual fault number in the ith testing interval; $M_i=\sum_{j=1}^{i} n_j$.

3 Equivalent Failure Times JM Model

3.1 Presentation of Equivalent Failure Times

In the JM model, the most basic assumption is: the rate of fault detection is proportional to the current fault content of the software. This assumption has reasonability to a certain extent, but deficiencies also, which is caused by:

(1) With the progress of the software testing, the experience of the testing personnel can't be neglected; [2]

(2) Software is differing from other product, sometimes finding one fault help to finding some other faults;

(3) While a fault is corrected, some new faults may be introduced.

The traditional models always make assumptions that the process of failures must submit a classical probability distribution, neglecting the other random factors of testing processes, which has been practically testified this method is not right [3]. That is, the rate of fault detection is not proportional to the current fault content of the software. Sometimes finding one fault is equivalent to finding several faults, while sometimes less than one fault, even negative. Therefore, we call the equivalent faults "equivalent failure times". And the ith equivalent failure times is noted as eqn_i . Lower part is the computation way of eqn_i .

3.2 Computation of equivalent failure times for JM model

Let n represents sample number. The method calculating the equivalent failure times from 1 to k samples is the following:

For samples 1--n，2--n，3--n，……，k+1—n, using computing method of JM model' parameters respectively, the total number of the corresponding faults is N_0 (1)， N_0 (2)， N_0 (3)，……， N_0 (k+1).

Let dn(i)= N_0 (i)- N_0 (i+1)

$$sn=\sum_{i=1}^{k} dn(i)=N_0(1)-N_0(k+1) \quad (8)$$

And then, dn(i) can be understood as: correcting one fault makes system decrease dn(i) faults. So the ith fault is equivalent to dn(i) faults. The can be defined as dn(i).

Considering the actual fault times is k, while both k times dn(i) and $\sum_{i=1}^{k} dn(i)$ are not equal k, dn(i) is necessary to be normalized to make $\sum_{i=1}^{k} dn(i)$ =k and relative proportion of k times dn(i) keeps fixedness. So the normalized equivalent failure times is defined as:

$$eqn_i=\frac{dn(i)}{sn}*k=\frac{N_0(i)-N_0(i+1)}{N_0(1)-N_0(k+1)}*k \quad \text{(i=1…k)} \quad (9)$$

For FC-JM model, using the same analysis, the equivalent failure times is calculated as:

$$eqn_i=\frac{dn(i)}{sn}*\sum_{j=1}^{k} n_j=\frac{N_0(i)-N_0(i+1)}{N_0(1)-N_0(k+1)}*M_k \quad (10)$$

In the cause of computing the equivalent failure times, N_0 and ϕ need to be computed time after time (usually with iterative method). For the anti-convergent data, some special handling should be performed (like merging the two adjacent failure interval). In addition, the corresponding current fault number based on the varying samples can be used to confirm the equivalent failure times.

3.3 Application of the equivalent failure times

The front introduction is the way of computing the equivalent failure times eqn_i . After replace the original failure times with eqn_i , computing the model parameters using the FC-JM model, and then we can get the improved N_0 and ϕ . The reasonability is showed as follows:

(1) Influence for replacing the original data with the forecasting data of the models.

At first, assume the original failure times are a_1、 a_2 ...... a_n , which is noted 1. What are replaced with the equivalent failure times are a_1、 a_2 ...... a_n . We can compute the parameters for JM model to get the model f1 by the least square method or the maximum likelihood estimate method. The forecasting value calculated from f1 is b_1 , $b_2 \dots b_k$, $b_{k+1} \dots b_n$, and the error of f1relative to data 1 can be denoted as:

$$e_{1-1} = \sum_{i=1}^{k} (b_i - a_i)^2 + \sum_{i=k+1}^{n} (b_i - a_i)^2 . \quad (11)$$

Next, after replacing the front k of data 1with b_1 , b_2 ... b_k , we get data 2: b_1 , b_2 ... b_k , a_{k+1} ... a_n . Then the error of f1 relative to data 2 is:

$$e_{1-2} = \sum_{i=1}^{k} (b_i - b_i)^2 + \sum_{i=k+1}^{n} (b_i - a_i)^2$$

$$= \sum_{i=k+1}^{n} (b_i - a_i)^2 \quad (12)$$

Finally, compute the parameters of FC-JM model using data 2 by least square method to get the model f2, which forecasting value is c_1 , $c_2 \dots c_n$. And then the error of f2 relative data 2 is

$$e_{2-2} = \sum_{i=1}^{k} (c_i - b_i)^2 + \sum_{i=k+1}^{n} (c_i - a_i)^2 . \quad (13)$$

From the principle of the least square method, the error of f2 relative data 2 reaches minimum. So $e_{2-2} \le e_{1-2}$, that is

$$\sum_{i=1}^{k} (c_i - b_i)^2 + \sum_{i=k+1}^{n} (c_i - a_i)^2 \le \sum_{i=k+1}^{n} (b_i - a_i)^2 , \quad (14)$$

Therefore $\sum_{i=k+1}^{n} (c_i - a_i)^2 \le \sum_{i=k+1}^{n} (b_i - a_i)^2$.

And then, the conclusion can be got: after replacing the front k with the new data, compute the new model with the new data, and the error of the new model relative to the original data decreases.

(2) Relationship between forecasting value b_i of model and the equivalent failure times.

To JM models, for example, assume under ideal circumstances, the failure data fully complies with the undetermined model. That is, the various data got from the models will be identical with the actual data and the various processing to the data of models and to the actual data is the same.

So the dispersion between N_0 (i+1) with samples i---n and N_0 (i+1) with samples i+1-----n should accurately equal to the ith failure times, that is $N_0(i) - N_0(i+1) = a_i$. However, in practice, the equation isn't satisfied sometimes because the model is not consistent with actual data entirely. Consequently, we approximatively consider $N_0(i) - N_0(i+1)$ is the undetermined ith failure times (b_i), which is the equivalent failure times not normalized. It shows the forecasting value b_i of the model can be replaced the equivalent failure times approximatively.

Furthermore, the above discussion is based on the least square method of the failure times. But in practice, the least square method for the interval data between the failures or the maximum likelihood estimate method is most often used. The error of the parameters derived from the different approaches. And therefore, the practical model is the approximate replacement for the foregoing model.
Based on the above, after replacing the first half of the original data with the equivalent failure times, re-gaining models. New models for the latter part of the original data will reduce data errors, and it can be expected that this approach is reasonable. Documentation [4] proposed "change point" and divided the failure into stages to describe using different models, and the size of the equivalent failure times is precisely specific manifestations of identity of the "change points". In documentation [5], the assumption for completely fault correction in JM model is changed, and the parameter variables are introduced. JM model is improved as:

$$P(t_i) = \phi[N_0 - \mu_{i-1}(i-1)] \exp\{-\phi[N_0 - \mu_{i-1}(i-1)t_i]\} , \quad (15)$$

Which thought is similar with the equivalent failure times.

4 Examples

We are still using the JM model on this point. Quote the error statistics data if naval tactical data system NTDS in U.S. Navy Fleet computer programming center development process. The original data is shown in table 1. JM models is used (n take 27, k to

21) to compute the parameters N_0 and ϕ, the results is shown in Table2.

Table1 Failure Data

Sequence number	1	2	3	4	5	6	7	8	9	10
Actual failure interval (per day)	9	12	11	4	7	2	5	8	5	7
Sequence number	11	12	13	14	15	16	17	18	19	20
Actual failure interval (per day)	1	6	1	9	4	1	3	3	6	1
Sequence number	21	22	23	24	25	26	27	28	29	30
Actual failure interval (per day)	11	33	7	91	2	1	87	47	12	9

Table2 Results

samples	N_0	ϕ
1− 27	28.1942917170663	8.35531423446163E-03
2− 27	26.8828137195636	9.04170797301899E-03
3− 27	25.4459691888116	1.03522829562772E-02
4− 27	24.1241015373118	.011782894461442
5− 27	23.1047223647998	1.18892453822767E-02
6− 27	21.9775074067755	1.26772862471688E-02
7− 27	21.0364528644514	1.22904461633291E-02
8− 27	19.9944984530646	1.25673843036225E-02
9− 27	18.856261161429	1.36406366404687E-02
10—27	17.8209723669244	1.39648938089898E-02
11—27	16.7320124802926	1.49064314672362E-02
12—27	15.8215200712629	1.39430593816926E-02
13—27	14.7761917215888	1.44118544174037E-02
14—27	13.8957292016348	1.32451100116188E-02
15—27	12.7795991550042	.014398054094112
16—27	11.8249961089191	1.39058037090416E-02
17—27	11.007245617513	1.22798669701364E-02
18—27	10.1966830975396	1.10059757984008E-02
19—27	9.51255935650725	9.43889258950176E-03
20—27	8.86845774826952	8.16867784416515E-03
21—27	9.33748221783454	5.33823078065302E-03
22—27	10.657588649157	3.48568266296093E-03

Replace the failure times from 1 to 21 with the equivalent failure times, the result is shown in Table 3.

Table3 Results after Replacement

Sequence number	Actual failure interval (per day)	equivalent failure times
1	9	1.57047980118649
2	12	1.72060478123773
3	11	1.58292129221795
4	4	1.22069482158953
5	7	1.34982693308123
6	2	1.12690197879721
7	5	1.24772841020283
8	8	1.36302605066567
9	5	1.23974641073665
10	7	1.3040169255711
11	1	1.09030417608047
12	6	1.25176866245317
13	1	1.0543437296871
14	9	1.33655287932272
15	4	1.14312615604878
16	1	.979246797589533
17	3	.970639284563703
18	3	.819230302642694
19	6	.771304259450235
20	1	-.561651401789956
21	11	-1.58081225133486
22	33	1
23	7	1
24	91	1
25	2	1
26	1	1
27	87	1
28	47	1
29	12	1
30	9	1
31	135	1

By FC—JM model, N_0 = 27.2078, ϕ=1.03887E-02.

5 Evaluation for the Improved Model

The forecasting value of the interval between ith failure and the i-1th failure is $MTBF_i = \frac{1}{\phi(N_o - i + 1)}$.

Now compare the result with the sum of absolute value and the quadratic sum of the error.

$$SE1 = \sum_{i=k+1}^{n} | y_i - MTBF_i | \tag{16}$$

$$SE2 = \sum_{i=k+1}^{n} (y_i - MTBF_i)^2 \tag{17}$$

The result is:

JM model: N_0 = 28.1943, ϕ = 8.35531E-03, SE1=192.104, SE2= 8125.424.

Improved model: N_0 = 27.2078, ϕ=1.03887E-02, SE1=175.012, SE2=7731.197.

Thus, improved SE1, SE2 value is smaller than the former, which shows the method proposed in this paper can commendably improve the model.

6 Conclusion

In this paper, provide the equivalent failure times; prove the reasonability that the equivalent failure times is used to improve software reliability models; show the formula computing the equivalent failure times in the example about JM model; and illuminate the method improving models with the equivalent failure times. The equivalent failure times represents the relative ratio, which disclosures the internal characters of every failure times in software system. And the equivalent failure times can be used to improve other models such as model based on unascertained theory [6]. Overall, further research on the equivalent failure times will help to more exactly analyze the reliability of the software. It is worthy of our further study.

Acknowledgement:
Project 60573088 supported by National Natural Science Foundation of China (NSFC); Project 603407 supported by National Natural Science Foundation of Hebei Province.

References:

[1] Xizi Huang. Software Reliability,Security and Quality Assurance[M]. *Publishing House of Electronics Industry*, 2002, pp.86-112..
[2] Hongwei Liu, Xiaozong Yang. A Software Reliability Growth Model with Bell-Shaped Fault Detection Rate Function [J]. Chinese Journal of Computers, 2005, 28 (5) P.908-913
[3] Feng-zhong Zou, Chuan-xiang Li. A Dual Stochastic Function Model for Software Reliability [J]. JOURNAL OF SOFTWARE. 1998, 9(1):69~73
[4] Phame H. Reliability Handbook [S]. *Springer Verlag*. New York. 2002..
[5] Muyun Fang, Yugui Qu. An Improvement on the Assumption of Thorough Faults-removal in the JM Reliability Model[J]. *Mini-micro Systems*, 2001, 22(2): pp.199-200.
[6] Yongqiang Zhang, Shengjuan Sun, Fujun Liu. Study of Uncertain Factors in Software Reliability with Mathematical Method [C]. The 9th International Computer Conference 2003: P138~141