

## **Selection of metrics in software quality evaluation**

**A V Kopyltsov**

Physics Department, Saint Petersburg State University of Aerospace Instrumentation,  
67 Bolshaya Morskaya Street, St. Petersburg, 190000, Russia

Department of Computer Science, Saint Petersburg State Electrotechnical University,  
5 Professor Popov Street, St. Petersburg, 197376, Russia

kopyl2001@mail.ru

### Convolution procedure

When solving problems in the field of quality of any things or services, there is usually not enough information to make the right decision. Therefore, in this case, a quality assessment is made with some error. An algorithm for assessing quality is proposed, which can be used to assess the quality of a fairly wide range of things and services. We consider a thing that is determined by a finite set of indicators. Each of the indicators has a normalized measure from the interval from 0 to 1. We consider the weighted sum  $S$  of these indicators with weighting factors. The difference from other quality assessment approaches is that the weighting coefficients are not explicitly calculated, but several restrictions are imposed on these coefficients. Firstly, it is assumed that the weighting coefficients take values from 0 to 1 and their sum is 1 (normalization condition). Secondly, it is assumed that the weights can take a finite number of values between 0 and 1, for example, in increments of 0.1 or 0.01 and others (discreteness condition). Thirdly, it is assumed that all indicators are arranged according to their importance for solving a specific task (priority condition). A relation is established between each two adjacent indicators. If  $K_i$  and  $K_{i+1}$  have coefficients for  $P_i$  and  $P_{i+1}$ , then the following options are possible:

- 1)  $K_i \geq K_{i+1}$ , in the case when the indicators  $P_i$  and  $P_{i+1}$  are approximately the same or  $P_i$  is slightly more than  $P_{i+1}$ ;
- 2)  $K_i > K_{i+1}$ , in the case when  $P_i$  is more than  $P_{i+1}$ ;
- 3)  $K_i \gg K_{i+1}$ , in the case when  $P_i$  is much more than  $P_{i+1}$ .

We find the weighted sum  $S$  of these indicators taking into account the listed conditions and options. It turns out that there are several such amounts. We find the average sum  $S_m$ , i.e. their arithmetic mean. It is possible to determine the variance and standard deviation. The resulting numerical value of  $S_m$  is an estimate of the quality of the thing. The procedure for obtaining  $S_m$  is called convolution.

### **Expert method for software quality evaluation**

The algorithm allows us to evaluate the numerical values of the following factors: practicality, integrity, efficiency, correctness, security, reliability, ease of use, evaluation, flexibility, the possibility of use in other conditions, mobility and the possibility of interaction.

A software product has a factor:

- 1) practicality if the software product is convenient for use;
- 2) integrity if unauthorized access to software and data is controlled;
- 3) efficiency if the required functions are performed without unnecessarily expensive resources;
- 4) correctness if the program meets the specifications and achieves the goals set by the user;
- 5) security if the software product allows avoiding danger;
- 6) reliability if the software product satisfactorily performs the necessary functions;
- 7) ease of use if the software product provides for the possibility of updating it in accordance with new requirements;
- 8) evaluation if it is possible to establish a criterion for the acceptability of a software product for a particular application;
- 9) flexibility if the effort required to modify the work program is negligible;
- 10) the possibility of use in other conditions if the reorientation of the program for other applications is possible;
- 11) mobility if the software product can be easily and efficiently used on computers of a different type;
- 12) the possibility of interaction if the effort required to connect one system with another is negligible.

The algorithm includes the following steps:

- 1) The formation of the names of indicators and factors.
- 2) Formation of numbers of indicators that determine factors.
- 3) The choice of the names of indicators, their numerical values and order relations between them.
- 4) Formation of factors determined by selected indicators.
- 5) Formation of order relations between indicators that determine factors.
- 6) Determination of the numerical values of factors, their average values, variances and standard deviations.
- 7) Determining the quality of a software product.
- 8) Outputting results to a display screen or printing device.
- 9) If the result is satisfied, then the end of the program. Otherwise, go to step 3.

### **Expert method for software quality evaluation**

The user is invited to choose from 26 indicators those that must be taken into account when analyzing this software product. The software product has the following indicators:

1) operability, if the program operates in the prescribed modes and processes the required amounts of information in accordance with the documentation in the absence of problems with technical means;

2) the ability to learn, if there is documentation and programs such that the logic of the functioning of the program as a whole and its parts is understandable to specialists;

3) communicative, if the software product and documentation contain a description of the input and output data, the contents and presentation of which are clear to specialists and carry useful information;

4) input / output volume, if the amount of input and output information satisfies the user;

5) input / output speed, if the speed of input and output of information satisfies the user;

6) access regulation, if it is possible to use individual components of software products;

7) access control, if possible to restrict access to resources and objects of the computer system in accordance with the required data protection model;

8) memory efficiency, if memory is used without unnecessarily expensive resources;

9) effectiveness of functioning, if functioning is carried out without unnecessary expenditure of resources;

10) traceability, if it is possible to create records (traces) about the work of software products and their environment;

11) completeness, if the software product contains all the necessary components;

12) robustness, if the computing system is able to recover in the event of erroneous situations;

13) accuracy, if the results produced by the software product have sufficient accuracy for their intended use;

14) error tolerance, if the software product is able to continue functioning in the presence of errors;

15) consistency, if the software product everywhere contains a single symbolism, terminology and meets the requirements;

16) simplicity, if the modular structure of software products is reasonable enough from the point of view of understanding and perception;

17) brevity, if the software product is short enough;

18) the availability of measuring instruments, if there are means for measuring;

19) the redistribution, if the software product allows you to expand the computing functions of individual modules and / or increase the RAM size for storing data;

20) commonality, if the user is provided with a sufficient number of functions when working with software products.

21) informational content, if the software product contains the information necessary and sufficient to understand the purpose of the source data, results, software, limitations, assumptions, individual components of the software products and the current state of the programs included in the software product during their operation;

22) modularity, if the software product is divided into modules, each with its own controlled dimensions, clear purpose and well-defined interface with the external environment;

23) machine independence, if the programs included in the software product can be executed on a computer of a different configuration;

24) the independence from other software, if the software product can function independently;

25) the unification of communication procedures, if the communication procedures are combined according to some principle;

26) unified data, if the data are combined according to some principle.

Each factor is determined by a set of indicators numbered from 1 to 26. The factor of practicality is determined by indicators that have numbers 1, 2, 3, 4, 5; integrity factor - 6, 7; efficiency - 8, 9; correctness - 10, 11, 15; security - 1, 6, 7, 12, 14, 15, 18, 22; reliability - 13, 14, 15, 16; ease of use - 15, 16, 17, 21, 22; evaluation - 16, 18, 21, 22; flexibility - 19, 20, 21, 22; the possibility of use in other conditions - 20, 21, 22, 23, 24; mobility - 21, 24; the possibility of interaction - 22, 25, 26.

### **Modified method of software quality evaluation**

It is assumed that the user wishes to have a software product characterized by factors  $F_1, F_2, \dots, F_n$  with numerical values  $A_1, A_2, \dots, A_n$ . In addition, he has the means to produce a software product, which is determined by indicators  $K_1, K_2, \dots, K_m$  with numerical values  $B_1, B_2, \dots, B_m$ . It is assumed that the material cost function  $X(B_1, B_2, \dots, B_m; R_1, R_2, \dots, R_m)$  is known, which is required when changing the numerical values of the criteria from  $B_1, B_2, \dots, B_m$  to  $R_1, R_2, \dots, R_m$ . Then, by convolving the numerical values of the selected criteria  $K_1, K_2, \dots, K_m$ , we obtain the values  $C_1, C_2, \dots, C_n$  of factors  $F_1, F_2, \dots, F_n$ . These values may differ from the desired values  $A_1, A_2, \dots, A_n$ . If the values  $C_1, C_2, \dots, C_n$  satisfy us, then these values are the desired ones. Otherwise, the numerical values of the criteria change with a sufficiently small step, so that the calculated values of the factors are close to the desired values of the factors with a predetermined accuracy. As a result, we will have several sets of numerical values of the criteria, upon convolution of which we obtain the desired values of the factors with a predetermined accuracy. The selection of one set of criteria values is carried out taking into account the minimization of material costs that are necessary to change the initial values of the criteria. The numerical value of product quality is defined as the convolution of the obtained factor values. If the quality does not meet the requirements imposed by the user, then the choice of criteria or order relations between the selected criteria should be changed. After several changes, the algorithm allows you to get the option that corresponds to the required quality value of the software product. The algorithm is designed in such a way that the number of indicators and factors can be easily increased or changed. With the help of this program, an expert assessment of the quality of a software product can be carried out as a first approximation.

### **Conclusion**

Thus, an expert method for assessing the quality of software products has been developed, which allows, to a first approximation, using a computer with minimal time and money to carry out an expert assessment of the quality of software products. This approach allows you to evaluate and minimize the material costs that are necessary for the development of software products with predefined properties. The proposed expert method can be extended to other tasks. In particular, with the help of algorithms and programs developed on the basis of the proposed method, an expert assessment of the quality and correction of education of schoolchildren or students, the state of the oxygen transport system in the human body, medical and rehabilitation assistance, and sanatorium and resort services and other tasks can be carried out as a first approximation.