ASSESSMENT OF THE BUGS PROBABILITY IN SOFTWARE SYSTEMS

Nikolay Petrov, Kremena Dimitrova* and Nikolay Kolev

Technical University of Sofia,
Faculty of Engineering and Pedagogy – Sliven,
59, Burgasko Shose Blvd, 8800 Sliven, Bulgaria,
e-mails: nikipetrov_1953@abv.bg; kr7emena@abv.bg;
e-mail: nkolev1965@gmail.com

Abstract. The relevance of the assessment is determined by the requirements of the modern information revolution 4.0. This revolution defines the requirement to minimize the risks of bugs in information processing and management systems. In this regard, a mathematical model has been proposed that allows evaluating the characteristics of bugs (software errors) in software systems. This creates a possibility to predict their reliability in design and operation. Assessing the likelihood (probability) of bugs is an element of overall reliability and influences decision makers about the future use of software systems. The numerical example to the created model is based on the processing of specific experimental data from observations and research. It has a probabilistic nature and reliability of prediction, which largely depends on the accuracy of the initial data and the depth of diagnosis over time.

Keywords: probability, bug, software systems, modeling, Industry 4.0, risk analysis. reliability.

1. INTRODUCTION

Industry 4.0 is related to the implementation of new technological solutions, defining the emergence of the potential of unknown risks [1]. In this sense, events related to the occurrence of bugs in Software Systems (SS) are risk factors for the latter. Existing mathematical models for estimating bugs (errors) in SS are designed to estimate the following parameters [2–4]:

- SS reliability indicators in the setup process;
- The amount of unmanifested bugs;
- The time to detect the next bugs;

*Corresponding author.

DOI: 10.7546/EngSci.LVII.20.02.03

• The time for manifestation of all bugs with a given probability, etc.

Currently, there are several classical mathematical models for SS reliability [2–7]:

≫ Time model of Musa;
≫ S-shaped growing model;
≫ Hossain Dahiya exponential model;
≫ Yamada exponential model;
≫ Weibull exponential model;
≫ Model of Kapur *et al.*

The models listed in this way are given in a suitable mathematical form in Table 1.

Table 1. Conventional software reliability models

Model name	Type of model	$P_{BUG,\Delta t}(t)$	Comments
Time model of Musa	Exponential model	$a(1-e^{-bt})$, $a \geq 0$, $b > 0$ – model parameters	Also known as Goel-Okumoto (G – O)
(G – O) S-shaped model	S-shaped model	$a[1-(1+bt)e^{-bt}]$, $a \geq 0$, $b > 0$ – model parameters	Modification of the G – O model
Hossain Dahiya	Exponential model	$a(1-e^{-bt})/(1+ce^{-bt})$, $a \geq 0$, $b > 0$, $c > 0$ – parameters of the model dynamics	Similar to G – O as $c \to 0$
Yamada exponent	Exponential model	$a(1-\exp(-a(1-\exp(-\beta t))))$ $a \geq 0$, $\alpha > 0$, $\beta > 0$ α, β – coefficient of proportionality	This model takes into account SS bug testing
Weibull exponent	Exponential model	$a(1-e^{-bt^c})$, $a \geq 0$, $b > 0$, $c > 0$	Similar to G – O as $c = 0$
Kapur *et al.*	Exponential model	$a(1-e^{-bpt})$, $p > 0$ – Kapur index	It is caused by poor debugging

The classic models for identification of reliability from Table 1 present the dynamics of the occurrence of bugs (failures and errors) and are suitable for large data samples (large SS). They allow efficient adjustment and testing of software programs, deciding on the feasibility of commissioning. In this way, the threshold of rational activity is established and the corresponding amount of remaining software bugs is determined. In practice, the small sample size, as well as the actual number of detected bugs (upon completion of setup) does not allow the use of classical mathematical models. For this reason it is appropriate to use simpler models whose formal accuracy is close to the accuracy determined by the available inputs for short time intervals [1, 5, 6, 8–16].

The aim of the paper is to create an appropriate model for estimating the probability of bug occurrence at short time intervals, based on the existing classical models for identifying the dynamics of bug occurrence in large SS, Table 1.

2. BASIC HYPOTHESIS AND THESIS

In the formalization of the mathematical model proposed by the authors, a hypothesis is accepted, which follows from [7, 15]: "Real research on the occurrence of bugs in the SS determines an exponential model of change in these events".

Each bug in the studied programs is independent and manifests itself in random moments of time with a constant average intensity (in the absence of corrections in the SS) in all time intervals of their operation. The type of commands executed in the program and the operation time between bugs are determined by the average time for execution of the commands on the respective computer and the average number of commands.

The thesis of the research follows from the adoption of a decentralized approach to studying [8, 9] and analyzing the results obtained in classical works such as [2–4,10,17].

The intensity of bug detection in real functioning programs depends inversely on the average speed of the computer and practically does not depend on the distribution of command types on the data processing routes.

The choice of SS tuning tests should be sufficiently representative. If possible, it should exclude a concentration of undetected errors (bugs) for the actual operating conditions of the programs. The lack of a priori data on artificially increasing the intensity of bugs and their distribution over time should be considered uniform and independent of external factors.

Bugs that cause distortion of computer performance and used SS are either fixed (restricted), or removed after testing, or not detected at all. During testing and adjustment, due to the increased attention to distortion of the performance of computer systems, bugs are much more likely to occur than in the interval of normal operation.

3. ESTIMATION OF THE PROBABILITY OF BUGS

Based on the analysis of information in known scientific papers [7, 13–16, 18–22] it follows that under normal operating conditions of the SS, the number of bugs occurring in the respective service intervals is distributed according to Poisson's law. As a result, we arrive at the conclusion: the duration of the continuous operation of the computer systems and SS, studied in the interval between the occurring bugs, is distributed exponentially at small intervals of time of stationary of these events.

Under these conditions of SS testing and operation, the intensity of detected bugs $\lambda_{DB}(\Delta t)$ for observation interval Δt decreases with increasing total (continuous) time of technical operation τ_{TTO}.

The probability of bugs occurrence $P_{BUG,\Delta t}(t)$ (equivalent to the probability of failure $Q_F(\Delta t)$ in hardware systems) for a time interval Δt will be determined by the basic equation of reliability [7, 13–15]:

$$P_{BUG,\Delta t}(t) + P_{FFO,\Delta t}(t) = 1. \tag{1}$$

In equation (1) the probability of failure-free operation $P_{FFO,\Delta t}(t)$ of SS, tested for time interval Δt at time for its actual technical operation τ_{RTO}, is determined according to the basic law of reliability of systems [15]:

$$P_{FFO,\Delta t}(\tau_{RTO}) = e^{-\int_0^{\tau_{RTO}} \lambda_{DB}(\Delta t)dt} = \exp\left\{-\int_0^{\tau_{RTO}} \lambda_{DB}(\Delta t)dt\right\}, \tag{2}$$

where $\lambda_{DB}(\Delta t)$ is the intensity of the detected bugs in the software system; e is the Napier'constant.

During a stationary period of operation, the intensity of bug detection is $\lambda_{DB}(\Delta t) = \lambda_B \cong \text{const}$. This allows the following transformations to be performed on $P_{FFO,\Delta t}(t)$:

$$P_{FFO,\Delta t}(\tau_{RTO}) = e^{-\int_0^{\tau_{RTO}} \lambda_{DB}(\Delta t)dt} = e^{-\lambda_B \int_0^{\tau_{RTO}} dt} = e^{-\lambda\tau_{RTO}}, \tag{3}$$

where τ_{RTO} is time for real technical operation of SS.

Decomposition of the function $P_{FFO,\Delta t}(t)$ of (3) follows in Taylor's order on the argument τ_{RTO}, assuming $\Delta t = \tau_{RTO}$. We obtain:

$$P_{FFO,\Delta t}(\tau_{RTO}) = 1 - \left[\lambda_B\tau_{RTO} - (\lambda_B\tau_{RTO})^2/2! + (\lambda_B\tau_{RTO})^3/3! + \cdots\right]. \tag{4}$$

Provided that the condition is met:

$$\lambda_B \tau_{RTO} \leq 1, \tag{5}$$

for a corresponding time interval τ_{RTO} it is valid [13, 15]

$$P_{FFO,\Delta t}(\tau_{RTO}) \cong 1 - \lambda_B \tau_{RTO}. \tag{6}$$

Formula (6) is particularly relevant for current engineering calculations of $P_{FFO,\Delta t}(\tau_{RTO})$, both hardware and software systems. From formulae (1) and (4) it follows that the probability of occurrence of bugs $P_{BUG,\Delta t}(\tau_{RTO})$ in SS is a function of λ_B and time for real technical operation τ_{RTO}. We have:

$$P_{BUG,\Delta t}(\tau_{RTO}) = \lambda_B \tau_{RTO} - (\lambda_B \tau_{RTO})^2/2! + (\lambda_B \tau_{RTO})^3/3! + \cdots. \tag{7}$$

Upon fulfillment of the condition $\lambda_B \tau_{RTO} \leq 1$ it is permissible to disregard the second and third articles of (7), from which it follows that $P_{BUG,\Delta t}(\tau_{RTO})$ for short time intervals τ_{RTO} will be determined by:

$$P_{BUG,\Delta t}(\tau_{RTO}) \cong \lambda_B \tau_{RTO}. \tag{8}$$

During the operation of the SS the time for the real technical operation τ_{RTO} is a part from time for total technical operation τ_{TTO}. Time τ_{TTO} contains the time τ_{OTP} to check the SS through the operation of the testing program (OTP) and the time for real technical operation of SS τ_{RTO}. From this the draws follow:

$$\tau_{TTO} = \tau_{RTO} + \tau_{OTP}, \tag{9}$$

$$\tau_{RTO} = \tau_{TTO} - \tau_{OTP}. \tag{10}$$

From expressions (8), (9) and (10) we have:

$$P_{BUG,\Delta t}(\tau_{RTO}) \cong \lambda_B (\tau_{TTO} - \tau_{OTP}). \tag{11}$$

After a mathematical transformation of (11) the final result necessary for scientific and practical research is obtained:

$$P_{BUG,\Delta t}(\tau_{RTO}) \cong \lambda_B \tau_{TTO} - \lambda_B \tau_{OTP}. \tag{12}$$

Assuming that the study is performed for one quarter of SS operation (necessary condition for the implementation of (5)), i.e. the time for total technical operation τ_{TTO} = const and the relative constancy of bug manifestation intensity λ_B (when using an appropriate debugging program), we come to the scientifically applied conclusion:

The probability of bugs occurrence in one SS is inversely proportional to the running time of the operation of the testing program τ_{OTP} used by software professionals.

The indication of the presence of bugs should not be ignored by specialists, even in the most stressful stages of programming in research and application and decision making.

4. NUMERICAL EXAMPLE

The probability of bugs occurrence $P_{BUG,\Delta t}$ as a function of duration of the operation of the testing program τ_{OTP} is studied for one quarter (92 days) work of the SS (with 12 hours of continuous operation per day), as well as $\tau_{TTO} = 92 \times 12 = 1104$ h and $\lambda_B = 0.16 \times 10^{-6}$ bug/hour.

The result of the research is presented in Fig. 1. For the studies shown in Fig. 1 it is assumed that at the beginning of the setting the duration of operation of the testing program $\tau_{OTP} = 0$. The studied SS has approximately $(10^5 \div 10^6)$ commands and it contains 1% bugs, i.e. the value of the probability of a bug occurrence is $P_{BUG,\Delta t} = 10^{-2}$.

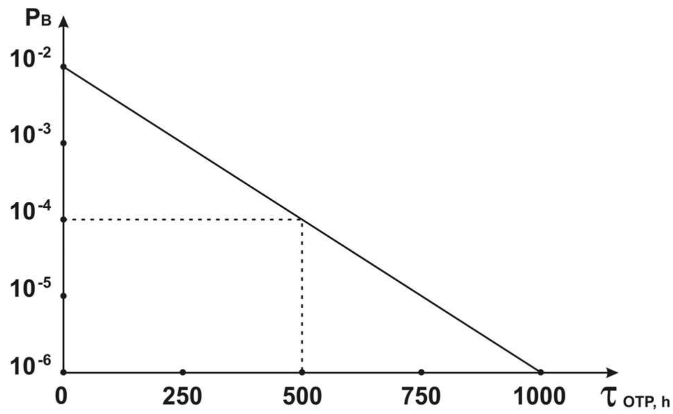

Fig. 1. The probability of a bug occurrence $P_{BUG,\Delta t}$, as a function of τ_{OTP}, measured in hours (h)

Upon entering the SS setup process and value of the duration of operation of the test program $\tau_{OTP} = 1000$ hours, the probability of a bug occurrence in each command decreases by four orders of magnitude and reaches the value $P_{BUG,\Delta t} = 10^{-6}$, i.e. one bug corresponds to 10^6 commands. Attention should be paid to the fact that in the first half of the study $\tau_{OTP} = 500$ hours and the duration of SS testing and setup is analyzed. This practically removes the bulk of the bugs (95%) [5]. The second half of the study analyzed the duration of testing and tuning, leading to the removal of 5% of detected bugs.

5. CONCLUSION

A model of estimating the probability of detecting bugs in small software systems has been proposed. The evaluation performed in this way allows improvement by using multi-criteria decision-making methods [10]. The model is especially relevant when using software systems for training and maintaining information contacts between scientists, lecturers and students.

REFERENCES

[1] Iv. Popchev and Ir. Radeva, Risk Analysis – An Instrument for Technology Selection, *Engineering Sciences* (2019) **LVI** (4) 5–20, DOI: 10.7546/EngSci.LVI.19.04.01

[2] J. Musa, *Software Reliability Engineering*: *More reliable software faster and cheaper*, 2nd Ed., Tata Mc Graw-Hill (2005).

[3] M. Shooman, *Software Engineering*: *Reliability*, *Development and Management*, Ney Jersey, Prentice-Hall (1984) 101, ISBN-10: 0070665575.

[4] M. Shooman and G. Richeson, Reliability of Shuttle Mission Control Center Software, in: Proceedings of Annual Reliability and Maintainability Symposium, New York, 1983, pp. 125–135.

[5] R. Peng and J. Liu, Simulated Software Testing Process Considering Debuggers with Different Detection and Correction Capabilities, *International Journal of Performability Engineering* (2017) **13** (3) 334–336.

[6] N. Petrov, *Probability*, *Independence*, Information Society, Astra, Lodz, Poland (2012) 85–90.

[7] N. Petrov, V. Dimitrov, and V. Dimitrova, Reliability of Technology Systems in Industrial Manufacturing, Publishing House “Akinik Publications” New Delhi, India (2016) 87–99, ISBN: 978-93-87072-59-6.

[8] I. Popchev, Decentralized Stabilization of Linear Continuous-time Systems by Static State Feedback, *Comptes rendus de l’Académie bulgare des Sciences* (1987) **40** (10) 65–68.

[9] I. Popchev, *Decentralized Systems*, Publishing House of the Bulgarian Academy of Sciences (1989) 10, UDK 681.3: 62–52, 278.

[10] V. Peneva and I. Popchev. Fuzzy Multi-Criteria Decision Making Algorithms, *Comptes rendus de l’Académie bulgare des Sciences* (2010) **63** (7) 979–992, ISSN: 1310-1331.

[11] N. Petrov, On the Reliability of Information as a Cybernetic Technico-Economic Problem, *IJSER* (2017) **8** (6) 1629–1632, ISSN: 2229-5518.

[12] N. Petrov, *Reliability of Information as a Problem of Cybernetics in Sea Risk Systems*, The EU, JOP "Black Sea Basin 2013", Romania (2013) 27–38, ISBN: 606-681-069-3.

[13] N. Petrov, *Reliability and Risk as a World Cybernetic Problem*, RL "G. Rakovski", Bulgaria (2017) 50, ISBN 978-619-7319-00-2.

[14] N. Petrov, The Wave Function of Microparticles as a Component of System Reliability, *IJMET* (2019) **10** (4) 830–839, ISSN: 0976-6340 (Print), ISSN: 0976-6359 (Online).

[15] N. Petrov, Reliability's Investigations of Risk Technical Systems, Third Additionally Publication, Publishing House "Zh. Uchkov" (2018) 160, ISBN: 978-954-391-120-2.

[16] V. Dimitrova and V. Dimitrov, Approach for Determining the Quantity of Single Flute End Mill Cutters for High-Speed Milling, Planned Out for Recovering, *International Journal of Innovative Science*, *Engineering & Technology* (2014) **1** (8) 264–266, ISSN: 2348-7968.

[17] N. Wiener, *The Human Use of Human Beings*: *Cybernetics and Society*, London (1988) 30–33, ISBN: 1-85343-075-7.

[18] V. Dimitrov and V. Dimitrova, Technical Resource of Single Flute End Mill Cutters – Prediction and Analysis, *International Journal of Innovative Science, Engineering & Technology* (2014) **1** (8) 261–264, ISSN: 2348–7968.

[19] N. Kolev, Wireless Data Transmitter from GSM Cell, *Trakia Journal of Sciences* (2010) **8** (Suppl. 3) 33–36, ISSN 1313-7069.

[20] R. Yankov, M. Ivanova, V. Dimitrova, and V. Dimitrov, Possibility of Evaluation of Insulation and Accumulating Ability of the Room on the Data of the Transition Process after Turning off the Heat, *International Journal of Scientific & Engineering Research* (2018) **9** (4) 196–198.

[21] N. Petrov, K. Dimitrova, and S. Ratchev, Determination of the Optimal Circuit-engineering Solution of Electronic Circuits, in: XXIst International Symposium on Electrical Apparatus and Technologies SIELA 2020, Bourgas, Bulgaria, 2020.

[22] K. Dimitrova, Variant Solution on the Transistor Protective Stage, *Journal of the Faculty of Engineering and Pedagogy – Sliven*, *Technical University* of Sofia (2014) **3**, ISSN 1312-3920.

Received June 02, 2020