

# Improving the Process of Training Specialists in the Development of Program Documentation based on Automated Assessment of the Quality of Skills Formation

Rustam A. Fayzrakhmanov<sup>1</sup>, Ivan S. Polevshchikov<sup>2</sup>, Irina A. Bobrova<sup>3</sup>

Department for information technologies and computer-based systems  
Perm National Research Polytechnic University  
Perm, Russia

<sup>1</sup>fayzrakhmanov@gmail.com, <sup>2</sup>i.s.polevshchikov@mail.ru, <sup>3</sup>bobrovairina1812@gmail.com

**Abstract**— The article is devoted to solving the problems of assessing the quality of software documentation in the training of software developers. Various CASE-tools are widely used at present to improve the efficiency of the software development process, including the preparation of documentation. However, it is necessary to improve methods and means of automating the formation of software development skills that contribute to improving quality and accelerating the learning process. The methodology for assessing the quality of the generated software documentation (on the example of the software testing process) in the course of students performing practical tasks is considered. The methodology is based on the use of mathematical methods, in particular, expert assessment methods. The article also describes a prototype of a subsystem (which is part of an automated training system) used to assess the quality of practical tasks. The advantage of applying this methodology when training at an enterprise is the ability to increase the level of professional skills, speed up the process of adaptation of young employees, and also (when evaluating experienced employees to perform real tasks) improve the quality of the documentation itself.

**Keywords**— *automation of the vocational training process; program documentation; expert assessment methods; software testing*

## I. INTRODUCTION

It is known that in the process of developing software, in addition to the program code itself, various documentation is created [1, 2].

Quality of the documentation affects the next steps in software development. For example, if the formed software requirements contain significant shortcomings, then this will negatively affect the stages of design and implementation of a software product. Or, for example, shoddily developed testing plan and test-cases can contribute that a number of possible errors will not be detected in the program when it is checked.

There are different approaches to improving the quality of program documentation. On the one hand documentation testing methods are used, which consist in verifying its compliance with a certain set of criteria [1]. The use of this type of testing allows you to identify and fix defects in the current version of the documentation.

Testing methods for program documentation and typical criteria for checking are described in detail in the scientific and methodological literature, for example [1, 2]. In particular, the work [1] describes the properties that qualitatively formed software requirements must have (completeness, atomicity, consistency, etc.), test cases (using a competent technical language to describe steps and expected results, maintaining a balance between specificity and generality, between simplicity and complexity, etc.), defect reports (in many ways similar to test cases). In [2], the desired characteristics of software requirements are presented, for example, uniqueness, completeness, and consistency.

On the other hand, there are many CASE-tools to reduce the complexity of the process of preparing documents by a specialist, and, as a result, the negative impact of the human factor [1, 2]. In particular, a brief overview of such tools (for example, software products for compiling test cases and defect reports) is presented in [1].

The level of professional competencies of a specialist (whose responsibilities include the development of this documentation) significantly affects the quality of compilation of program documentation. For example, in order to correctly describe the steps of the test case, the expected results of the test case, the essence of the defect identified in the program, it is necessary to form appropriate skills with the specialist. The use of CASE-tools allows only partially automate the preparation of documents.

An urgent task is the development of automation tools and appropriate algorithms that allow us to assess the quality of practical tasks when training specialists in the skills of developing software documentation. The solution to this problem should be based on the development of computer

---

The research is supported by a stipend of the President of the Russian Federation to young scientists and post-graduate students (No. SP-100.2018.5), which was assigned by the grants Council of the President of the Russian Federation.

technologies used to automate the training process, accelerating this process and contributing to the improvement of its results [3–6].

## II. METHODOLOGY OF AUTOMATED ASSESSMENT OF THE QUALITY OF EXERCISES IN TRAINING DEVELOPMENT OF SOFTWARE DOCUMENTATION

A methodology of assessment the quality of carrying out practical tasks (exercises) has been developed when training beginners in the development of software documentation. The methodology can be used in the course of training of students of higher education and secondary vocational education institutions, as well as in-company training of employees. It is assumed that the instructor (expert) prepares an exercise aimed at the development of program documentation, which is then performed by a group of trainees. An example of such a task could be the development of a set of test cases for a small software system, such as a software module for automatic distribution of the training load in the department.

Application of the considered methodology provides implementation of the subsystem of automated training system (ATS) used for professional training of software developers [6]. The methodology of assessment the quality of execution of practical tasks includes the following steps:

**Step №1.** The instructor defines many quality indicators for assessment of the document prepared by the trainees:  $A = \{a_i | i = \overline{1, N}\}$ , where  $a_i$  – a separate quality indicator,  $N$  – the total number of indicators.

In particular, the instructor can use the quality indicators (criteria for testing software documentation) discussed in the literature on software development [1, 2] (as mentioned in the Introduction above). The teacher's choice of indicators largely depends on: the specifics of a particular software system; regulations and traditions of the organization carrying out the training process (especially in internal training); personal experience in software development.

Let's review an example. To evaluate the quality of the exercise related to building a set of test cases, the instructor selected the following quality indicators (based on the properties of the quality test cases given in [1]):  $a_1$  – use of a competent technical language in the description of steps and expected results;  $a_2$  – maintaining a balance between specificity and community;  $a_3$  – maintaining a balance between simplicity and complexity;  $a_4$  – being able to detect errors in the program with a high probability;  $a_5$  – following a sequence in the description of steps and expected results (relative to the purpose of the test case);  $a_6$  – no unnecessary actions in the description of test-case steps.

**Step №2.** For each  $i$ -th quality indicator, a weight  $w_i$  is determined, which indicates the importance of this indicator in assessing the quality of the document prepared by the trainee during the exercise.

The instructor sets the input data from which the system then automatically calculates the weights. The instructor can manually adjust the values.

Weights can be calculated both by direct evaluation and by paired comparisons [7, 8].

Using the direct evaluation method, each quality indicator is mapped to a numerical value on a specific scale:  $B = \{b_i | i = \overline{1, N}\}$ , where  $b_i$  – value associated with  $i$ -th indicator. The higher the value  $b_i$ , the more important  $i$ -th indicator is when evaluating a particular exercise. The choice of scale depends on the teacher's knowledge of the features of the compared indicators, as well as on the required accuracy of measurements.

The formula used to calculate weights  $w_i$  is:  $w_i = b_i / \sum_{i=1}^N b_i$ . Continuing to consider the example above (step №1), we compare the values of  $b_i$  quality indicators (on a 10-point scale):  $b_1 = 6$ ,  $b_2 = 9$ ,  $b_3 = 7$ ,  $b_4 = b_5 = 10$ ,  $b_6 = 8$ .

Where  $\sum_{i=1}^N b_i = 6 + 9 + 7 + 10 + 10 + 8 = 50$ , and weights of indicators are equal:  $w_1 = 6/50 = 0,12$ ;  $w_2 = 0,18$ ;  $w_3 = 0,14$ ;  $w_4 = w_5 = 0,2$ ;  $w_6 = 0,16$ .

Using the method of paired comparisons, preference of each quality indicator is established by comparison of all possible pairs of indicators [7]. Initially, a matrix of paired comparisons of indicators is formed, which we will present as  $C = (c_{ij})$ , where  $c_{ij}$  – numerical estimation of preference of  $i$ -th quality indicator over  $j$ -th, where  $c_{ij} = 1/c_{ji}$ ,  $c_{ij} \neq 0$ ,  $i = j = \overline{1, N}$ . On the basis of matrix  $C$  the weights of quality indicators are calculated:

$$w_i = v_i / \sum_{g=1}^N v_g, \text{ where } v_i = \sqrt{\prod_{j=1}^N c_{ij}}.$$

A teacher's choice of the most preferred method in each particular situation depends on a number of factors. In particular, if it is easier for the teacher to organize all quality indicators immediately [7], then the method of direct evaluation should be chosen. Otherwise, it is more rational to make a pair comparison of indicators, but in this case it will be necessary to obtain more data from the expert.

**Step №3.** A training group is given a document development exercise. Trainees perform it in a certain time.

**Step №4.** At the end of the exercise, the instructor reads the document created by each of the trainees and evaluates it according to the established quality indicators.

For each of the indicators, the score is displayed on a line [0;1] (when the screen forms are filled in for clarity, the score

is displayed in the form of interest), where 0 (0%) means that the document does not comply with the quality indicator, and 1 (100%) means that the score is fully compliant.

The results of this step represent the matrix  $D = (d_{ki})$ , where  $d_{ki} \in [0;1]$  – assessment of the document created by the  $k$ -th trainee ( $k = \overline{1, M}$ , and  $M$  – number of trainees in the group) by  $i$ -th quality indicator.

For this example, assume that the following matrix is obtained from the assessment of the exercise by a group of five trainees for the six quality indicators above:

$$D = \begin{pmatrix} 0,6 & 0,7 & 0,65 & 0,45 & 0,85 & 1 \\ 0,35 & 0,65 & 0,5 & 0,3 & 0,6 & 0,25 \\ 0,75 & 0,7 & 1 & 0,95 & 0,8 & 0,85 \\ 0,6 & 0,5 & 0,75 & 0,9 & 0,55 & 0,45 \\ 0,35 & 0,15 & 0,5 & 0,1 & 0,1 & 0,45 \end{pmatrix}$$

**Step №5.** Quality indicator weights  $w'_i$  are calculated based on the spread of estimates obtained in step № 4. Weight  $w'_i$  for the  $i$ -th quality indicator, the more the rating  $d_{ki}$  relative to this indicator [7]. Calculation of  $w'_i$  values allows to take into account actual results of assessment of documents created by group of trainees.

The process of calculating the weights  $w'_i$  can be represented by the following sequence of actions [7]:

5.1. Calculation of average estimates for each  $i$ -th quality indicator by formula:  $d_i = \sum_{k=1}^M d_{ki} / M$ . For the example of matrix  $D$  above (step № 4), the values of  $d_i$  will be:

$$d_1 = (0,6 + 0,35 + 0,75 + 0,6 + 0,35) / 5 = 0,53; \quad d_2 = 0,54; \\ d_3 = 0,68; \quad d_4 = 0,54; \quad d_5 = 0,58; \quad d_6 = 0,6.$$

5.2. Calculation of variation values for each  $i$ -th indicator

by formula:  $R_i = \frac{\sum_{k=1}^M |d_{ki} - d_i|}{M \cdot d_i}$ . For the reviewed example:

$$R_1 = \frac{2 \cdot |0,6 - 0,53| + 2 \cdot |0,35 - 0,53| + |0,75 - 0,53|}{5 \cdot 0,53} \approx 0,272; \\ R_2 \approx 0,319; \quad R_3 \approx 0,229; \quad R_4 \approx 0,57; \quad R_5 \approx 0,352; \\ R_6 \approx 0,433.$$

5.3. Calculation of the sum of the spread values by formula:  $R = \sum_{i=1}^N R_i$ . For the reviewed example:  $R \approx 2,175$ .

5.4. Calculation of weights  $w'_i$  directly by formula:  $w'_i = \frac{R_i}{R}$ . For the reviewed example:  $w'_1 \approx 0,125$ ;  $w'_2 \approx 0,146$ ;  $w'_3 \approx 0,105$ ;  $w'_4 \approx 0,262$ ;  $w'_5 \approx 0,162$ ;  $w'_6 \approx 0,199$ .

**Step №6.** Calculation of generalized weights  $w''_i$  of quality indicators by formula  $w''_i = \alpha w_i + \beta w'_i$ , where  $\alpha$  and  $\beta$  – coefficients reflecting the importance of weights  $w_i$  and  $w'_i$  respectively. Here  $\alpha + \beta = 1$ . The following values are used frequently [7]:  $\alpha = \beta = 0,5$ .

For the reviewed example (when  $\alpha = \beta = 0,5$ ):  $w''_1 \approx 0,122$ ;  $w''_2 \approx 0,163$ ;  $w''_3 \approx 0,123$ ;  $w''_4 \approx 0,231$ ;  $w''_5 \approx 0,181$ ;  $w''_6 \approx 0,18$ .

In fact,  $\alpha$  and  $\beta$  determine the importance of each of the two types of quality weights ( $w_i$  and  $w'_i$ ) when calculating the generalized weight. The selection of  $\alpha$  and  $\beta$  depends on different factors. In particular, in determining  $\alpha$  and  $\beta$ , it is essential how reliable and accurate the weights  $w_i$  obtained by expert assessment based on the teacher's data. The selection of values  $\alpha$  and  $\beta$  can also be done by analyzing the accumulated data on the document assessment process over a period of time.

**Step №7.** The system calculates the complex assessment of the document prepared by each  $k$ -th trainee by the formula:

$L_k = \sum_{i=1}^N w''_i d_{ki}$ . The teacher can correct the automatically obtained complex assessment.

For the reviewed example:  $L_1 \approx 0,7$ ;  $L_2 \approx 0,43$ ;  $L_3 \approx 0,85$ ;  $L_4 \approx 0,64$ ;  $L_5 \approx 0,25$ .

The higher the complex assessment, the higher the quality of the prepared document. Thus, in this example, the best way to handle the exercise of the trainee № 3, the worst – the trainee № 5.

**Step №8.** It is possible to convert the obtained value of the complex assessment  $L_k$  into a 4-point scale widely used in the education system [9] (this step may be necessary, for example, if students of the educational institution perform the exercise).

For example, the conversion can be performed according to the following rule:  $0,85 \leq L_k \leq 1$  – «excellent» (5);  $0,7 \leq L_k < 0,85$  – «good» (4);  $0,5 \leq L_k < 0,7$  – «satisfactory» (3);  $0 \leq L_k < 0,5$  – «unsatisfactory» (2).

The system initially converts to a 4-point scale automatically, and the teacher can then adjust the score.

**Step №9.** The teacher can prepare remarks and recommendations for the trainee on the drafting of the document aimed at improving its quality.

### III. SUBSYSTEM FOR AUTOMATED ASSESSMENT OF THE QUALITY OF EXERCISES IN TRAINING DEVELOPMENT OF SOFTWARE DOCUMENTATION

Functional requirements for the ATS subsystem, by means of which implementation of the above-mentioned procedure is

realized, are presented using the UML use case diagram [2] in Fig. 1.

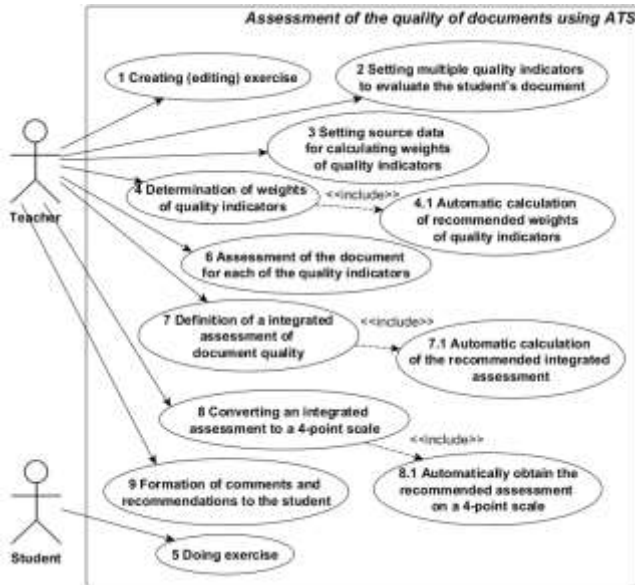


Fig. 1. Functional requirements for the ATS subsystem

Apparently on Fig. 1, the subsystem allows to calculate the values used at assessment of skills of the trainee recommended to the teacher.

Let's consider models of user interfaces of the created subsystem realized as the web application. The interface for definition the weights of quality indicators (in relation to exercise on development of set of test cases) is provided on Fig. 2. For each indicator the teacher selects value of importance from the revealing list (value from 1 to 10) with a possibility of manual entry. After clicking of the "Calculate" button on the screen the recommended value of weight which the teacher, if necessary, can change manually in a text box is displayed.

Quality indicator name	Importance of quality indicator	Quality indicator weight, %
1 Use of a competent technical language in the description of steps and expected results	6	12 (12%)
2 Maintaining a balance between specificity and commonality	9	18 (18%)
3 Maintaining a balance between simplicity and complexity	7	14 (14%)
4 Being able to detect errors in the program with a high probability	10	20 (20%)
5 Following a sequence in the description of steps and expected results (relative to the purpose of the test case)	10	20 (20%)
6 No unnecessary actions in the description of test-case steps	8	16 (16%)

Fig. 2. Defining quality indicator weights

The interface for assessment of set of test cases created by the trainee is provided on Fig. 3. Having clicked a hyperlink "Go to viewing the created test cases", the teacher can study the made document. During the assessment process for each quality indicator the teacher selects a value from the list (from

0% to 100%) with the possibility of manual entry. After clicking of the "Calculate" button, the recommended values of complex assessment (on 100-mark and 4-mark scales) which the teacher, if necessary, can change manually are displayed. The text area for input of remarks and recommendations to the trainee on improvement of the document is provided.

Go to viewing the created test cases

(By clicking on this link, you can read and evaluate the quality of the created document.)

Quality indicator name	Assessment according to a quality indicator, %
1 Use of a competent technical language in the description of steps and expected results	60
2 Maintaining a balance between specificity and commonality	70
3 Maintaining a balance between simplicity and complexity	66
4 Being able to detect errors in the program with a high probability	46
5 Following a sequence in the description of steps and expected results (relative to the purpose of the test case)	85
6 No unnecessary actions in the description of test-case steps	100

Final grade (%):  (  )

Final grade (on a 4-point scale):  (  )

Remarks and recommendations:

Fig. 3. Assessment of set of test cases

#### IV. CONCLUSION

The developed methodology allows us to formally (based on the use of mathematical methods) evaluate the quality of compiling program documentation during vocational training. The application of this methodology at the enterprise helps to increase the level of development of skills and accelerate the adaptation of a young specialist:

1) In the course of in-company training, tasks for the development of documentation are issued that are similar to the real production problems that are solved in this organization. The assignment is evaluated by a teacher with extensive experience in practical work on the preparation of documents.

2) At the initial stage of labor activity, the implementation of real production tasks (for some customers) is initially evaluated by a highly qualified specialist (for example, the head of a development team or a leading programmer). On the one hand, this allows you to assess the level of development of skills. On the other hand, a young specialist, based on a list of remarks, can improve the existing version of the document (which helps to improve the quality of the software product as a whole).

Perspectives of further researches in the field of automated scoring of quality of drawing up program documentation are:

1) Involvement of the group of experts for receiving scales of figures of merit and assessment of the document made by the trainee. This situation is possible, for example, when holding the student's Olympiad in higher education institution. As a result, application of methods of group expert assessment will be required [7].

2) Execution of the automatic analysis of documents for their preliminary estimate on some of indicators. For example,

implementation of search of phrases it seems "the large volume of data", "low speed", "high quality", "the evident interface" which can confirm incompleteness of work on the document, a possibility of its subjective interpretation [1].

#### REFERENCES

- [1] Kulikov S.S. *Testirovanie programmnogo obespecheniya. Bazovyy kurs* [Software testing. Basic course]. Minsk: Four quarters Publ., 2017. 312 p. (In Russian)
- [2] Orlov S.A., Tsil'ker B.Ya. *Tekhnologii razrabotki programmnogo obespecheniya: Uchebnik dlya vuzov. 4-e izd. Standart tret'ego pokoleniya* [Software development technologies: Textbook for universities. 4th ed. Third generation standard]. St. Petersburg: Peter, 2012. 608 p. (In Russian)
- [3] Bouhnik D., Carmi G. E-learning Environments in Academy: Technology, Pedagogy and Thinking Dispositions. *Journal of Information Technology Education: Research*. 2012, V. 11, pp. 201-219.
- [4] Kovacic Z., Green J. Automatic Grading of Spreadsheet and Database Skills. *Journal of Information Technology Education: Innovations in Practice*, 2012, V. 11, pp. 53-70.
- [5] Lisitsyna L.S., Smetyuh N.P., Golikov S.P. Models and Methods for Adaptive Management of Individual and Team-Based Training Using a Simulator. *IOP Conference Series: Earth and Environmental Science*, 2017, V. 66, No. 1, P. 012010.
- [6] Fayzrakhmanov R.A., Polevshchikov I.S. The use of mathematical methods to automate the control of skills in the study of software testing algorithms. *XXII Mezhdunarodnaya konferentsiya po myagkim vychisleniyam i izmereniyam (SCM-2019)* [XXII International Conference on Soft Computing and Measurements (SCM-2019)]. SPb.: St. Petersburg Electrotechnical University "LETI", 2019. Section 2. pp. 162-166. (in Russian)
- [7] Gudkov P.A. *Metody sravnitel'nogo analiza. Ucheb. posobie* [Methods of comparative analysis. Textbook]. Penza: Publishing House Penz. state University, 2008. 81 p. (In Russian)
- [8] Goldstein A.L. Multicriteria assessment of alternatives. *Vestnik PNIPU. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniya* [PNRPU Bulletin. Electrotechnics, Informational Technologies, Control Systems], 2012, No. 6, pp. 18-24. (in Russian)
- [9] Novikov D.A. *Statisticheskie metody v pedagogicheskikh issledovaniyakh (tipovye sluchai)* [Statistical techniques in pedagogical researches (standard cases)]. M.: MZ-Press, 2004. 67 p. (In Russian)