UDC 004.932.2:616-006.6

MODELING AND SOFTWARE IMPLEMENTATION OF IT- PROJECT RISKS ASSESSMENT PROCESS

Lyudmyla Honchar[1], Pavlo Pushak[2], Andriy Petruk[3]

Ternopil National Economic University

[1] PhD., associate professor; [2)3)]Master's degree student

I. Statement for the task

For today to check the quality of software, standardized universal indicators have not yet been developed. Their selection and analysis of changes largely depends on the software product which is being developed, the developer organization and the knowledge and experience of the developers themselves. Therefore, the process of risk assessment of software at the design stage is relevant and somewhat complex and lengthy in time, the result of which can significantly affect the cost of the program [1].

II. The purpose of the work

The purpose of the research is to model and program the implementation of the method of software risk assessment at the stage of its design.

III. The algorithm of the program

To assess the risks of IT-projects, it is offered to use a method based on genetic calculations. The result of the operation of the genetic algorithm is to identify the optimal percentage deviations of risks that are possible at the time of designing a software project.

In our case, according to the algorithm of the operation of the program, the user must specify (stages 14): information on the state of the project, the degree of damage from potential risks, the setting of genetic operators, value of the probability of occurrence of risks. At the stage of setting up the genetic algorithm, we set the parameters of the functioning of the genetic algorithm, namely: the methods of cross-breeding, the probability of mutation, the percentage of population fragmentation. At stage 5, we form a symbolic representation of information on the status of the project implementation and the limit values of 5 types of risks. Symbolic model of risk presentation - is the coding of the input data into a binary code for further formation of the chromosome gene's phenotype. For the formation of the initial population (stage 6) the strategy of "focusing" is selected, which gives the opportunity to form all possible combinations of input data, in case when several risk groups will not change, that is, it is a constant. Next, at the same stage, evaluate the health of the population.

Sort the value of the fitness function from smaller to larger for further selection (stage 7). Selection (stage 8) is carried out on the basis of the cut-off method, namely, cut off the prescribed percentage of chromosomes with the highest value of health.

To cross-breeding (stage 9) we select parents randomly from those present in the formed population. There are two methods for cross-breeding: one-point and two-point cross-breeding.

At stage 10, perform a mutation with a given probability, that is, if the chromosome should be mutated, so we completely replace it with another randomly generated or random gene. The health of the population (stage 11) is based on the value of the fitness function (in our case, the amount of degrees of damage to the present software risks). To evaluate the health of the chromosome is split into genes, after which the value of genes, due to their symbol representation, is multiplied by the given coefficients of influence of the group of risks on the degree of damage. The total damage (whole chromosome) will give a percentage of losses when given in the chromosome risks. As a result of the importance of health, we depict the minimum, maximum and average values of the functioning on the graph (stage 12). At stage 13, check whether a given number of epochs of the genetic algorithm has been reached or if there are 2 chromosomes in the population. If not, then proceed to stage 7, if so, then we output the results (stage 14) of the assessment of possible risks and the degree of expense they can lead to.

To complete the operation of the GA two options selected, depending on the user specified percentage cut-off, possible options for GA to finish work at an earlier stage.

Proposed genetic algorithm for estimation are optimal for given input values, which are accompanied by expert assessments, depending on the degree of expenditure for a specific type of risk.

Conclusion

The developed program, depending on the configuration of the genetic algorithm, in a short time finds a minimum degree of damage in the identified process of modeling of risks and can be used both in the development of methodological recommendations, as in assessing software risks at its manufacturers.

References

1. James McCaffrey, Analysis of Vulnerabilities and Project Risks Using PERIL. [Electronic Resource] - Access mode: http://msdn.microsoft.com/en-us/magazine/dd315417.aspx
2. I.Sutskever, J.Martens, G.Dahl, G.Hinton. On the importance of initialization and momentum in deep learning. J. Machine Learning Research, 2013, vol. 28, no. 3, p. 140.

UDC 681.215

MATHEMATICAL AND SOFTWARE IMPLEMENTATION FOR IMPROVING THE EFFICIENCY OF THE MOBILE DEVICE CONTENT PROTECTION

Lyudmyla Honchar [1), Sergii Kondyuk [2), Vitaliy Gritsiv [3), Bohdan Kostyk [4)

Ternopil National Economic University

[1)PhD., associate professor, [2)3)4)Master's Degree Student

I. Statement for the task

In our time, the big problem is that some of the information in the field of economics, politics, as well as individual information can be widely available and have no protection against illegal tampering, copying, blocking or destroying. To solve this problem there are various ways and methods for improving the effectiveness of mobile device content protection [1].

II. The purpose of the work

The purpose of scientific research is the program implementation of the method of increasing the effectiveness of the content protection of mobile devices.

III. Kutter-Jordan-Bossen method for protecting the content of a mobile device

In addition to robustness, the Kutter-Jordan-Bossen algorithm is quite simple to implement: for embedding a digital watermark (DW), there is no need to perform bulky linear transformations of a digital image (DI), DW is built by manipulating color components.

Each image consists of pixels, each pixel represents an combination of three color matrices: red - R, green - G, blue - B, and matrices of transparency - A. The embedding is performed in the blue channel, as the system of human vision is the least sensitive to blue [2].

Let S_i be the bit we embed, container $I = \{R, G, B\}$, $p = (x, y)$, is a pseudo-random position in which the attachment is executed. The secret bit is embedded in the blue channel by modifying the brightness:

$$l(p) = 0{,}299r(p) + 0{,}587g(p) + 0{,}114b(p), \qquad (1)$$

$$b(p) = \begin{cases} b(p) + ql(p), & if \quad S_i = 0 \\ b(p) - ql(p), & if \quad S_i = 1 \end{cases} \qquad (2)$$

where q – coefficient, which specifies the energy of the data bit, which is built on (based on the functional purpose and the features of the steganosystem). Its value depends on the purpose of the scheme. The greater q, the higher the robustness of an attachment, but the stronger its visibility. Extracting a bits by the recipient is carried out without the presence of the original image, namely blindly. For this purpose, the prediction of the value of the output, unmodified pixel is base (p)d on the values of its neighbors. It is suggested to use the values of several pixels located in the same column and the same row for the pixel estimation. The method used a "cross" pixel size 7×7. The estimate b'' is calculated by the formula (3):

$$b(p) = \frac{1}{4c}\left(-2b(p)\sum\nolimits_{i=-c}^{+c} b''(x+i, y) + \sum\nolimits_{k=-c}^{+c} b''(x, y+k)\right), \qquad (3)$$

where c – number of pixels from the top (bottom, left, right) from the estimated pixel.