The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2018
24-27 May, 2018, Kyiv, Ukraine

The Secondary Software Faults Number Evaluation Based on Correction of the Experimental Data Exponential Line Approximation

Oleksandr Rudenko[1], Elena Odarushchenko[1], Zinaida Rudenko[2], Maryna Rudenko[3]
[1] Poltava National Technical Yuriy Kondratyuk University, Ukraine, {elena.odarushchenko, olexantr1}@gmail.com
[2] Poltava Oil and Gas College of Poltava National Technical Yuriy Kondratyuk University, Ukraine, sina.mar.tar@gmail.com
[3] Kyiv National Linguistic University, Ukraine, mar22.rr@yandex.ua

***Abstract* — The paper presents an analysis of existing approaches to the secondary software faults problem. The new method for secondary software faults assessment is considered. It is based on the comparison of secondary software faults statistics and the data from corrected exponential approximation line of faults trend. A comparative analysis of the obtained results with the results obtained by estimating the number of secondary faults by the statistical data of the detected faults using the regression line, which characterizes the indirect proportion of the number of detected faults to time, is carried out.**

***Keywords* — software reliability; fault; secondary software fault; exponential line approximation; software reliability growth model**

INTRODUCTION

The widespread use of information technologies in various human activities sets advanced tasks to the software developers. The reliability is known to be one of the core factors that depend on competitiveness and efficiency of nowadays computer technologies. Therefore, an accurate software reliability analysis is an extremely important task.

Analysis of software reliability parameters is usually performed through models, which are based on assumptions and analytical expressions. During the last decade, in the works [1]-[9] attention was drawn to the fact that most models do not contain an assumption of secondary software faults presence – faults, which can be entered during the elimination of identified (primary) faults. Additionally, the number of models assumes that no new faults are entered while eliminating an identified faults (Jelinsky-Moranda model, exponential model, Shick-Walterton model).

There are four approaches to evaluate the problem of secondary software faults. The studies conducted in [2], [4]-[7] are aimed to explain the physical nature of secondary software faults. The basics of the software dynamics theory as an approach to reliability assessment that accounts the presence of secondary faults were developed. Consideration of software faults display process as the result of determined faults flow is the main feature of software dynamic (unlike the probability parameters are known from reliability theory). The second approach is based on the time series theory and allows detaching secondary faults from the overall faults flow [3]. The third one – is the use of simulation. Software faults detection and appearance of new during the elimination of identified ones are reproduced by the model generator. The generator uses simulation statistics that similar to the real projects statistics [8]. The fourth approach proposes to use modified software reliability models, which accounts the secondary software faults parameter in the risk function. This parameter presents the number of secondary faults and can be determined by comparing the faults trend values to the corresponding values of the regression line. Higher probability of software faults during the early testing stages is considered [9].

The secondary defect factor is one of the components of imperfect debugging. A number of works are devoted to the issues of imperfect software debugging. The models that take various parameters of this process are developed in these works. In the work [10] is proposed the software reliability growth model linking, with a common parameter between these two functions, the phenomenon of imperfect debugging and learning. Model of software reliability growth, based on heterogeneity of Poisson process taking into account imperfect debugging is proposed in the work [11]. A simulation approach to modelling the fault detection and fault correction processes is proposed in the work [12]. Debuggers with different contributions to the speed of detection and correction of defects are considered. The detection process and the process of defect correction with the

978-1-5386-5902-1

inclusion of the function of testing the effort and imperfect debugging are considered in the article [13]. The inclusion of the force testing function and the introduction of defects in their detection is shown. Pair models of the processes of detecting and correcting defects are obtained on the basis of various assumptions about the introduction and correction of defects. The work [14] presents a model for describing software failures during debugging. The model allows using an imperfect debug script considering the possibility of implementation the new errors in the software at the design stage. Bayesian analysis of the model is developed and the derivation of an unknown number of states of a hidden Markov model is shown. In the article [15] is considered simulation and Bayesian estimation of the multiplicative failure intensity, components of which evolve stochastically at the testing stages. Attention to the modelling of parameters is focused – frequency of failure detection and a number of malfunctions. An explanation of the concept of imperfect debugging with the help of the model under certain conditions is given. The article [16] considers the software reliability growth model including the generalized modified test function Weibull in an imperfect debugging environment with constant and time-varying error detection coefficients. Parameters used in the models are estimated using the maximum likelihood estimation method and the nonlinear least squares method. The performance of the proposed models is checked using the mean-square error, the accuracy of the evaluation, the χ^2 - test.

In works devoted to imperfect debugging of software the factor of secondary defects is sometimes taken into account in a complex with other factors characterizing imperfect debugging but is not separately allocated.

It should be noted that the main problem in evaluating the number of secondary faults is the lack of secondary faults statistics. Faults are detected, fixed, but no statistics are directly kept on the secondary faults. It is only possible therefore to estimate their number in time intervals but not the accurately assess their number.

The paper presents a method of estimating the number of secondary software faults based on the comparison of adjusted exponential line approximation to the faults statistics data.

MAIN PART

The choice of exponential approximation is explained by the fact that parameters that characterize reliability, which are used in many models of software reliability assessment, are calculated using formulas containing an exponential component.

Examples of such models are the Musa-Hamilton model [18], the Schneidevind model [19], the basic S-shaped model [20], the Ohba model [21], and others.

In the Musa-Hamilton model [18], the number of failures (identified faults) is calculated by the formula

$$m(\tau) = E_0\left(1 - e^{-\frac{C}{E_0 T_0}}\right), \tag{1}$$

where T_0 – pre-operating time between failures before starting the setup or operation, E_0 – the initial number of faults, C – the coefficient of proportionality.

The process of faults identification in the Schneidewind model [19] is considered as a heterogeneous Poisson process with a fault identification level that decreases exponentially. The bounce rate has the following form

$$\lambda(t) = Be^{-Ki}, \tag{2}$$

where i – is the sequence number of the interval, $B>0$, $K>0$ – are the constants of the model.

The basic S-shaped model [20] is specifically designed to determine the reliability of software and does not proceed from the hardware reliability theory unlike a number of other models. Intensity of failures

$$\lambda(t) = BK^2 te^{-Kt}, \tag{3}$$

where B, $K>0$ – the parameters of the model.

The generalization of the S-shaped model is the Ohba model [17], the intensity of which failures

$$\lambda(t) = ab^2 te^{-bt}, \tag{4}$$

where a – the initial number of faults in the software, b – the intensity of fault elimination.

The method for estimating the number of secondary faults contains the following sequence.

1 Based on the trend of the existing statistics of identified faults, an exponential approximation line is constructed and its equation is determined.

As a rule, fault statistics is obtained by detecting their number in some equivalent time intervals. Trend of faults is a polyline which consistently combines points with coordinates $(t_i;n_i)$, where t_i – the time of the end of i time interval; n_i – the number of faults identified during the i time interval.

2 From the exponential approximation equation, the points $(t_1;n_1)$ and $(t_k;n_k)$ are determined, where t_1 is the time of the end of the first time interval; t_k – time of the end of the last time interval; n_1, n_k – the corresponding values of the approximating function.

3 Definition of an exponential equation which is passing through points with coordinates $(t_1; n_1 - \sigma)$ and $(t_k; n_k)$, where σ – standard deviation of the number of identified faults.

Adjustments of the approximating line is based on the fact that on the early stages of software testing the probability of getting secondary faults is higher, because the number of identified faults is higher and therefore more action to eliminate them are conducted.

4 The values of the adjusted exponent are at the time points t_i $(i = \overline{1,k})$.

5 The number of secondary faults in time intervals is determined by the formula

$$n^{\text{III}} = n_i - n_i^*, \; i = \overline{1,k}, \quad (5)$$

where n_i – the number of faults detected by the i time interval, n_i^* – the value of the corrected line of exponential approximation at time points t_i.

6 The result of calculations by formula (5) is rounded up to the integers, in addition, in cases where a negative result is obtained n^{III} considered to be zero.

Let us consider an example of the application of the proposed method for estimating the number of secondary faults on the basis of statistics, described in studies [17], [22]-[24] (Table 1).

Observations were conducted during the year. 67 faults were detected during this time. These fault numbers are monthly grouped.

TABLE 1. FAULT STATISTICS

Month	Number of detected faults
1	13
2	11
3	8
4	7
5	6
6	5
7	4
8	6
9	3
10	2
11	1
12	1

According to fault statistics the trend of faults is created, the exponential trend line (Fig. 1) was constructed, its equation was determined

$$n = 18{,}17 e^{-0{,}22t}. \quad (6)$$

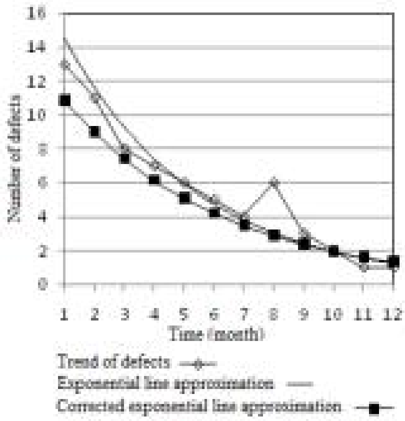

Figure 1. Trend line of fault number, exponential approximation and adjusted exponential approximation

Substituting in (6) instead of t 1 and 12, we determine the points $(1; 14{,}58)$ and $(12; 1{,}3)$. Calculating the standard deviation of the number of identified faults $\sigma = 3{,}78$, we obtain a point through which the corrected line of exponential approximation runs $(1; 10{,}8)$.

In accordance with 5[rd] method paragraph, we substitute the equation

$$n = \alpha e^{-\beta t} \quad (7)$$

instead of the t and n point coordinates $(1; 10{,}8)$ and $(12; 1{,}3)$ getting a system of equations for finding the parameters α and β of the equation of the corrected line of exponential approximation

$$\begin{cases} \alpha e^{-\beta} = 10{,}8 \\ \alpha e^{-12\beta} = 1{,}3 \end{cases} \quad (8)$$

By solving the system of equations (8) and substituting (7) the found values of the parameters α and β we get the desired equation

$$n = 13{,}09 e^{-0{,}19t}. \quad (9)$$

Finding n_i^* substituting in (9) values from 1 to 12. Using the formula (5) after. The results of the corresponding calculations are shown in Table. 2.

TABLE 2. CALCULATION OF THE NUMBER OF SECONDARY FAULTS

Month	Number of detected faults	x_i^*	$n^{вт}$
1	13	10,82	2
2	11	8,95	2
3	8	7,4	1
4	7	6,12	1
5	6	5,06	1
6	5	4,19	1
7	4	3,46	1
8	6	2,86	3
9	3	2,37	1
10	2	1,96	0
11	1	1,62	−1(0)
12	1	1,34	0

In the last column of the table 2 an estimate of the number of secondary faults in each time interval is shown. In accordance with paragraph 6 of the sequence of estimating the number of secondary faults $n^{вт}$ in the eleventh month is considered to be zero, although in the calculations the obtained value is equal to 1.

Method of estimating the number of secondary faults provided the absence of non-random factors that influence the appearance of faults is described [17], [23], [24]. This method has the following sequence:

1 According to statistics of the number of faults in each time interval, using the least squares method, the equation of the regression line is determined, which characterizes the inverse proportional dependence of the average number of identified faults on time.

2 The absolute number of the difference between the value of the fault statistics and the value of the regression function at the moment of defining the fault statistics is calculated.

3 The number of secondary faults is calculated as the difference between the result obtained in paragraph 2 and the standard deviation in the statistics of the number of faults multiplied by the coefficient $(k+1-x)^{-1}$ where k – the number of test intervals (the number of identified faults), x – the sequence number of the interval (the identified fault). The result is rounded to the integer.

The coefficient is caused by a higher probability of entering of secondary faults at the initial stages of testing (exploitation) of software.

Research of various statistics of accounting information system has shown the existence of the inverse relationship between the sequence number of the time interval and the number of faults, detected on the corresponding interval. Accordingly the number of secondary faults is evaluated using the formula

$$n^{вт} = \left|y - a - \frac{b}{x}\right| - \frac{1}{k+1-x}\sigma_y, \quad (10)$$

where y – the number of detected faults in a time interval; x – sequence number of time interval; a, b – coefficients of the equation of the regression line $y = a + b/x$; k – number of time intervals; σ_y – standard deviation on y; $n^{вт}$ – estimated value of the number of secondary faults, rounded to the whole $\left(n^{вт} \geq 0\right)$.

As a rule, the fact of making of faults at the last intervals of testing (exploitation) of software is not taken into account if the result of paragraph 3 obtained at the value of the regression function is bigger than the corresponding empirical value, since the number of identified faults in most cases is small (or absent at all), respectively, and the probability of fault entering is low.

In table. 3 the results of calculating the number of secondary faults by two methods are shown.

TABLE 3. COMPARISON OF THE NUMBER OF SECONDARY FAULTS CALCULATED BY TWO METHODS

Month	$n^{вт}$ (use of approximation)	$n^{вт}$ (use of regression)
1	2	2
2	2	2
3	1	1
4	1	1
5	1	1
6	1	0
7	1	0
8	3	1
9	1	0
10	0	0
11	−1(0)	1(0)
12	0	0

From the table 3 we conclude that the results of the first five and last three time intervals coincide, the average intervals have a deviation in 1 fault and only the results of the eighth time interval don't coincide. The incommensurability of the results in the eighth time interval can be explained by the fact that the results of calculations of the number of secondary faults in both methods are rounded to the whole (in the first case the result is 2,86 (Table 2), in the second one, 1,38 [23]), that is, the rounded difference between these values is 1 fault.

Consequently, we can assume that the results obtained by the proposed method, in general, coincide with the results obtained by the method using the regression line.

CONCLUSIONS

The method of estimating the number of secondary faults according to the statistics of faults numbers, which is based on the comparison of the fault statistics with the values of the corrected line of exponential approximation, was developed.

The results are consistent with the previously obtained results by comparing the values of the trend of faults with the corresponding values of the regression line, but the proposed method is technically simpler.

Further directions of research aimed at estimating the number of secondary faults by statistical data is the use of other types of approximating functions, as well as taking into account the accuracy of approximation in calculations.

REFERENCES

[1] Maevsky D. A. Influence of Secondary Faults on the Reliability of Dynamic Information Systems / D. A. Maevsky // Bulletin of the NTU "KhPI". - 2012. - 50 (956). - P. 54-58.

[2] Maevsky D. A. Structural dynamics of software systems and prediction of their reliability in the presence of secondary faults / D. A. Maevsky // Radioelectronic and computer systems. - 2010. - No. 3 (44). - P. 103-108.

[3] Maevsky D. A. The use of the theory of time series to distinguish the authoritative errors at the stage of software testing / D. A. Maevsky, O. P. Zhekov // Radioelectronic and computer systems. - 2011. - No. 2 (16). - P. 82-85.

[4] CASE-evaluation of critical software systems. T. 2. Reliability [Monograph] / Odarushchenko O. N., Kharchenko V. S, Maevsky D. A. and others. - Ed. Kharchenko V. S. - Kh. Nat. aerospace Un-t them. NOT. Zhukovsky "KhAI", 2012. - 292 p.

[5] Maevsky D. A. Modeling of reliability in the theory of dynamics of software systems / D. A. Maevsky // Electrotechnical and computer systems. - 2011. - Issue (04) 80 - P. 147-153.

[6] Maevsky D. A. Software reliability. Non-probabilistic approach / D. A. Maevsky, H. D. Maevskaya, A. A. Leonov // RT&A # 03 (26) - 2012, P. 8-20.

[7] Maevsky D. A. A New Approach to Software Reliability / Dmitry A. Maevsky // Lecture Notes in Computer Science : Software Engineering for Resilient Systems. – Berlin : Springer, 2013. – № 8166. – P. 156–168.

[8] Mishchenko V. O. The thermodynamic approach to modeling the process of increasing the reliability of software taking into account "secondary faults" / V. O. Mishchenko // Bulletin of V. N. Karazin Kharkiv National University, series "Mathematical modeling. Information Technology. Automated Control Systems" - 2015. - Issue 28. - P. 91-106. .

[9] Odarushchenko O. N. Counting secondary faults in software reliability models / O. N. Odarushchenko, A. A. Rudenko, V. S. Kharchenko // Mathematical Machines and Systems, 2010, No. 1. - P. 205-217.

[10] Pham H. An Imperfect-debugging Fault-detection Dependent-parameter Software. International Journal of Automation and Computing. 2007; 04(4): 325-328

[11] Mane M., Joshi M., Kadam A., Joshi S.D. Software Reliability and Quality Analyser with Quality Metric Analysis Along With Software Reliability Growth Model. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3803-3806

[12] Peng R., Liu J. Simulated Software Testing Process Considering Debuggers with Different Detection and Correction Capabilities. International Journal of Performability Engineering, Vol. 13, no. 3, 2017, 334-336

[13] Peng R, Li Y.F., Zhang W.J., Hu Q.P. Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction. Reliability Engineering & System Safety. 2014; 126: 37-43.

[14] Pievatolo A., Ruggeri F., Soyer R. A Bayesian hidden Markov model for imperfect debugging. Reliability Engineering and System Safety, vol. 103, pp. 11–21, 2012.

[15] Aktekin T., Caglar T. Imperfect debugging in software reliability: a Bayesian approach. European Journal of Operational Research, vol. 227, no. 1, pp. 112–121, 2013

[16] B. Pachauri B., A. Kumar A., J. Dhar J. Software reliability growth modeling with dynamic faults and release time optimization using GA and MAUT. Applied Mathematics and Computation, vol. 242, pp. 500–509, 2014

[17] Odarushchenko O. N. A method of evaluating the reliability of software taking into account secondary faults / O. N. Odarushchenko, A. A. Rudenko, V. S. Kharchenko // Radioelectronic and computer systems. -2012. - No. 7 (59). - P.294-300.

[18] Hamilton P.A., Musa J. D. Measuring reliability of Computation Center Software // Proc.3-thInternat. Conf. on Software. Eng. May 10-12 1978. - P.29-36.

[19] Schneidewind N. F. Software Reliability Model with Optimal Selection of Failure Data // IEEE Transactions on Software Engineering. - 1993. - Vol. 19, No. 11. Nov. - P. 1095-1104.

[20] Yamada S., Ohba M., Osaki S. S-shaped software reliability grows modeling for software error detection // IEEE Trans. Reliability. - 1983. - R-32. - № 5. - P. 475-518.

[21] Ohba M. Software Reliability Models // IBM J. Res. Develop. - 1984. - 28. - № 4. - P. 428-443.

[22] Antoshchuk S. G. Forecasting the number of errors at the stage of exploitation of adaptable accounting information systems / S. G. Antoshchuk, D. A. Maevsky, S. A. Yaremchuk // Radioelectronic and computer systems. - 2010. - No. 6 (47). - P. 204-210.

[23] Kharchenko V. S. Analysis of scenarios and determination of parameters for assessing the reliability of software taking into account secondary faults / V. S. Kharchenko, O. N. Odarushchenko, A. A. Rudenko, E. B. Odarushchenko // Control, navigation and communication systems. - Kyiv: State Enterprise "Central Research Institute of Navigation and Management", 2011. - Issue 2 (18). - P. 273-280

[24] Kharchenko V. S. Accounting of the secondary defect factor in software reliability assessment/ V. S. Kharchenko, A. A. Rudenko, O. N. Odarushchenko, E. B. Odarushchenko // Scientific statements of the Belgorod State University. History. Political science. Economy. Computer science. - Belgorod: Publishing house "Belgorod", 2013. - № 22 (165). Issue 28/1. - pp. 153-160.