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

Tetiana Hovorushchenko,

Doctor, Senior Researcher, Associate Professor, Lecturer of System Programming Department, Khmenlnitsky National University

Andriy Krasiy,

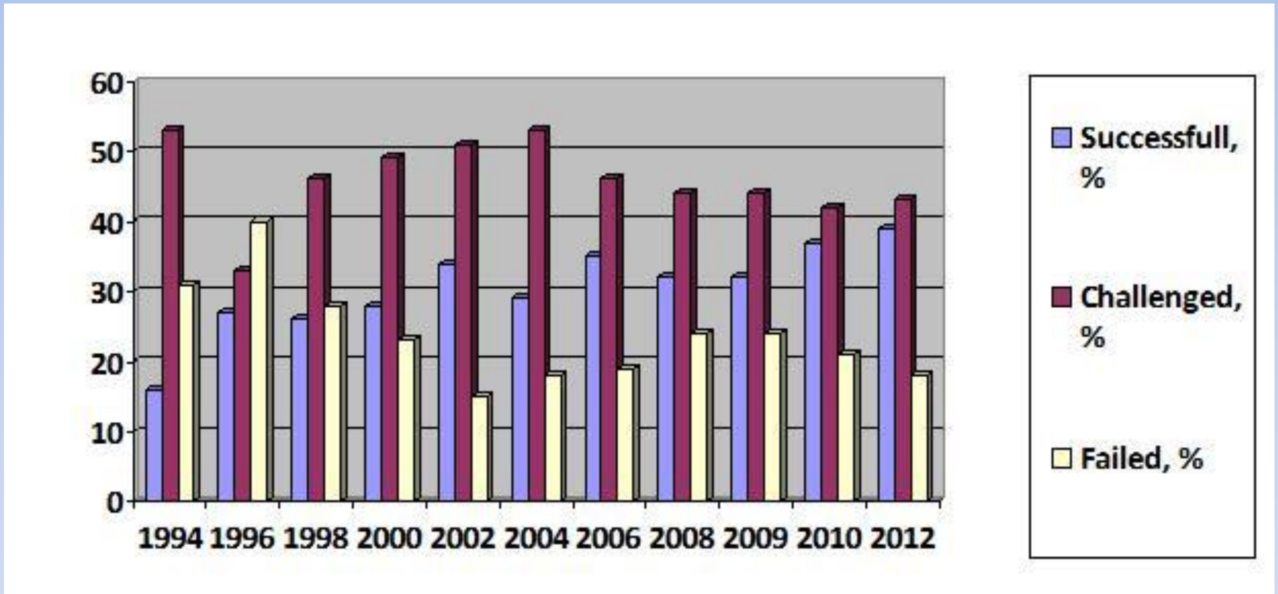
PhD student of System Programming Department, Khmenlnitsky National University

Plan of Report

1. Introduction: Actuality of Research; Research Aims and Tasks
2. Method of Evaluating the Success of Software Project Implementation Based on Analysis of Specification Using Neuronet Information Technologies (MESSPI)
3. Experiments
4. Conclusions
5. Questions & Discussion

1. Introduction: Actuality of Research; Research Aims and Tasks

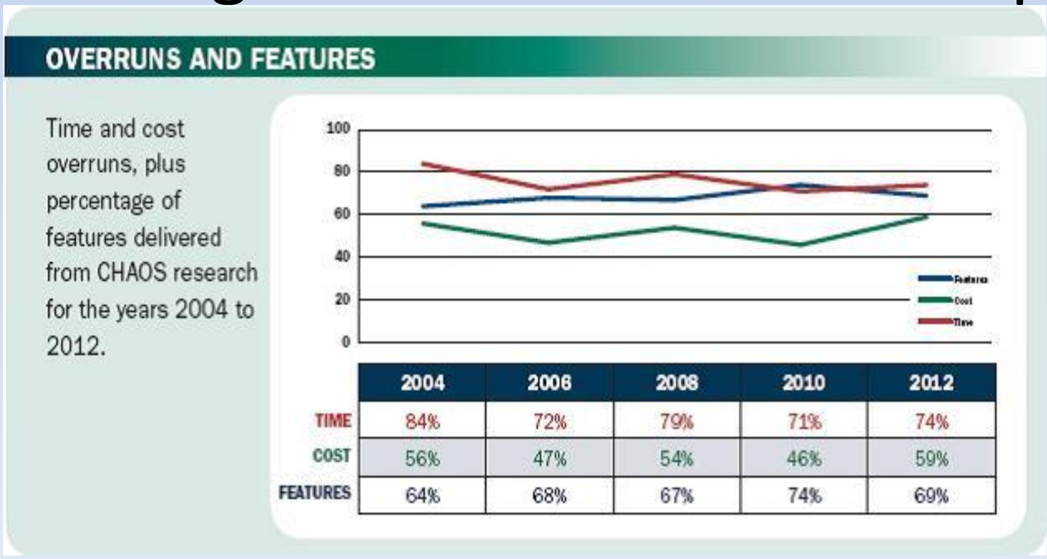
Statistics of success of software projects implementation in 1994-2012, according to The Standish Group International (CHAOS Manifesto report, 2013):



Successful are projects, that delivered on time, on budget and have required features and functions; challenged – are projects, that late, over budget, and/or with less than the required features and functions; failed – are projects, that cancelled prior to competition or delivered and never used.

Statistics of success of software projects implementation showed that the increase in the number of successful projects, and fall in the number of failed projects in 2010-2012 are, but the rate of challenged projects (that late, over budget, and/or with less than the required features) is the constant value (42-46% projects).

Statistics on time and cost overruns and percentage of features delivered of challenged software projects for 2004-2012, according to CHAOS Manifesto report:



	2004	2006	2008	2010	2012
TIME	84%	72%	79%	71%	74%
COST	56%	47%	54%	46%	59%
FEATURES	64%	68%	67%	74%	69%

These statistics reflect the high rate of non-quality (the failed and the challenged) software projects in terms of interpretation of software quality as the degree of compliance with the software characteristics of requirements.

The errors of requirements formulation are 10-25% of all errors, and the larger software volume leads to more errors at the requirements formulation stage.

The analysis of errors of embedded (built-in) and application software, which led to the famous disasters and incidents and were made at the stage of the requirements formulation, is given in next Table:

Event	Cause	Consequences
Embedded software		
In 1971 "Cosmos-419" did not start to Mars [6]	Incorrectness of SRS	Loss of device
In 1971 the station "Mars 2" could not undock from the ship [6]	Incorrectness of SRS	The task was not completed
Chinook helicopter crash in 1994 [6]	Incorrectness of SRS	29 people were killed
"Death" sessions of radiation therapy with Therac-25 [7, 8]	Incompleteness of SRS	6 patients received a lethal radiation dose
Explosion of rocket Ariane 5 in 1996 [7, 8]	Mismatch of requirements to ensure reliability and maximum allowable load	The cost of equipment and development - 7.5 billion USD, "lost profits" - 2 billion USD

Event	Cause	Consequences
Application software		
Falling rating on the exchange trading of company Dow Jones Industrial Average (1987) [7, 8]	Incorrect calculation of load on software - incorrect SRS	Loss of 500 billion USD
The failure of New York Bank system [9]	Lack of memory due to incorrect requirements	Loss of 32 billion USD
Error in the game "The Lion King" of "Disney" company (1994) [10]	Incompleteness of SRS	Loss of reputation, significant financial losses
Error Y2K - error of the second digits saving in the year of date (1999) [7, 8]	Incorrectness and incompleteness of SRS	Loss - 500 billion USD
In 1990 AT & T held the 9-hour accident [8]	Problems with boundary conditions in the SRS	75 million unrealized calls, loss of 60 million USD
In 1990 AT & T held the 9-hour accident [7]	Problems with hidden boundary conditions in the SRS	Disability of services associated with data transmission

In works of Levenson & other the fact is confirmed, that the causes of many incidents and accidents through software are in the SRS, rather than in coding. 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.

Definition 1. 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 - 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

Analysis of works of Fenton, Chen, Maedche, Fatwanto, Rehman shows, that the existing methods and tools 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 (SRS).

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 software requirements specification [IEEE 830-1998] in the next formalized form:

$$SRS=<R1,R2,R3,R4>, \qquad (1)$$

where R1 – the set of indicators of section 1 of the SRS, R2 – the set of indicators of section 2 of the SRS, R3 – the set of indicators of section 3 of the SRS, R4 – the set of indicators of section 4 of the SRS.

The **first stage of MESSPI** is prediction of characteristics of software project on the analysis of specification, that consists in the processing of sets R1-R4 of indicators and in the determining of the set of relative values of main software project characteristics:

$$SCH=\{Cs,Dsp,Cx,Cp,Ub,Qs\}, \qquad (2)$$

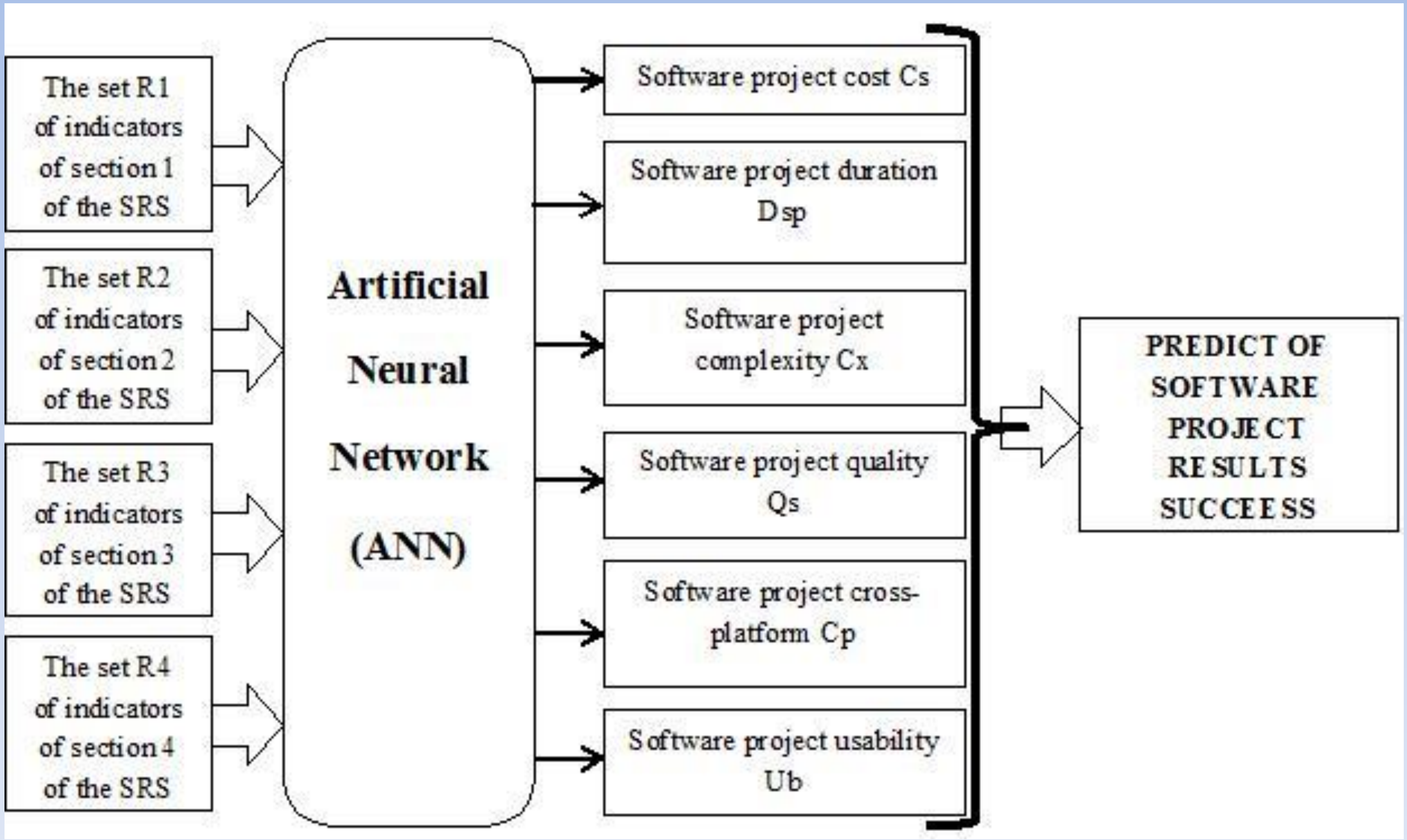
where Cs – software project cost, Dsp – software project duration, Cx – software project complexity, Cp – software project cross-platform, Ub – software project usability, Qs – software project quality.

Some indicators of specification affect the above characteristics, but **equations are 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.

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 previous works the ANN was developed, which processes and approximates the set of SRS indicators and provides the predicted quantitative values of characteristics. 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 previous works, so this information is not represented in this report.

The concept of neuronet prediction of characteristics of software project based on the analysis of specification:



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).

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

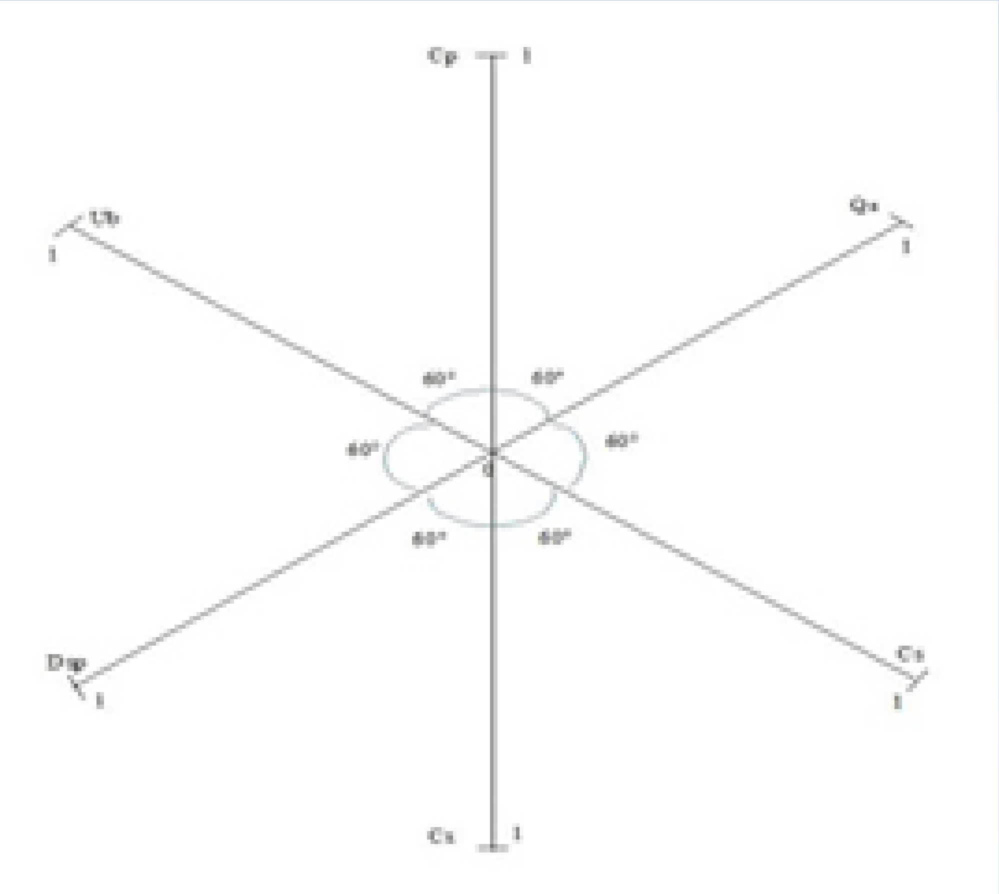
$$SCH_{ANN}=\{Cs_{ANN}, Cx_{ANN}, Dsp_{ANN}, Ub_{ANN}, Cp_{ANN}, Qs_{ANN}\} \quad (3)$$

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.

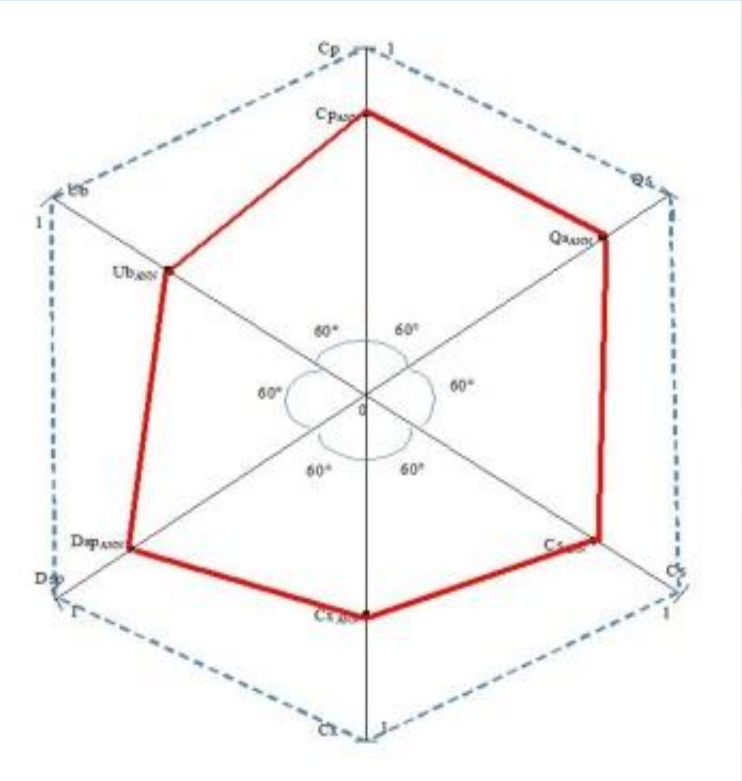
Definition 2. Integrative indicator Iip_{Sp} – is the quantitative indicator of project implementation success based on the set SCH_{ANN}.

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).



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 red line on next figure:

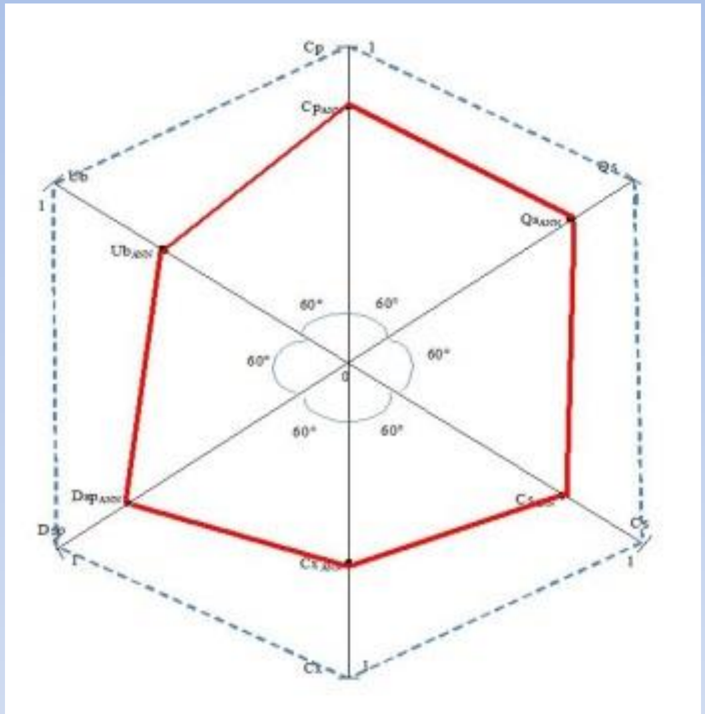


For calculation of integrative indicator Iip_{Sp} we will divide the hexagon into six triangles, will calculate the area of each triangle with two sides (value of characteris-tics) and angle between them (60°) and will add the obtained values of triangles areas:

$$S_{CsOCx}=½*Cs_{ANN}*Cx_{ANN}*\sin 60°= =0.5*0.866*Cs_{ANN}*Cx_{ANN}, \quad (4)$$

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

We will need also the maximum possible value of integrative indicator of project: Iip_{max} – is the area of hexagon CsCxDspUbCpQs highlighted dotted blue line on figure:



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:

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

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

Formalization of first – third stages of MESI will have the following form:

$$\langle R1, R2, R3, R4\rangle \rightarrow SCH_{ANN}=f_1(\langle R1, R2, R3, R4\rangle) \rightarrow lip_{Sp}=f_2(SCH_{ANN}) \rightarrow P_{lip}=f_3(lip_{Sp}, lip_{max}), \quad (8)$$

where function f_1 is realized by trained ANN, function f_2 is calculated by the formula (5), function f_3 – by the formula (7); indicator lip_{max} is calculated by the formula (6).

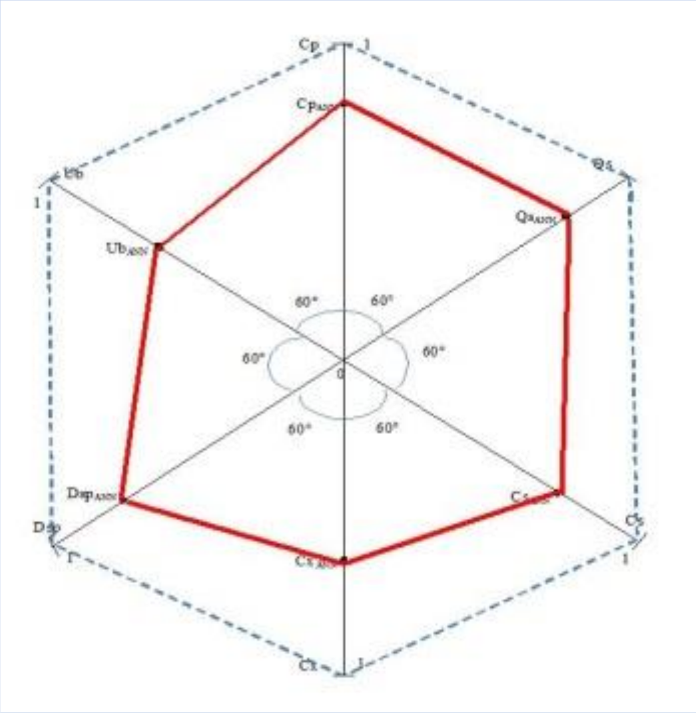
The compensation of values of the characteristics with the same value of integrative indicator is not always correct. Then **the fourth stage of MESSPI** is the testing of the stability and acceptability of characteristics compensations. **If the hexagon $Cs_{ANN}Cx_{ANN}Dsp_{ANN}Ub_{ANN}Cp_{ANN}Qs_{ANN}$** (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).

Let us recall the graphic representation of the hexagon, area of which is integrative indicator of project:



Here are the steps to determine of the angles of the hexagon:

1) calculate the unknown third side for each triangle by law of cosines:

$$CpQs^2 = Cp^2+Qs^2-2*Cp*Qs*\cos 60° \quad , \ldots \qquad (9\text{-}14)$$

2) find one unknown angles in each triangle by law of cosines:

$$(OQsCp)=\arccos([Qs^2+CpQs^2-Cp^2]/[2*Qs*CpQs]), \ldots \quad (15\text{-}20)$$

3) find second unknown angle in each triangle by theorem about the sum of angles:

$$(OCpQs)=180°-60°-(OQsCp) \quad , \ldots \qquad (21\text{-}26)$$

4) find the angles of the hexagon:

$$(Qs)=(OQsCp)+(OQsCs), \ldots \qquad (27\text{-}32)$$

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** (characteristics of software project are stable and their compensatory effects are acceptable).

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 next table:

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 given certain values of characteristics for five different software projects. We show these values and the corresponding values of integrative indicators in next table.

The data of previous table show that all five software projects have the same integrative indicator Iip_{Sp}=0.894, but have significantly different relative values of characteristics.

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

We need to check the convexity of the hexagons for all examined software projects for determination of value of indicator Ace_{Sp} :

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 research 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** 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). 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 successful 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.

5. Questions & discussion

Thank you for attention! Questions, please!

Tetiana Hovorushchenko:

tat_yana@ukr.net

+38-095-11-22-544

skype: tat_yana_13

Andriy Krasiy:

andriy-krasiy@yandex.ua

+38-097-867-14-39