Tofig H. Kazimov [1]***, Tamilla A. Bayramova*** [2]
Institute of Information Technology of ANAS, Baku, Azerbaijan
[1]tofig@mail.ru, [2]tamilla@iit.ab.az

DOI: 10.25045/jpit.v08.i1.12

COMPARATIVE ANALYSIS OF AVAILABLE MODELS FOR ESTIMATION OF SOFTWARE RELIABILITY

This article provides information about one of the main signs of software – reliability and models which are used for its estimation. Some of mathematical models that is applied in estimation of reliability of software tools have been studies.

Keywords: *reliability, analytical model, dynamic model, empirical model, software engineering, failure rate, degree of reliability.*

Introduction

The experience of creation and implementation of complex information systems has shown that, in most cases, the rejections occurring during the operation are related to the defects of facilities, and this paves way to damages. One of the main problems currently facing the software engineering is to increase the quality of software.

The ISO 9126: 1991standard [1] defines six characteristics and sub-characteristics of the quality. Reliability is shown here as one of the main characteristics of the quality of software. Reliability is an important factor that determines the performance and efficiency of software. Therefore, special attention should be paid to the reliability issues during the designing process. Ensuring the reliability of the entire lifecycle of software by experts should be carried out using methodology, technological instruments, standards and regulatory documents. To guarantee the reliability of software, the development and implementation of effective methods and tools is very essential, which notify about defects and prevent them enabling their performance during the emergence of these defects.

Whilst designing the interdependence between costs and work schedule of the system, it is important to accurately form the requirements for the reliability of the system. Hence, they exactly affect the cost of the project. The cost and reliability of software is closely linked with each other. Thus, design, testing and support requirements are higher when a software is developed with high reliability, which also results in an increase in their prices.

Software reliability–software product ability to perform certain functions with high accuracy in certain circumstances and period of time [2].

Some key concepts of the software used for the SE evaluation are as follows [3]:

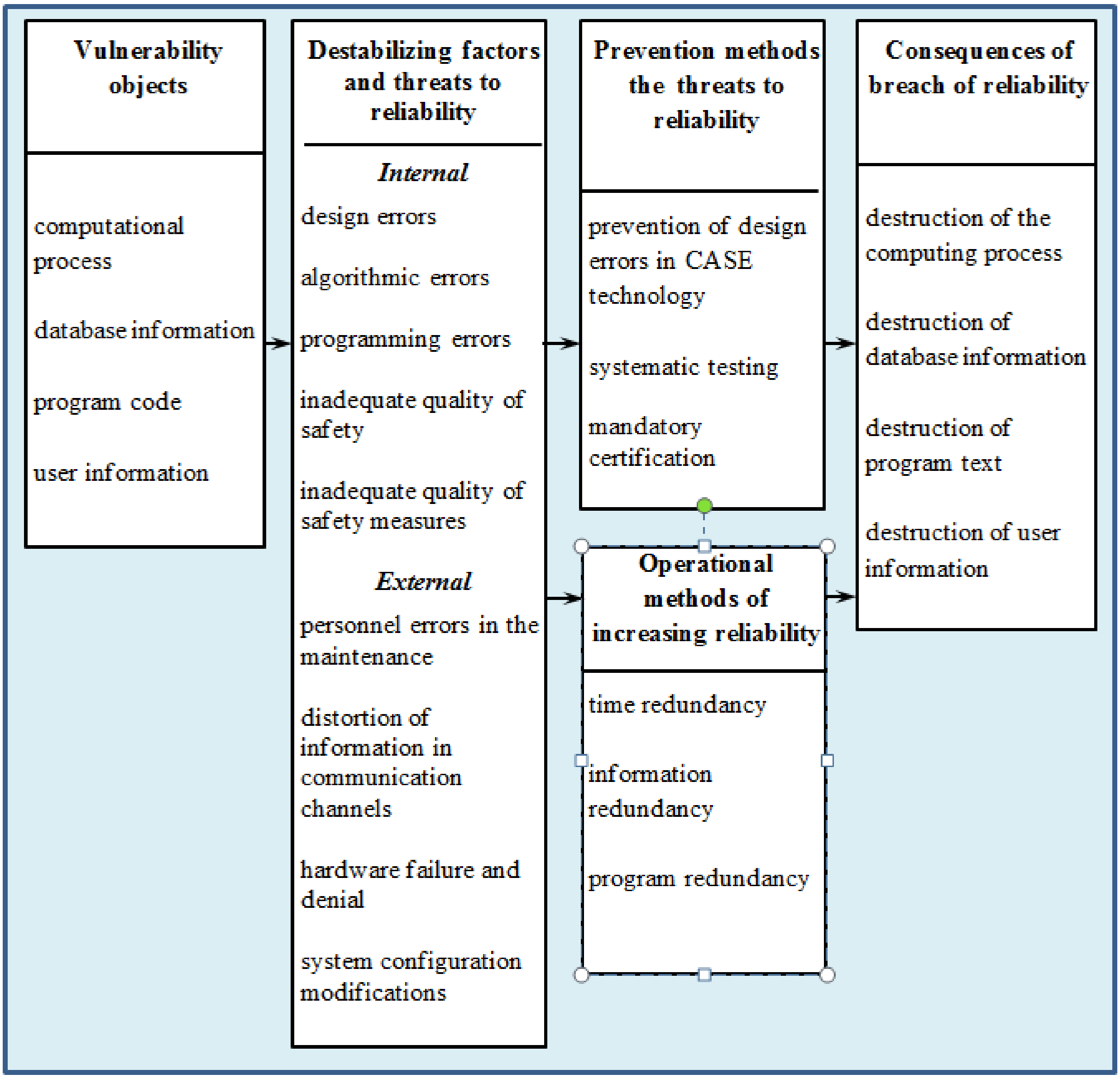
- software denial (unsustainably avoidance of software operating characteristics from the specified requirements);
- reliability rate (characterized by the probability of working without denials in a certain period of time);
- probability of software denial;
- software denial intensity;
- software frequency (preserving performance the working ability during data processing);
- software robustness (limiting own defects and the negative effects of the external environment or durability against them).

Errors in software without incidental effects are random. They are distinguished according to their feature and outcome. In [4] it is presented a model showing the relationship between the key components that affect the reliability of software (Figure 1).

There are two approaches to the evaluation of software reliability: qualitative and quantitative. The qualitative approach is based on a system of requirements that are defined by standards and institutional regulatory documents. These requirements are verified when evaluating the reliability of software. In the quantitative approach, the reliability of software is estimated based on different mathematical models [5].

The reliability models are used to assess the quantitative indicators of the software reliability. The reliability models are the mathematical models established to estimate the dependence of parameters, which are known in advance or identified during the problem solution. The software reliability can be calculated through these models at various stages of its lifecycle.

Figure 1. Model showing the relationship between the key components that affect the



reliability of software

Currently, various mathematical models and their modifications are available to define the reliability of software. Each of these models sets the parameters arisen from the software development, testing and supporting stages. Denials and defects detecting period are the key factors. However, the models based on the parameters as the complexity of software systems, operating environments and input data are also available. As the software reliability models are various, the approaches to their classification also differ [6].

In [7, 8], the most common classification of the reliability models is shown (Figure 2).

The empirical models are based on the analysis of the structural characteristics of the programs. There is a close connection between the software complexity and its credibility. Complexity means the software size (number of application modules), the number of operators, the number of subsystems and inter-module interfaces, complexity, and so on. The advantage of the empirical models is easy calculations and the absent of complex formulas. The disadvantage of the model is the fact that the calculations are approximate and does not reflect the dynamics of the processes during the operation.

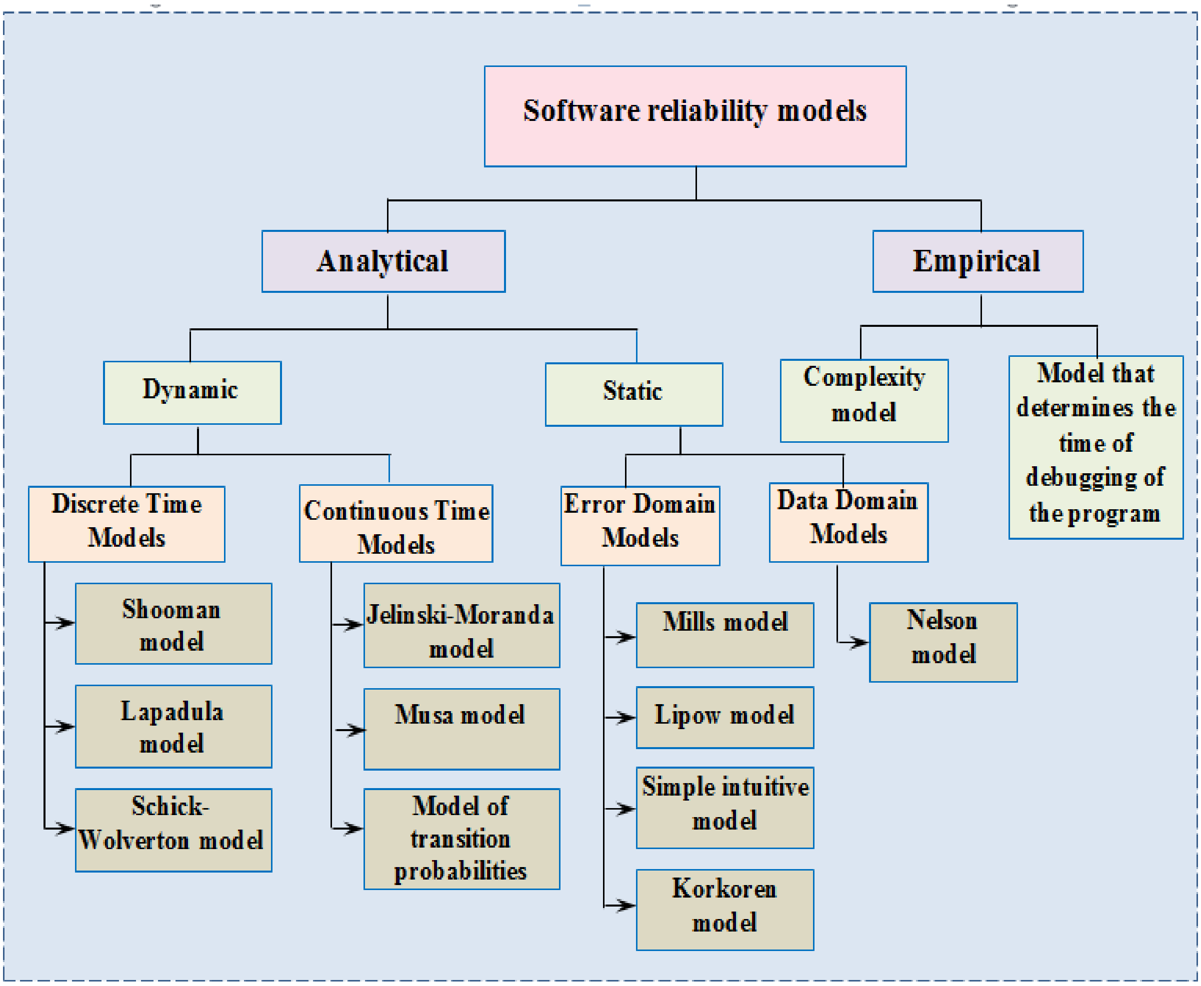


Figure 2. Classification of software reliability models

Analytical models are based on the data obtained during the trials, and enable to calculate quantitative indicators of the reliability. Analytical models are divided into two groups, namely dynamic and static models. Dynamic models consider the dependence of denials on the time. Static models differ from the dynamic models for the fact that the defect times are not taken into consideration. Here, the dependence of denials on the number of either tests or input data is taken into account.

Figure 3 illustrates the reliability models for Goel classification [9].

The models that do not take into account the number of errors are based on the estimation of the intervals between the software denials and predict the number of remaining errors in the software. After each denial, the reliability is assessed and the period up to the next denial is calculated [10].

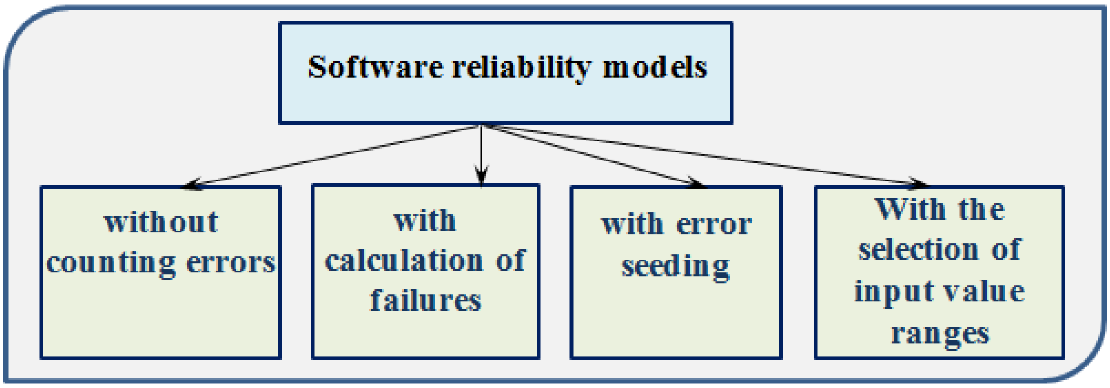


Figure 3. Reliability models for Goel classification

The models estimated based on the number of denials are derived from the number of errors discovered during the given period. The failure of the software depending on the time is considered a stochastic process, whereas the number of errors is a random quantity.

The models estimated by generating imitated errors are based on the number of corrected errors in the software code and on the addition of imitated errors (type and number of which are known beforehand). The ration of the number of predicted undetected errors in the program code to the number of undetected imitated errors is achieved and compared to the ratio of the number of detected errors to the number of detected imitated errors. The result of this comparison is used to evaluate the reliability of software.

The models calculated based on the field of values of input data, are generating random input data, and the reliability is assessed based on the results of these test data.

In addition to the above mentioned classification, Hatch, Polonnikov-Nikandrov, Fatuyev and other classifications are also available [11, 12].

Due to the development of the software industry, the classification scheme of the reliability models has been significantly expanded and a new classification technique has been developed. In [13], common factors used in the classification of software reliability models are summarized (see Table 1 and Figure 4).

Table 1

Common Factors Used to Classify Reliability Models

Classifications features	Used parameters
Time structure	– error detection period – period between error detection – number of errors during given time
Software structure (complexity)	– length of the program – volume of the program – number of functions and data
Detecting errors	Software is added a known number of errors
Structure of input data field	This data is tested several times. Most of the total number of trials shall be successful
Structure of software text	Error distribution in software text (which part of text has more errors)

The reliability models shown in this article enable formulating an idea about the software product by being applied during the trials at different stages of the life cycle of software. For example, the overwhelming majority of mistakes detected and corrected in the trial stage suggests that the product test is about to end, since the number of remaining errors is minimal. In fact, this may not be true. In this case, the application of software reliability models can simplify the issue.

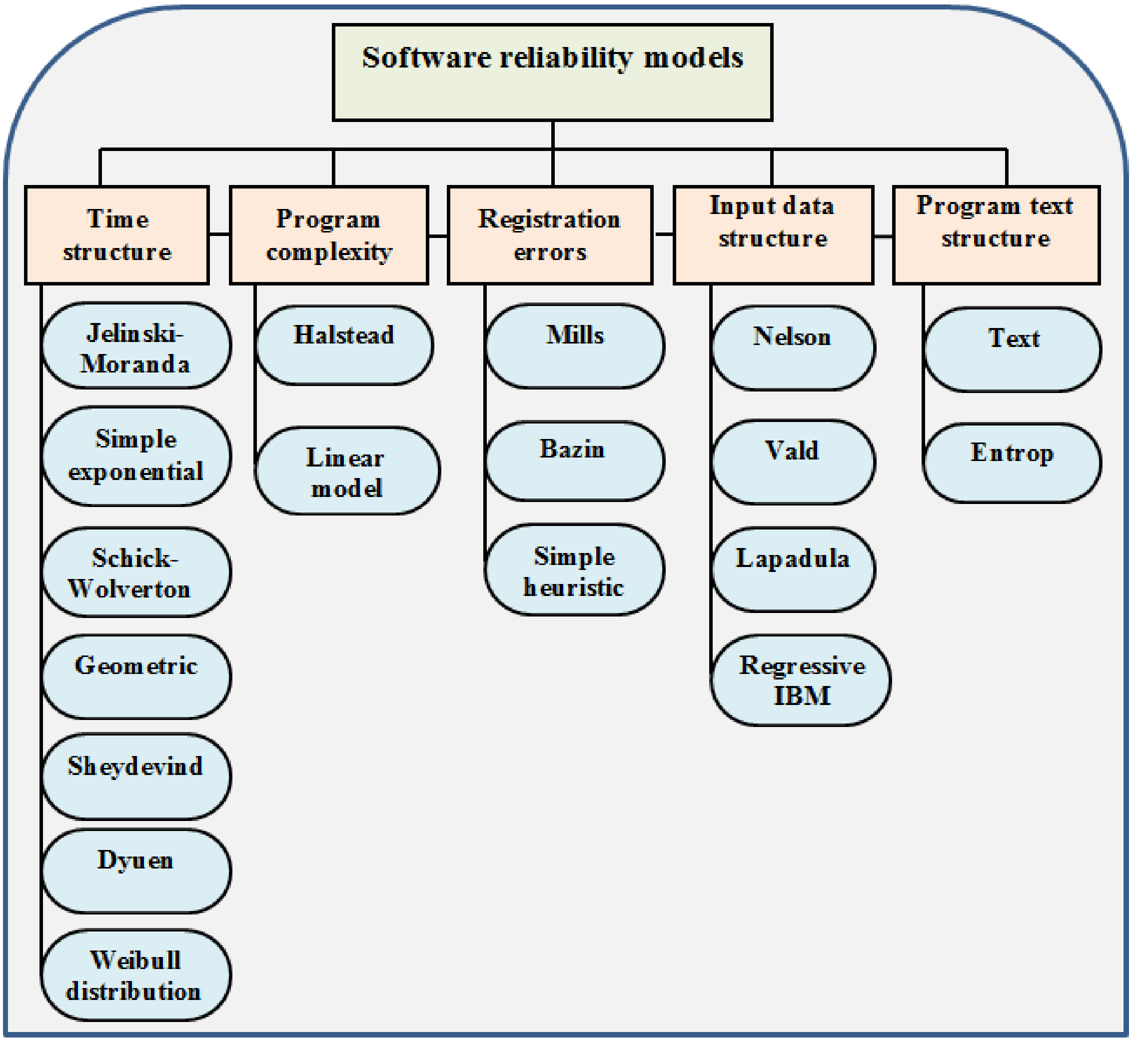


Figure 4. Classification of reliability models by common factors

Given that uneven distribution of errors and that software failure is an unstable process, it is almost impossible to calculate the reliability based on these classic models. In this case, quantitative and qualitative analysis methods are used. These methods may include *Fault Tree Analysis (FTA), Reliability Block Diagram (RBD), Failure Mode and Effects Critical Analysis (FME(C)A), Failure Mode Effect and Diagnostic Analysis (FMEDA)* [14].

Conclusion

Reliability issues are essential when the errors during software designing and managing are not detected on time and can cause serious problems for human life and the environment.

Various mathematical models have been developed to assess the reliability of software, which enable calculation of the reliability characteristics (performance time until denial, preparation coefficient, denial probability etc.). It should be noted that there is no completely reliable model. Consequently, the absolute value of the reliability can not be proved, thus, detection is also impossible. The above-mentioned models are mainly theoretical approaches and some restrictions arise from the application. These models do not take into account that new errors can be added to the software code when correcting existing errors. There is no such a model that the application of which is not limited. For this reason, the development of more advanced methods for correctly identifying, evaluating and predicting factors affecting the reliability of software, is one of the topical issues.

In recent years, the following main areas to assess the reliability of software are predominant:

- analysis of defect origins [15];
- calculation of the time of denial occurrence taking into account the hardware of distribution function (backup opportunity and time, the probability of unrecoverable denials etc.), [16, 17];
- software development for the prediction of denials [18];
- predicting defects by using genetic programming and time series [19, 20];
- software reliability modeling using genetic algorithms [21].

References

1. ISO 9126: 1991. Information technology. Software evaluation. Quality characteristics and guidance on their use, p. 186.
2. IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Technology (ANSI), 1283 p.
3. Myers G. Software Reliability. M .: Mir, 1980, 360 p.
4. Lipaev V.V. Software reliability. M.:Sinteg, 1998, 232 p.
5. About metric approach to evaluation of the quality and reliability of software, www.hups.mil.gov.ua/periodic-app/article/14636/soi_2002_6_60.pdf
6. Sukert A.N., Goel L.A. A guidebook for software reliability assessment / Proc. Annual Reliability and Maintainability Symp. Tokyo (Japan), 1980, pp.186–190.
7. Classification of software reliability models, https://studfiles.net/preview/6179493/
8. Blagodatskikh V.A., Volnin V.A., Poskakalov K.F. Standardization of software development: Tutorial. M.:Finance and statistics, 2005, 288 p.
9. Goel A. L. Software reliability models: Assumptions, Limitations and Applicability// IEEE Transactions on Software Engineering, vol. SE-11, no.12, 1985, pp.1411–1423.
10. Jelinski Z., Moranda P.B. Software Reliability Research, Statistical Computer Performance EvaluationW.Freiberger, Academic Press, New York, 1972, pp.465-484.
11. Polonnikov R. İ., Nikandrov A.V. Methods for evaluation of the reliability indicators of software. SPb.:Politechnics, 1992, 78 p.
12. Fatuev V. P. Reliability of automated information systems: Tutorial // V. P. Fatuev, V.İ.Vysotsky, V.İ.Bushinsky, Tula: TSU, 1998, 104 p.
13. http://konf.x-pdf.ru/18tehnicheskie/416395-1-metodi-ocenivaniya-nadezhnosti-programmnihsredstv-uchetom-vtorichnih-defektov.php
14. Kharchenko V.S., Sklyar V.V., Tarasyuk O.M. Methods for modeling and evaluating the quality and reliability of software. Kharkov: National Aerospace University «Kharkov Aircraft Institute», 2004, 188 p.
15. Kumaresh S., Ramachandran B. Defect Prevention Based on 5 Dimensions of Defect Origin // International Journal of Software Engineering & Applications (IJSEA), 2012, no.3, pp.87–98.
16. Markov A.S. Models of evaluation and planning of software testing for information security requirements // Bulletin of MSTU, «Priborostroenie», 2011, p.90–103.
17. Mirtskhulava L., Khunjgurua M., Lomineishvili N., Bakuria K. Software Reliability Prediction Model Analysis // International Journal of Computer, Information, Systems and Control Engineering, 2014, no.6, pp.927–932.

18. Al-Rahamneh Z., Reyalat M., Sheta A.F., Bani-Ahmad S. A New Software Reliability Growth Model: Genetic-Programming-Based Approach // Journal of Software Engineering and Applications, 2011, no.4, pp.476–481.
19. Tsakonas A., Dounias G. Predicting Defects in Software Using Grammar-Guided Genetic Programming // Proceedings of the 5th Hellenic conference on Artificial Intelligence, SETN, Syros, 2008, pp.413–418.
20. Raja U., Hale J.E., Hale D.P. Temporal Patterns of Software Evolution Defects: A Comparative Analysis of Open Source and Closed Source Projects // Journal of Software Engineering and Applications, 2011, no.4, pp.497–511.
21. Silvia R.V., Aurora P. A grammar-guided Genetic Programming framework configured for data mining and software testing // International Journal of Software Engineering and Knowledge Engineering, 2006, no.2, pp.245–267.