Providing Functional Safety of Hardware-Software Complexes as Mandatory Condition of Forming Digital Eco-Environment at Innovative Project Implementation

Vladimir Gvozdev
dept. Computer Science and Robotics
Ufa State Aviation Technical University
Ufa, Russia
wega55@mail.ru

Alexander Levkov
dept. Computer Science and Robotics
Ufa State Aviation Technical University
Ufa, Russia
projekctor@gmail.com

Darya Blinova
dept. Computer Science and Robotics
Ufa State Aviation Technical University
Ufa, Russia
blinova.darya@gmail.com

Nicholas Rovneyko
dept. Computer Science and Robotics
Ufa State Aviation Technical University
Ufa, Russia
nikolaos@mail.ru

Aliya Davlieva
dept. Computer Science and Robotics
Ufa State Aviation Technical University
Ufa, Russia
aliyasr21@gmail.com

***Abstract*—A key part digital eco-environment plays at the implementation of innovative projects brings about qualitative changes of requirements towards the functional safety of hardware-software complexes. The paper dwells upon the issues of increasing the functional safety of hardware-software complexes by means of reasonable resource allocation for locating and eliminating defects. The purpose of the research is the development of the information support of planning resources for defect elimination in software modules. The scientific idea is the development of the method of building linear regression dependencies on the basis of common use of expert assessments and measuring data. The approach novelty consists in the transformation of expert assessments and measuring data towards a single form of the random value distribution law.**

Keywords—hardware-software complex, innovative project, defect, time for defect elimination, expert assessments

I. INTRODUCTION

As of today there has been much talk on the 4th industrial revolution, a so-called "Industry 4.0" uniting physical, biological and digital systems.

One of the basic provisions of the "Industry 4.0" is forming information environment for all participants of design, maintenance, upgrade. In this connection one can observe increase in the requirements towards reliability, vulnerability reduction and provision of information system survivability which demands developing the theoretical basis to resolve these tasks.

The purpose of the digital eco-environment is, firstly, providing information for flexible reconfiguration of business applications in compliance with the change of requirements and preferences the product consumers express. Secondly, it is aimed at timely distribution of new data, obtained as a result of practical activities, among the actors. A mandatory requirement towards the properties of digital eco-environment is the representation of data and information to various actors in the form corresponding to the peculiarities of the tasks being resolved [1,2]. In view of the above, one can draw a conclusion that the implementation of modern innovative projects consists in timely provision of the actors with reliable information services implemented with the help of hardware-software complexes that provide the conditions for the efficient solution of business tasks.

An important task directly connected with the management of information service functional safety is the task of managing the functional safety of software systems implementing the services. The analysis of references dedicated to provision of software system functional safety allows suggesting the following classification of approaches: the first category is oriented towards the elimination of defects in software product structure. The second category is oriented at the assessment of compliance of consumptive properties of hardware-software complexes with the user requirements (functional and non-functional). It should be highlighted that these approaches are "two sides of the same coin" and oriented towards achieving the same level of quality which is defined in the statement of work on the hardware-software complex development.

Functional safety is significantly defined by software products' reliability which, in its turn, defined by the number of defects. The number of unrevealed defects is defined by the quality of planning the quality of resources necessary for defect elimination.

Managing the functional safety of hardware-software complexes in the framework of the aforementioned approaches based upon the use of descriptive mathematical models. As the examples oriented towards defect elimination one can provide the models of "software reliability growth models" and "detect density models".

One of the tasks within the framework of managing functional safety is resource planning, generating hardware-software complexes necessary, in particular, for the search and elimination of defects in hardware-software complexes [3,4].

Defect elimination stipulates for the solution of the following information tasks:

a) establishing the fact of defect elimination on the basis of analyzing defect manifestation symptoms (deviation of the item behavior from the reference one);

b) locating the defect position, taking decision on the practicability of its elimination with the account of the aftermath of item behavior deviation from the reference one;

c) defect elimination and building a system preventing errors which lead to the occurrence of such defects in future.

The article is dedicated to the development of methodological support for planning the scope of resources for eliminating defects in software components. The purpose of the research is the development of the information support of planning resources for defect elimination in software modules. The scientific idea is the development of the method of building linear regression dependencies on the basis of common use of expert assessments and measuring data. The approach novelty consists in the transformation of expert assessments and measuring data towards a single form of the random value distribution law.

II. Analysis of Approaches Towards the Assessment of the Number of Defects in a Software Component

The content of models relating to the group of reliability growth models is making up empirical dependencies of defect manifestation in software products in relation to the resources spent for software test and debugging. The content of models associated with the group of defect density models is building empirical dependencies of defect number on structural peculiarities of hardware-software complexes [5].

The limits of the mentioned models are:

- Orientation towards the search and elimination of defects which are the main causes of software product failures without establishing defect root causes;
- Empirical model character; absence of the generality property in these models; low model potential.

On the basis of the conducted analysis of references one can draw a conclusion that by today a range of methods for analyzing software systems reliability has been developed. These methods are applied at high level design, detail design and coding stages. The sources of the developed methods are the methods used to analysed the reliability of technical systems oriented towards the processing of measuring data.

There is a necessity of developing existing analytical means for the software system reliability analysis taking into account the peculiarities of self-perception of software component properties of hardware-software complexes by various users.

III. Evaluating Resources for Defect Search and Elimination

Defect management is a rather complex problem. One of the tasks is the system of defect taxonomy [6,7]. Different authors suggest various sets of “good taxonomy” indicators. Thus, Chambers suggests using completeness, accuracy, reliability and ease of use indicators [8]. Baker & Krokos suggest using internal validity, reliability and functional utility as such criteria [9]. Liu defines the classification system quality as a share of misclassified objects [10]. The research does not cover the problem of defect classification as it is a separate complex task.

In [11] is points to the fact that defect analysis should start from identifying defect classes. The aforementioned paper provides a rationale for the assumption that defect classification is the basis for common use of defect data obtained in various organizations. However, this paper does not discuss important issues connected, firstly, with lawful aspects of data submission and, secondly, with the provision of data comparability as reference data are obtained in various conditions (it is preconditioned by various organization of software product consumers). One can suggest that as of today there are significant implications for these processes, which is confirmed by the generation of a non-commercial organization Standish Group [12].

The references explain that software products should be developed and tested by various groups of developers [13]. The paper [14] states that most companies pay insufficient attention to documenting the software product development and test results. Due to this, the following situation is quite common: the results of development and tests are registered on separate occasions while the amount of data on the defects revealed and the complexity characteristics are different.

Defect elimination to the acceptable level requires planning the scope of resources for conducting object (the term “object” is interpreted as per GOST R 51901.5-2005.) tests, defect locating and elimination at the stage of innovative project initiation.

Resource scope planning for test conducting is based upon the prediction of defect number at the stage of software implementation. The predictions of defect number are based upon historical data on previously detected defects in similar projects.

The exclusion principle stated by Lotfi Zadeh is the following: high accuracy of describing some system excludes its high complexity [15]. At small number of data using high-order models is not practical while the number of single-type defects in one organization, as a rule, is quite a limited value.

The papers [16,17] state the provision on the presence of linear dependence between the indicators of software system component complexity and defect number. Historical data archives provide the basis for establishing functional dependencies between the indicators of software complexity and the identified defect number at the test stage. The requirement specifications provide the basis for building the tests with the help of black and white box methods (the basis of test building with the help of a white box method is an algorithm flow chart).

At present there is an approach used for the assessment of defect number on the basis of the HSC component complexity characteristics. As software components development and testing are the tasks of different specialists, the scope of reference data on complexity characteristics and a number of revealed defects is versatile.

IV. FORMAL PROBLEM STATEMENT

Software product development and defect identification are the tasks for various specialist groups [18,19]. This leads to the impossibility of forming a correlation table. In addition, the volume of data characterizing the complexity and time of defect elimination are definitely different as various types of defects can be revealed in one software [20]. However, the defects belonging to one class can be considered to be homogeneous in statistical terms as they are obtained in one organization. The aforementioned facts explain the necessity to develop a method for the information support of planning resources for defect elimination in software modules.

The reference data of the task of building linear regression dependencies on the basis of common use of expert assessments and measuring data are provided below. It should be mentioned that, taking into account multiple defect classes, each class will correspond to its own regression model.

Given:

a) Multiple expert evaluations of the complexity of the previously developed software products $\{c_1,..,c_N\}$.

b) Time spent for defect elimination $\{\tau_1,..,\tau_M\}$. The values $\{\tau\}_1^M$ are measuring.

c) "Development complexity" is an integral characteristics of a software product. It is defined not only by formal characteristics of a product (algorithm structural properties; implementation tool, etc.) but also by intuitively assessed properties (communicative skills of a customer representative; service provider motivation, etc.). The complexity is a subjective expert evaluation. The experts are key service providers responsible for the development of software system subsystems' development. Be specific, it is assumed that $c \in [0;10]$ is a numerical score.

d) Data losses are possible, both the data on complexity assessment and time spent for defect elimination, i.e. the sample scopes for $\{c\}_1^N$ and $\{\tau\}_1^M$ are different.

e) The data homogeneity is stated due to:

- External environment stability (the same customer; the tasks belong to one subject area). At the same time, at the implementation of various software products one should contact with different representatives of a customer.
- However, the group of developers is quite stable.

The unimodality of distribution laws $f(c)$ and $f(\tau)$ follows from the statement on data homogeneity.

Solution:

To obtain linear model parameters the following method is suggested [21,22]:

Step 1. On the basis of the sample $\{c\}_1^N$ the authors define the distribution function $F(c)$. In a similar way, by the sample $\{\tau\}_1^M$ the distribution function $F(\tau)$ is defined.

Step 2. There is a direct problem of searching for the random value distribution law on the basis of the known correlation [23]:

$$F(y) = \int_{\varphi(x)<y} f(x)dx . \quad (1)$$

By means of resolving a reverse problem of defining the function of random argument [24] the authors obtain the dependence of defect elimination time on the complexity $\varphi(c)$:

$$A: \{F(c), F(\tau)\} \to \tau^* = \varphi(c) . \quad (2)$$

Step 3. By solving the problem presented in Fig. 1:

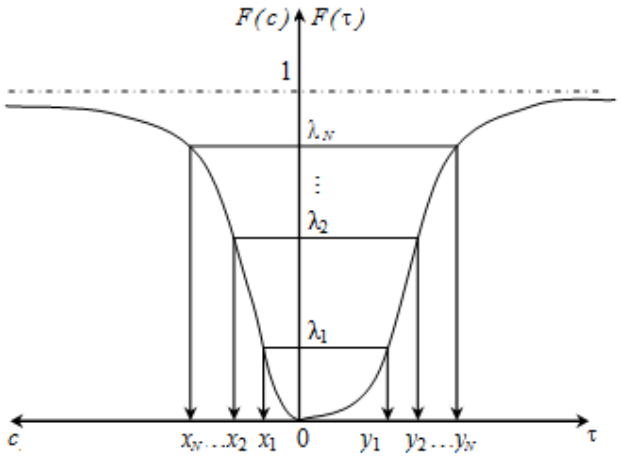

Fig. 1. Reverse problem of defining the function of random argument

The authors obtain the values $\xi = F^{-1}(\lambda_i)$, $(i = \overline{1;N})$ where $\lambda_i \in [0;1]$ – a uniformly distributed random value while ξ takes the values of x and y, a set of comparable $\{x\}_1^N$ and $\{y\}_1^N$ is formed making up a table of commonly observed values. From which define the correlation factor $r_{c\tau}$.

A graphic illustration of the diagram is given in Fig. 2.

The proposed method stipulates for the implementation of the following method.

a) Providing for compatibility of data required for defect elimination:

$$\tau_j^* = \frac{\tau_j - \tau_{\min}}{\tau_{\max} - \tau_{\min}} \cdot 10 . \quad (3)$$

In this case $\tau^* \in [0;10]$.

b) On the basis of $\{c\}_1^N$ calculate the mathematical expectation of complexity $M[c]$.

c) On the basis of $\{\tau^*\}_1^M$ calculate the defect elimination time $M[\tau^*]$.

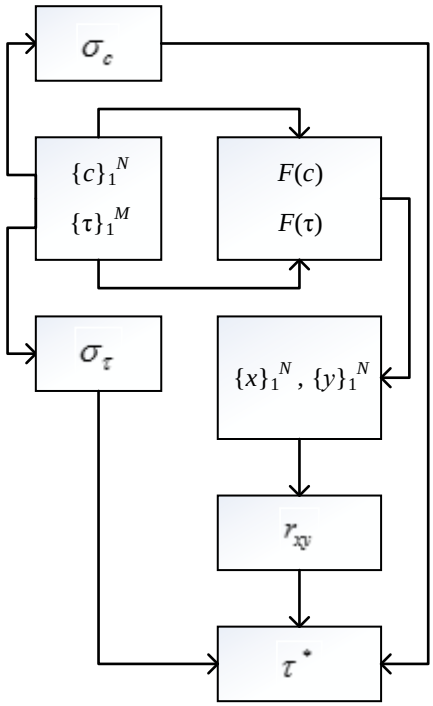

Fig. 2. Graphic illustration of the diagram for linear model defining

d) State a linear character of dependence between *c* and τ ($\tau^* = \varphi(c)$ linear), i.e. $\tau^* = a + bc$, where linear dependence parameters of *a* and *b* are defined as [20]:

$$a = M[\tau^*] - M[c] \cdot \frac{\sigma_\tau}{\sigma_c}, b = r_{c\tau} \frac{\sigma_\tau}{\sigma_c}, \quad (4)$$

where σ_c – mean square complexity deviation, σ_τ – mean square deviation of defect elimination time.

Here $\sigma_c = \sqrt{\int_0^{10} c^2 f(c)dc - (\int_0^{10} cf(c)dc)^2}$, $\sigma_\tau = \sqrt{\int_0^{10} \tau^{*2} f(\tau^*)d\tau^* - (\int_0^{10} \tau^* f(\tau^*)d\tau^*)^2}$, $r_{c\tau}$ – correlation factor.

e) Calculation of normalized time on defect elimination.

- The expert defines complexity in points as c_k.
- From the regression equation: $\tau_k^* = a + bc_k$.

f) Calculation of expected time for defect elimination:

$\tau_k^* = \frac{\tau_k - \tau_{\min}}{\tau_{\max} - \tau_{\min}} \cdot 10$, from which it follows

$\tau_k = \frac{\tau_k^* (\tau_{\max} - \tau_{\min})}{10} + \tau_{\min}$.

Method limitations:

The value of τ_k should be located $\tau_k \in [\tau_{\min}; \tau_{\max}]$ where $\tau_{\min} = \min_j \{\tau_j\}_1^M; \tau_{\max} = \max_j \{\tau_j\}_1^M$.

This is explained by the fact that statistical evaluations are relevant only in the area of actually observed random values.

V. Conclusion

Therefore, the authors managed to develop an approach united to search for the defects of various classes. The developed method for information support of resource planning for defect elimination in hardware-software complex modules is built upon building linear regression dependences on the basis of common use of expert evaluations and measuring data. The approach novelty consists in the transformation of expert assessments and measuring data towards a single form of the random value distribution law. The suggested method allows developing a formalized procedure for calculating the expected time for defect elimination which makes it possible to implement a software tool on its basis within the framework of the automated design systems of software products.

Acknowledgment

The research is supported by the grant of the Russian Foundation for Basic Research № 18-00-00238 “Decision support methods and models for innovative project management based on knowledge engineering”.

References

[1] Schwab K. Fourth Industrial Revolution: Monograph. M.: Publishing house "E", 2017. 208 p.

[2] Günther Schuh, Reiner Anderl,. Jürgen Gausemeier, Michael ten Hompel. Industrie 4.0. Maturity Index. Managing the Digital Transformation of Companies. Acatech_STUDY // www.infosys.com

[3] Lipaev V.V. Functional software security. M.: SINTEG, 2004. 348 p. (in Russian)

[4] Druzhinin G.V. Reliability of automated systems. M.: "Energy", 1977. 536 p. (in Russian)

[5] P.Bellini, “Comparing fault-proneness estimation models” // Proc. of 10th IEEE International Conference on Engineering of Complex Computer Systems. 2005. P. 205-214.

[6] Fuqun Huang. Human Error Analysis in Software Engineering, Theory and Application on Cognitive Factors and Risk Management - New Trends and Procedures, Fabio De Felice and Antonella Petrillo, IntechOpen, 2017. DOI: 10.5772/intechopen.68392/.

[7] Vaibhav Anu, Wenhua Hu, Jeffrey C Carver, Gursimran S Walia, Gary Bradshaw. "Development of a Human Error Taxonomy for Software Requirements: A Systematic Literature Review"// Information and Software Technology 103, 2008. P. 112-124.

[8] Douglas A., Wiegmann and Esa Rantanen. Defining the relationship between human error classes and technology intervention strategies // ARL-02-1/NASA-02-1.

[9] David P.Baker & Kelley J. Krokos. Development and validation of aviation causal contributor’s error reporting systems // Human Factors, 49(2), 2007. P.185-199.

[10] Yang Liu. The evaluation of classification models for credit scoring // Arbeitsbericht Nr. 02/2002. Institute für Wirtschaftsinformatik.

[11] Rawat M.S., Dubey S.K. Software Defect Prediction Models for Quality Improvement: A Literature Study/ IJCSI International Journal of Computer Science Issues. 2012. Vol. 9, Issue 5, № 2. P. 288-296.

[12] http://www.standishgroup.com/about

[13] Myers G. J. Reliability Software. – M.: MIR, 1980. 359 p. (in Russian)

[14] McConnell S. How much does a software project cost? – M.: St. Petersburg: Peter, 2007. 297 p.

[15] Lotfi A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. – M.: Znaniy. 1974. P. 5-49. (in Russian)

[16] Morozov A., Janschek K., Yusupova N. On the influence if control from properties to software error location // Proceedings of the International Workshop Innovation of Information Technologies, Dresden, Germany, 2010. (in Russian)

[17] Michael R. Lyu. Handbook of Software Reliability Engineering, Volume 1. Front Cover. IEEE Computer Society Press, 1996 - Computers – 850 p.

[18] Fuqun Huang, Binm Liu. Software defect prevention based on human error theories// Chinese Journal of Aeronautics. 2017, 30(3), pp. 1054-1070.

[19] Lagit Kumar Singh, Anil Kumar Tripathi, Gapika Vinod. Software Reliability Early Prediction in Architectural Design Phase: Overview and Limitations // Journal of Software Engineering and Applications, 2011, 4. P. 181-186.

[20] Gvozdev V.E., Chernyakhovskaya L.R., Davlieva A.S. Decision support in management of hardware-software complex functional safety on the basis of ontological engineering // International Russian Automation Conference, RusAutoCon, 2018. P.1-5.

[21] Gvozdev V.E., Subhangulova A.S., Bezhaeva O.Ya. Linear correlations estimation of technical objects parameters without correlation table of empirical data // Vestnik UGATU, Ufa, 2015. Vol.19. No. 4(70). P. 106-117. (in Russian)

[22] Gvozdev V.E., Bezhaeva O.Y., Subhangulova A.S. Analysis of linear relations objects random parameters on the basis of measurement data / Proceedings of the 16th Workshop on Computer Science and Information Technologies/ Sheffield, England. 2014. Vol. 2. P. 13-15.

[23] Pugachev V.S. Theory of probability and mathematical statistics. - 2nd ed. - M. Fizmatlit, 2002. 496 p. (in Russian)

[24] Ventsel E.S. “Operation research. Objectives, principles, methodology” M.: Nauka, 1980, 208 p. (in Russian)