Method of Evaluating the Success of Software Project Implementation Based on Analysis of Specification Using Neuronet Information Technologies

Tetiana Hovorushchenko[1], Andriy Krasiy[2]

[1] Khmelnitsky National University, Khmelnitsky, Ukraine
tat_yana@ukr.net
[2] Khmelnitsky National University, Khmelnitsky, Ukraine
andriy-krasiy@yandex.ua

Abstract. The actuality and importance of skill to evaluate the possible success of software project based on SRS were showed in this paper. The aim of research is prediction of characteristics and evaluating the success of software project implementation based on analysis of SRS. Method of evaluating the success of software project implementation based on analysis of SRS using neuronet information technologies was first proposed. This method provides the prediction of success of software projects implementation, comparison of software projects on the basis of SRS and choice of the best SRS of project.

Keywords: software requirements specification (SRS), software project, success of project implementation, SRS indicators, project characteristics, integrative indicator of project, the degree of success of the project implementation.

Key Terms: Model-Based Software System Development, Software Component, Software System, Specification Process.

1 Introduction

Statistics of success of software projects implementation according to The Standish Group International [1] showed that the rate of challenged projects (that late, over budget, and/or with less than the required features) is the constant value (42-46% projects). These statistics reflect the high rate of non-quality (the failed and the challenged) software projects in terms of interpretation of software quality [2].

As shown in [3], the errors of requirements formulation are 10-25% of all errors. The analysis of errors of embedded and application software, which were made at the stage of the requirements formulation, is given in [4]. In [5-7] the fact is confirmed, that the causes of many incidents and accidents through software are in the SRS, rather than in coding. In [6] the experiment is described, which showed that the software versions written by different developers for the same requirements, contain the joint errors associated with errors of SRS. These experimental statements leads to the need to deepen of the SRS analysis. So *the actual and important* is the skill of evaluation of the success of project implementation on the basis of SRS. *The aim of this research* is the prediction of the characteristics and evaluation of success of implementation of software project based on the SRS analysis.

The success of software project implementation is timely execution of software project within the allocated budget and with realization of all necessary features and functionality. It can be estimated at the design stage based on the predicted values of the main project characteristics [8-10] - duration, cost, complexity, cross-platform, usability and quality. *Duration* is the sequence of the project stages based on the needs of project management. The relative duration is evaluated as compared to other software projects. *Cost* is difficult to assess at the early stages because it is highly dependent on the number of lines of code (the cost of one line is 0.5$). At the early stages of the life cycle we can evaluate the relative cost (as compared to other projects). *Complexity* is determined by the number of interacting components, the number of connections between the components and the complexity of their interactions. *Cross-platform* is the ability of software to run on more than one hardware platform and/or operating system. *Usability* is effectiveness, profitability and satisfaction of users by software project. *Quality* is the degree of compliance with the software characteristics of requirements. From the determinations of characteristics it is clear that none of them are part of other characteristic, that justifies this choice [9, 10].

Analysis shows, that the existing methods and tools [9, 10] of characteristics determination are not suitable to evaluation of their values at the stage of requirements formulation, since they focus on the ready source code. The known methods (Using natural language processing technique, Using CASE analysis method, QAW-method, Using global analysis method, O'Brien's approach, Method to discover missing requirement elicitation, Selection of elicitation technique, Comparison and categorization of requirements elicitation techniques, Techniques for ranking and prioritization of software requirements) and tools (OSRMT, Tools by LDRA, Sigma Software, DEVPROM, CASE.Analytics) of SRS analysis and existing technologies of risk management (SEI, SRE, CRM, TRM, FSI, ERM) [9-13] are not suitable for quantitative evaluation of the project characteristics, because all are targeted to control over compliance with requirements of SRS, but none of them define the predicted values of characteristics on the SRS analysis.

Then for prediction of success of software project implementation on the analysis of SRS *the task of research* is development of method of evaluating the success of software project implementation based on analysis of specification.

2 Method of Evaluating the Success of Software Project Implementation Based on Analysis of Specification Using Neuronet Information Technologies (MESSPI)

Method of evaluating the success of software project implementation based on analysis of SRS consists of next stages: 1) neuronet prediction of characteristics of software project based on the analysis of specification; 2) interpretation of the received relative values of the software project characteristics; 3) evaluation of the degree of success of the software project implementation; 4) testing of the stability and acceptability of compensations of software project characteristics.

Let the software project is specified by the SRS [14] in the next formalized form:

$$SRS=\langle R1,R2,R3,R4\rangle, \tag{1}$$

where R1 – the set of indicators of section1 of the SRS, R2 – indicators of section2, R3 – indicators of section3, R4 – indicators of section4. Selection and possible values of SRS indicators from the sets R1-R4 were detailed in [9].

The *first stage of MESSPI* is prediction of software project characteristics on the SRS analysis, result of that is determining of the relative values of characteristics:

$$SCH=\{Cs,Dsp,Cx,Cp,Ub,Qs\}, \tag{2}$$

where Cs – software project cost, Dsp –duration, Cx –complexity, Cp – cross-platform, Ub – usability, Qs – quality.

Some indicators of specification [9] affect the above characteristics, but equations is not known, by which can calculate the characteristic value on the basis of the sets of SRS indicators – all available formulas of characteristics evaluation is oriented to ready source code [9, 10]. Hecht-Nielsen's theorem proves the possibility of solving the task of representation of multidimensional function of arbitrary form on the artificial neural network (ANN). Therefore, ANN will be used to implement of the unknown functions of dependence of the project characteristics on SRS indicators. In [9] the ANN was developed, which processes and approximates the set of SRS indicators and provides the predicted quantitative values of characteristics - Fig. 1. Selection and possible values of ANN inputs, equations for ANN functioning and forming of ANN outputs (predicted relative values of the characteristics) were detailed in [9], so this information is not represented in this paper.

ANN of characteristics prediction based on the SRS analysis was trained so that all values of characteristics are the values of the interval (0, 1]. The value of each characteristic nearly to 0 negative affects on the success of project implementation (high cost, duration and complexity; low quality, usability, cross-platform). The value nearly to 1 positive impacts on the success of the project implementation (low cost, duration, complexity; high quality, usability, cross-platform).

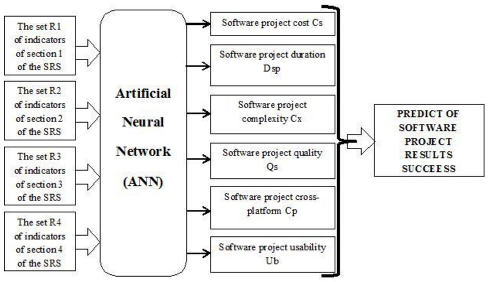

Fig. 1. The concept of neuronet prediction of characteristics of software project based on the analysis of specification

Let the ANN provided the following set of values of characteristics of project Sp:

$$SCH_{ANN}=\{Cs_{ANN}, Cx_{ANN}, Dsp_{ANN}, Ub_{ANN}, Cp_{ANN}, Qs_{ANN}\} \quad (3)$$

The developers and customers are difficult to comprehensively assess the success of software project implementation on the basis of the ANN's relative values of main characteristics. Therefore, the *second stage of MESSPI* is the interpretation of the received relative values of the project characteristics.

For this we introduce the integrative indicator of software project. *Integrative indicator Iip_{Sp}* – is the quantitative indicator of project implementation success based on the set SCH_{ANN}. We cannot to establish mutual dependence of them and to determine their impact on the integrative indicator of software project - these formulas and functions are not available. Therefore, we assume that all six predicted characteristics are equally important to the success of the project, and the integrative indicator of project depends equally on all six characteristics. In the absence of formulas and functions the simplest and the most obvious way of definition of integrative indicator of project is the using of its graphic presentation (in the classic radar chart, the axes of which there are six characteristics of the project - Fig. 2). Then the integrative indicator of project is area of figure, which are shaped the predicted (by ANN) values of the project characteristics. Because ANN predicts the values of 6 characteristics, the coordinate system (Radar chart) will have 6 axes (the angle between the axes is 60°), and in accordance the integrative indicator of project is area of the hexagon $Cs_{ANN}Cx_{ANN}Dsp_{ANN}Ub_{ANN}Cp_{ANN}Qs_{ANN}$ highlighted thick line on Fig. 3.

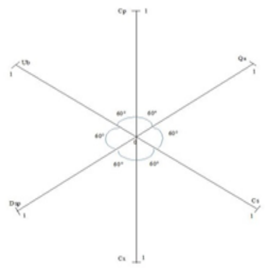

Fig. 2. The coordinate system for Iip_{Sp}

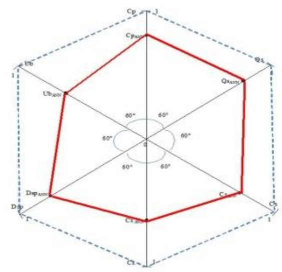

Fig. 3. The graphical representation of Iip_{Sp} and Iip_{max}

For calculation of integrative indicator Iip_{Sp} we will divide the hexagon into six triangles, will calculate the area of each triangle with two sides (value of characteristics) and angle between them (60°) and will add the obtained values of triangles areas:

$$S_{CsOCx}=½*Cs_{ANN}*Cx_{ANN}*\sin60°=0.5*0.866* Cs_{ANN}*Cx_{ANN}, \quad (4)$$

$$Iip_{Sp}=0.5*0.866*(Cs_{ANN}*Cx_{ANN}+ Cx_{ANN}*Dsp_{ANN}+ Dsp_{ANN}*Ub_{ANN}+ Ub_{ANN}*Cp_{ANN}+ +Cp_{ANN}*Qs_{ANN}+ Qs_{ANN}*Cs_{ANN}) \quad (5)$$

The order of hexagon axes was selected taking into account of features of ANN training and for reasons of inability of compensation of the low values of some characteristics by high values of other characteristics (as all six characteristics are important for the software project). Formula (5) shows that pairwise multiplication of the characteristics values can allow these compensations. Therefore, the upper part of the coordinate system has three axes for characteristics Ub, Cp, Qs, and the lower part consists of three axes for characteristics Dsp, Cx, Cs, for which the rule of ANN training is: the value of characteristic nearly to 0 means high cost, duration, complexity and low quality, usability, cross-platform. The junction of axes for characteristics from different categories was selected in pairs exactly as low value of cost (Cs→1) shall not compensate low value of quality (Qs→0), short value of duration (Dsp→1) can not compensate low value of usability (Ub→0).

We will need also the maximum possible value of integrative indicator of project: Iip_{max} – is the area of hexagon CsCxDspUbCpQs highlighted dotted line on Fig. 3. ANN was trained so that maximum possible value of each characteristic – is 1. Then:

$$Iip_{max}=0.5*0.866*(1*1+ 1*1+ 1*1+ 1*1+ 1*1+ 1*1)=2.598 \quad (6)$$

By itself, the integrative indicator of project is uninformative to the developer and customer due to the difficulty of interpretation of its value, therefore the *third stage of MESSPI* is the evaluation of the degree of success of project implementation based on the integrative indicator of project. The value Iip_{max}=2.598 – is the best value of integrative indicator, then the degree P_{Iip} of success of project implementation is:

$$P_{Iip}=Iip_{Sp}/Iip_{max}=Iip_{Sp}/2.598=0.385*Iip_{Sp} \quad (7)$$

The value of the degree of success of the software project implementation nearly to 0 indicates the low success of software project implementation.

As mentioned above, the compensation of values of the characteristics with the same value of integrative indicator is not always correct. Then the *fourth stage of MESSPI* is the testing of the stability and acceptability of characteristics compensations. If the hexagon $Cs_{ANN}Cx_{ANN}Dsp_{ANN}Ub_{ANN}Cp_{ANN}Qs_{ANN}$ (area of which is the integrative indicator) will be convex, the characteristics of software project is considered the stable, and their compensatory effects are acceptable (valid). We introduce the *indicator* Ace_{Sp} *of stability and acceptability of compensatory effects of the characteristics.* This indicator will take the value "True", if characteristics are stable, their compensatory effects are acceptable (i.e. hexagon is convex).

Criterion of convexity of hexagon is the simultaneous fulfillment of two conditions: 1) the same sign of sines of all angles of the hexagon; 2) the sum of all the angles of hexagon is 720° (by theorem about sum of the angles of convex polygon).

Here are the steps to determine of the angles of the hexagon (by Fig. 3): 1) calculate the unknown third side for each triangle by law of cosines; 2) find one unknown angles in each triangle by law of cosines; 3) find second unknown angle in each triangle by theorem about the sum of angles; 4) find the angles of the hexagon.

After finding of the angles of the hexagon we should find sines of obtained angles and compare their signs. And we should find the sum of the obtained angles and compare this sum with 720°. If the sum of the angles of hexagon is 720° and sines of angles have the same signs, then hexagon is convex, accordingly indicator of stability and acceptability of compensatory effects of the characteristics Ace_{Sp}=True.

3 Experiments

We performed experiments on the practical use of the MESSPI. For this we considered four alternative software projects, developed by different teams of developers to solve the same task – development of support system (web-portal) for practices of students of IT-specialties. Each development team consists of three IT professionals: project manager, requirements engineer and web-developer. Specialists from different teams had the same level of qualifications and the same experience in similar projects: project manager and requirements engineer of each team previously worked in three similar successful projects, web-developer of each team previously worked in two similar successful projects. All four development teams represented the different software companies of Khmelnitsky. Each development team had the equal opportunity to communicate with the customer for identification of customer requirements. Three joint meetings of all developers of four teams and representatives of the customer were organized. In addition, individual meetings of team representatives and representatives of the customer took place. As a result of working together with customer representatives all four development teams offered their SRS.

The sets R1-R4 of SRS indicators were formed for the each of four SRS and submitted for processing to the ANN. The results of ANN (predicted relative values of the characteristics), the calculated by MESSPI integrative indicators and degree of success of these projects implementation are in Table 1.

Table 1. Predicted relative values of characteristics, calculated integrative indicators and degree of success of four software projects implementation

Characteristics and indicators of software project	Values for Project1	Values for Project2	Values for Project3	Values for Project4
Cost Cs_{ANN}	0.8	0.22	0.39	0.59
Duration Dsp_{ANN}	0.9	0.19	0.41	0.57
Complexity Cx_{ANN}	0.75	0.31	0.37	0.62
Usability Ub_{ANN}	0.85	0.15	0.5	0.56
Cross-platformCp_{ANN}	0.87	0.21	0.47	0.57
Quality Qs_{ANN}	0.89	0.17	0.49	0.61
Integrative indicator Iip_{Sp}	*1,847*	*0,113*	*0,501*	*0,894*
The degree of success P_{Iip}	*0.7111*	*0,0435*	*0.1929*	*0.3442*

Thus, the results of Table 1 demonstrate that Project1 has the greatest predicted degree of success of implementation (71%) and Project2 has the smallest predicted degree of success of implementation (about 4%). Therefore the Project1 (SRS of Project1) was proposed to the developer and the customer for solution of their task.

If we will not take into account the compensation of low values of some characteristics by high values of other characteristics in the calculation of integrative indicator of the project, there is a risk for the obtaining of following results. Let the ANN gived certain values of characteristics for five different software projects. We show these values and the corresponding values of integrative indicators in Table 2.

The data of Table 2 show that all five software projects have the same integrative indicator Iip_{Sp}=0.894, but have significantly different relative values of characteristics. We need to check the convexity of the hexagons for all examined software projects for determination of value of indicator Ace_{Sp} - Table 3.

Table 2. Examples of compensation of characteristics for different software projects

Characteristics and indicators of project	Values for Pr.4	Values for Pr.5	Values for Pr.6	Values for Pr.7	Values for Pr.8
Cost Cs_{ANN}	0.59	0.7	1	1	0.93
Duration Dsp_{ANN}	0.57	0.57	0.57	0.57	0.57
Complexity Cx_{ANN}	0.62	0.62	0.62	0.62	0.62
Usability Ub_{ANN}	0.56	0.56	0.56	0.403	0.56
Cross-platformCp_{ANN}	0.57	0.57	0.57	0.57	0.57
Quality Qs_{ANN}	0.61	0.503	0.289	0.403	0.33
Integrative indicator Iip_{Sp}	*0.894*	*0.894*	*0.894*	*0.894*	*0.894*

Table 3. Testing of the stability and acceptability of compensatory effects of the characteristics for eight software projects

Values	Pr.1	Pr.2	Pr.3	Pr.4	Pr.5	Pr.6	Pr.7	Pr.8
Sine of angle Qs	+	+	+	+	+	-	+	-
Sine of angle Cs	+	+	+	+	+	+	+	+
Sine of angle Cx	+	+	+	+	+	+	+	+
Sine of angle Dsp	+	+	+	+	+	+	+	+
Sine of angle Ub	+	+	+	+	+	+	+	+
Sine of angle Cp	+	+	+	+	+	+	+	+
Indicator Ace_{Sp}	True	True	True	True	True	*False*	True	*False*

The testing of the stability and acceptability of compensations of characteristics of software projects showed that for Project6 and Project8 the characteristics are unstable, i.e. compensations of these characteristics are unacceptable.

4 Conclusions

This paper shows: the need of deepening of the SRS analysis; the dependence of quality and success of software project implementation on the SRS; the actuality and importance of the skill of evaluation of software project implementation success based on the SRS; the need of support of the choice of the best SRS for the project.

The authors first proposed the method of evaluating the success of software project implementation based on analysis of specification using neuronet information technologies. MESSPI differs from the known methods (analysed in [8-13]) that provides the prediction of the success of software projects implementation based on only SRS. The practical significance of the proposed method is the support in the comparison of software projects on the basis of SRS, the choice of the best SRS of project, and control for SRS quality also (SRS quality is very importance, as known [14]). The proposed method is suitable only for software projects, for which SRS are existing and available. This method helps to "cut off" the software projects with failed SRS, because, as shown above, the software projects with failed requirements and specifications can not be successfull at the implementation.

The authors have following perspectives for future researches: 1) increasing of the veracity of ANN functioning for increasing of the MESSPI veracity; 2) selection of variant component for ANN; 3) providing recommendations about that is necessary to be changed in the SRS, that project became successful; 4) development of information technology for prediction of characteristics and evaluation of success of software project implementation based on the SRS analysis; this information technology should support: the SRS indicators collection, the processing of this data by ANN, the collection of the relative values of characteristics, the calculation of the integrative indicator and the degree of success of the software project implementation, and testing of the stability and acceptability of characteristics compensations.

References

1. The Standish Group International: CHAOS Manifesto – Think big, act small. Technical report, CHAOS Knowledge Center (2013)
2. Bourque, P., Fairley, R.: Guide to the software engineering body of knowledge (SWEBOK): Version 3.0. A project of the IEEE Computer Society (2014)
3. McConnell, S.: Code complete. Microsoft Press (2013)
4. Pomorova, O., Hovorushchenko, T.: The modern problems of software quality evaluation. Radioeletronic and computer systems. 5, 319-327 (2013) [in Ukrainian]
5. Levenson, N.G.: Systemic factors in software-related spacecraft accidents. In: AIAA Space Conference and Exposition, pp.1-11 (2001)
6. Levenson, N.G.: Software challenges in achieving space safety. Journal of the British Interplanetary Society. 62, 265-272 (2009)
7. Ishimatsu, T., Levenson, N., Thomas, J., Fleming, C., Katahira, M., Miyamoto, Y., Ujiie, R.: Hazard analysis of complex spacecraft using systems-theoretic process analysis. Journal of Spacecraft and Rockets. 51, 509-522 (2014)
8. Maedche, A., Botzenhardt, A., Neer, L.: Software for people: fundamentals, trends and best practices. Springer-Verlag Berlin Heidelberg, Berlin (2012)
9. Krasiy, A.: Modelling of process of prediction of software characteristics based on the analysis of specifications. Computer-Integrated Technologies: Education, Science, Industry. 66-76 (2014) [in Ukrainian]
10. Fenton, N.: Software metrics: A rigorous approach (3rd edition). CRC Press (2014)
11. Chen, A., Beatty, J.: Visual models for software requirements. MS Press, Washington (2012)
12. Fatwanto, A.: Software requirements specification analysis using natural language processing technique In: International Conference on Quality in Research, pp.105-110 (2013)
13. Rehman, T., Khan, M.N.A., Riaz, N.: Analysis of requirement engineering processes, tools/techniques and methodologies. I.J. Information Technology and Computer Science. 40-48 (2013)
14. IEEE 830-1998. Recommended practice for software requirements specifications (1998)