UDC 004.891.3: 004.3

Govorushchenko T.O., candidate of technical sciences

EVALUATON OF SOFTWARE HIDDEN MISTAKES IDENTIFICATION RELIABILITY

Khmelnitsky National University

Conception of software testing process reliability increasing and category neuronet model of software testing process reliability increasing at the expense of software hidden mistakes identification is described. Technique of evaluation of software hidden mistakes identification reliability on base proposed model is adduced.

Preamble. Software testing with the purpose identification of mistakes, essence of which consists in program and its specification discrepancies, is conducted at software developing stage and at debugging stage. But as a result of objective (software testing technique imperfection, tests incompleteness, funds and time insufficiency) and subjective (insufficient qualification of software developers and testers) factors after software testing at developing and debugging stage in program mistakes is remained. These mistakes are related to hidden mistakes.

Hidden mistakes differes from found mistakes, what hidden mistakes on specified time after software testing at developing and debugging stage exists and still not founds. At that mistakes, conditional of defects, somehow influences on system by way of software.

Hidden mistakes identification is conducted after software developing and debugging stage, on which software testing was partial technological operation, is served as separate technological process and is called repeated software testing.

Target setting. For software testing process reliability and software quality increasing are used not only defect software testing at developing and debugging stage, but and repeated testing with the purpose of software hidden mistakes identification after basic testing. It is confirmed the software reliability and quality depends on quantity of found mistakes including hidden mistakes.

The principal direction of presented research is techniques and means of software testing process reliability increasing at the expense of software hidden mistakes identification at repeated software testing.

Repeated software testing basic tasks:

1) to find a maximum amount of hidden mistakes with view of their further removal. The hidden mistakes will be founded using known testing methods;

2) as possible not to make new hidden mistakes during the time of repeated testing, and in case they occurs have possibility for their removal.

Conception of software testing process reliability increasing. Essence of hidden mistakes identification technique consists in consistent certain mistakes identification with the following removal of theirs beginnings motive. In case of identification mistakes of some type, system finds the mistakes of either type. Hidden mistakes identification is started with most widespread and important mistakes types.

In literature [1] software mistakes classification of priorities and categories is known. Mistakes classification of priorities is realized in accordance with norm of mistakes seriousness level determination.

This approach was specified in respect of software hidden mistakes [2]. All of the hidden mistakes we will split up for appearances on insignificant (I), moderate (M), serious (S) and catastrophic (C).

By Insignificant (I) *hidden mistakes* will think such, that do not influence on user actions, program product with their presence suitable for use.

By Moderate (M) *hidden mistakes* will think such, that influence on user action. But program product with their presence will be suitable for use.

By Serious (S) *hidden mistakes* will think such, that bring false results, by reason of what program product unapt for use.

By Catastrophic (C) *hidden mistakes* will think such, that bring to disfigurement

over information (data), by reason of what program product unapt for use and results of its work brings about refusal of computer system.

Insignificant hidden mistakes will classify as a lowermost level category – first (1). Moderate hidden mistakes will classify, accordingly, a level 2; serious - level 3. Highest by level will think a catastrophic level 4. Classified by rank, levels of hidden mistakes will be four.

Certain quantity of the insignificant mistakes (I′) gives rise to appearance of several moderate mistakes types (M′), which, in one's turn, gives rise to appearance of certain serious mistakes quantity (S′), and they, accordingly, gives rise to catastrophic mistakes (C′).

Starting data for repeated testing realization are information about basic testing method and operation and finding during basic testing mistake type. It demands composition of reports about basic testing process and results, what is made not always. But firms and collectives, who seriously works on software testing reliability and quality increasing, so fulfil

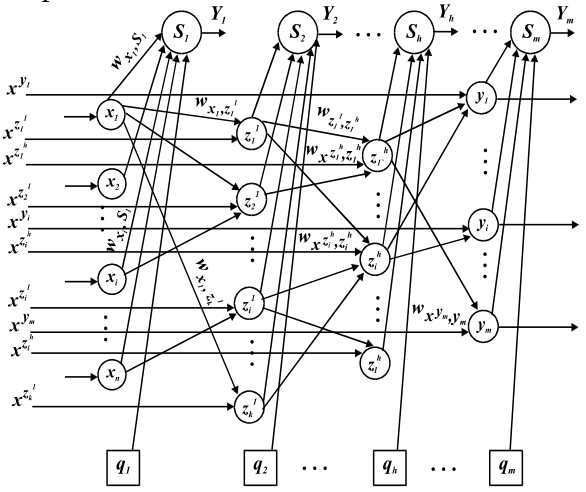
Proposed repeated testing conception is provided for difficultyformalized task decision. In particular it applies to software mistakes importance and interference, carelessness of starting data about off-the-shelf mistakes and so on. Therefore this tasks is solved by calculus of approximations or linear algebra methods very difficult or impossibly.

Category neuronet model of repeated software testing process. On the base proposed conception of software testing process reliability increasing category model of repeated software testing was developed. At the heart of this model artificial neural network (ANN) underlies [2, 3].

Choice of ANN instrument is motivated by ANN enables to allow for importance (weights) every types of latent and non-latent mistakes, thresholds of boundary quantity of admissible mistakes of each category, interference of different category hidden mistakes.

Repeated testing task is determination of weights of influence of one category mistakes on beginnings of others category mistakes. This task is solved by ANN training.

Attached approach is represented by generalized abstract compound ANN, in which a structure of MLP (multi-layer-perceptron) type composed with structure of simple Rosenblatt perceptron (one layer ANN, where weights of first layer invariable, and weighed components of entrance vector on neuron entrance of second layer are summarized). This ANN structure is presented on pic.1.



Pic.1. Category model on base ANN, which different categories software mistakes connection represents

Functional ANN Y_1 is determined by formula::

$$Y_1 = F_{S_1}\left(\sum_{i=1}^{n} x_i w_{x_i} - q_1 w_{q_1}\right).$$

Functional ANN Y_h:

$$Y_h = F_{S_h}\left(\sum_{i=1}^{h}\left(f(z^h)\cdot\left(\sum_{i=1}^{h} z_i^{h-1} w_{z_i^{h-1},z_i^h} + \sum_{i=1}^{\ell} x^{z_i^h} w_{x^{z_i^h},z_i^h}\right)\right)\cdot w_{z_i^h,S_h}\right) - q_h w_{q_h,S_h}$$

where $f(z^h)$ - activation function of neurons of latent layer h; z_i^{h-1} - starting value of activity of i-th neurons of latent layer $(h-1)$; $w_{z_i^{h-1},z_i^h}$ - weight coefficient of connection between i-th neurons of ANN latent layer $(h-1)$ and latent layer h; $x^{z_i^h}$ - i-th ANN entrance, which is directly connected with i-th neuron of layer h; $w_{x^{z_i^h},z_i^h}$ - weight coefficient of connection between entrance $x^{z_i^h}$ and i-th neurons of layer h; $w_{z_i^h,S_h}$ - weight coefficient of connection between i-th neurons of layer h and neuron S_h.

Activation function of latent (associative) layer neurons is hyperbolic tangent function. Activation function of effector layers is linear function. Results of linear activation function of neurons of effector layers are in the range [-1; 1]. Results as 1 or 0 (mistake of *i*-th category level present or absent) are interested, therefore results transformation (round-up) is realized in the following way: $Y_i=1$, if $Y_i>0$; $Y_i=0$, if $Y_i<0$ or $Y_i=0$.

ANN simulation model was developed in Matlab [4, 5]. ANN training and testing process in Matlab was researched [4, 5]. As a result conclusions were made: ANN training fault depends on criterion of training quality evaluation and on form of entrance data presentation. At time index and at index "epoch quantity" training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele and its modifications (Fletcher-Reevse training algorithm, Polak-Ribeyra training algorythm) are the best for considered ANN.

Evaluaton of software hidden mistakes identification reliability. From proposed model it is resulted: when $Y_h > 0$, mistakes of category, which correspond with Y_h, are in program. Let without taking interference at first p_x first category level mistakes, p_{z^1} second category level mistakes, …, p_{z^h} h-th category level mistakes are in program. Taking into account interference of previous category level mistakes on the following level mistakes were stood: p_x - first category level, $p_{z^1} + p_{z_i^{h-1}}$ - h-th category level.

Assume that every category level mistakes and mistakes, which evolved from interference of every previous level on the following level, are identified.

Criterion of software mistakes identification process reliability is quantity of found mistakes under the proposed model.

Reliability D of software hidden mistakes identification process by repeated testing is equal:

$$D = kn_1 p_x + kn_2 \frac{p_{z^1} + p_{x_i}}{p_{z^1}} + \ldots + kn_h \frac{p_{z^h} + p_{z_i^{h-1}}}{p_{z^h}} + \ldots ,$$

where $KN = \{kn_h\}$ - set of norm-coefficients of category hidden mistakes.

Software hidden mistakes identification process reliability increasing is equal $\Delta D = 1 - \frac{D'}{D}$, where D' - software hidden mistakes identification process reliability without taking interference hidden mistakes of previous category level on the mistakes of following category level.

Reliability of hidden mistakes identification process during repeated testing for described above four category level – I, M, S, C is equal:

$$D = kn_1 \cdot p_x + kn_2 \cdot \frac{p_{z^1} + p_{x_i}}{p_{z^1}} + kn_3 \cdot \frac{p_{z^2} + p_{x_i}^{z^1}}{p_{z^2}} + kn_4 \cdot \frac{p_{z^3} + p_{x_i}^{z^2}}{p_{z^3}}$$

From j-th sample following values for repeated software testing reliability determination (table. 1).

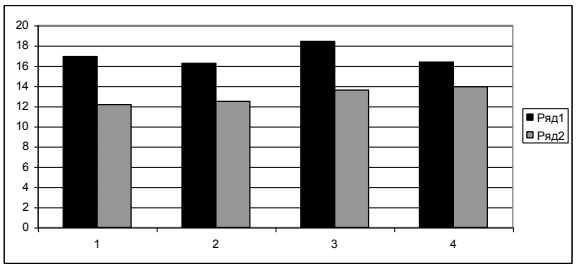
Table 1

Values for repeated software testing reliability determination

№ exp.	p_x	p_{z^1}	p_{z^2}	p_{z^3}	p_{x_i}	$p_{x_i}^{z^1}$	$p_{x_i}^{z^2}$	**D**	**D′**	**ΔD**
1	28	16	10	4	6	4	2	16,92	12,16	0,28
2	32	20	10	6	8	6	2	16,26	12,48	0,23
3	46	28	14	8	10	6	4	18,40	13,60	0,26
4	50	30	18	10	12	8	2	16,36	13,92	0,15

After processing of expert data, which were received from 9 experts-programmers of Sitronics Telecom Solutions (branch-office, Khmelnitsky), norm-coefficients of category hidden mistakes are equal: $kn_1 = 0{,}08$; $kn_2 = 0{,}22$; $kn_3 = 1{,}7$; $kn_4 = 8$.

Reliability of software hidden mistakes identification process during repeated testing with j-th sample is presented on the bar graph (pic.2).



Pic.2. Bar graph of increasing reliability

On the bar graph series 1 (black) represents the reliability D of hidden mistakes identification process during repeated testing, series 2 (grey) represents reliability D' of hidden mistakes identification process without taking interference hidden mistakes of previous category level on the mistakes of following category level

Considering of previous category level mistakes interference raises the reliability of hidden mistakes identification process on 15-28%.

Conclusions. Proposed category model of repeated software testing at the expense of hidden mistakes identification improves the software quality and enables sharp estimation of hidden mistakes identification process reliability.

Reliability increasing will be more, then more hidden mistakes, which influences on the beginnings of following category level mistakes, will be identify.

Література

1. Robert Culbertson, Chris Brown, Gary Cobb. Rapid testing – Prentice hall PTR, 2001. - 384 p.

2. *Локазюк В.М., Пантелєєва (Говорущенко) Т.О.* Категорійна модель процесу повторного тестування дефектів програмного забезпечення // Вісник Технологічного університету Поділля – Хмельницький: ТУП, 2004. – ч.1, т.1, с. 53 – 58.

3. *Lokazyuk V.M., Govoruschenko T.O.* Category Model of Process of Repeated Software Testing // Proceedings of the Third IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems : Technology and Applications. – Sofia, Bulgaria, 2005. – p.241-245

4. *Говорущенко Т.О.* Дослідження моделі вирішувача системи повторного тестування прикладного програмного забезпечення // Вісник Хмельницького національного університету – Хмельницький: ХНУ, 2007 - №3, т.1, с.236-244

5. *Govoruschenko T.O.* Model of decision maker of repeated application software testing system // Радіоелектронні і комп'ютерні системи – Харків: НАУ "ХАІ", 2007 – № 7, с.191-198