Information Technology of Predicting the Characteristics and Evaluating the Success of Software Projects Implementation

Andriy Krasiy[1], Tetiana Hovorushchenko [2]

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

Abstract. The aim of this research is the development of the information technology (model, method and tools) of predicting the characteristics and evaluating the success of the software projects implementation based on the analysis of the software requirements specification (ITPCES). ITPCES structure was first time proposed. One of the non-realized components of ITPCES is the intelligent system of predicting the characteristics and evaluating the success of the software projects implementation (SPCES). This research is devoted to design of SPCES and experiments with it. SPCES gives the conclusion about probably category of success of the software project implementation by analyzing the software requirements specification (at the early stages of the life cycle). The SPCES conclusions provide to the customer the ability of the comparing the proposed software projects and provide to the customer the data for the grounded and informed choice of the most successful software project.

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

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

1 Introduction

In recent years, the software industry has reached a level of evolution where the development of software systems is user-oriented [1]. At the present definition of software quality [2], if the goals of the project don't meet the needs of users, the software will not be qualitative and successful, even if the modern technologies and the most qualified developers were involved to its development. But until now the development of successful and high-quality software products don't become the norm - statistics [3] says that in 2012 only 39% of software projects are successful, but 43%

ICTERI 2016, Kyiv, Ukraine, June 21-24, 2016

Copyright © 2016 by the paper authors

of software projects are challenged and 18% of software projects are failed, i.e. 61% software projects aren't successful and qualitative - Fig. 1.

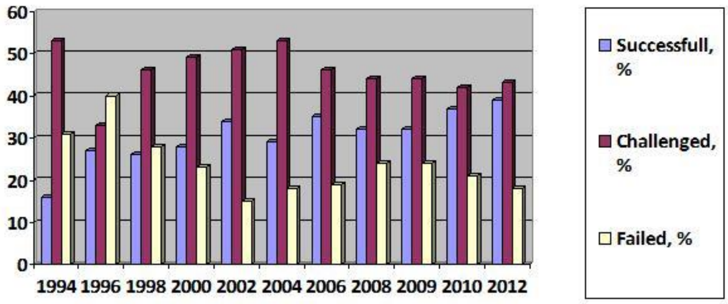


Fig. 1. Statistics of success of software projects implementation in 1994-2012

In [4-6] the fact, that almost all the causes of software incidents and accidents are latent in the software requirements specification (SRS), is confirmed. The vast majority of software accidents arises from false requirements but not from coding bugs. Software versions, written by the different developers for the same requirements, contain the number of the common bugs associated with errors or inaccuracies of requirements (SRS) [5]. Paper [7] says, that the main causes of the failure of software projects are the misconceptions of project managers on real deadline and budget for providing the user functional requirements. So paper [7] again confirmed that the most of the software problems are associated with SRS. Then the quality and success of the software project implementation depend on the SRS, resulting in the need to deepen the analysis of specifications.

Then *the actual task* is the ability to evaluate the potential success of software project implementation based on the software project characteristics (project cost, duration, complexity, usability, cross-platform, quality), the predicted values of which can be obtained by analyzing the SRS indicators. *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.

The analysis of SRS structure [8, 9] showed, that the SRS requirements provide the set of indicators, on the basis of which the customer and the developer can get the predicted quantitative values of the characteristics of software projects. For establishment of the dependence between the basic software project characteristics and SRS indicators the software requirements specifications and finished applications (realized by these SRS) were analyzed. For this analysis six types of software projects (Web-applications, mobile applications, e-learning applications, applications for statistics and accounting, automated systems, information systems) were considered. For each type of software projects 30-50 tasks of different complexity were studied. For each task 1-3 SRS (proposed by the various developers) and 1-3 finished applications (written by the analyzed specifications) were selected. For this the course projects in discipline "Technology of software systems design", the diploma papers, the projects of students scientific group «SOFTWARE» of Khmelnitsky National

University (15% of all projects; moreover only student projects, that were devoted to solving the real-world tasks and were successfully applied in different industries, was considered), the SRS and applications of the software companies of Khmelnitsky (“Avivi”, “Smile”, LLC «STU Electronics») were studied. Thus, we selected 200 tasks, for which 410 SRS and appropriately 410 applications were developed (for various industries, i.e. software projects of different types were selected) and we analyzed them: what SRS indicators differed in the selected specifications, what characteristics of finished applications changed depending on it, and what values had the SRS indicators in these SRS. The conducted analysis of finished SRS and applications led to the conclusion about dependence the basic software project characteristics on the SRS indicators for all types of software projects [10].

The analysis of the methods and tools for determination of the software projects characteristics [11, 12] led to the conclusion that they are focused on ready code, but not on existing SRS that is unusable at the early stages of the software projects life cycle. The research of the methods and tools of the SRS analysis [13-15] showed that they are aimed at monitoring the implementation of requirements and don't determine the predicted values of software projects characteristics. Thus, the existing methods and tools of SRS analysis and software project characteristics determination are not acceptable for the quantitative evaluation of the software project characteristics based on only requirements analysis and for evaluating the success of the software projects implementation.

The *task of this research* is the development of the information technology (model, method and tools) of predicting the characteristics and evaluating the success of software projects implementation based on analysis of the SRS.

2 Information Technology of Predicting the Characteristics and Evaluating the Success of Software Projects Implementation (ITPCES)

Considering the definition of information technology [16, 17], the structure of the information technology of predicting the characteristics and evaluating the success of software projects implementation is represented on Fig. 2:

Information Technology of Predicting the Characteristics and Evaluating the Success of Software Projects Implementation (ITPCES)				
Object – the success of software projects implementation	**Goal** – predicting the characteristics and evaluating the success of software projects implementation	**Model** –neuronet model of predicting the software projects characteristics based on the SRS analysis	**Method** – method of evaluating the success of software projects implementation based on analysis of SRS (MESSPI)	**Tools** – intelligent system of predicting the characteristics and evaluating the success of software projects implementation (SPCES)

Fig. 2. The structure of information technology of predicting the characteristics and evaluating the success of software projects implementation

Fig. 2 shows that the basis of ITPCES is the developed neuronet model of predicting the software projects characteristics based on the SRS analysis and method of evaluating the success of software projects implementation based on analysis of SRS (MESSPI) and also intelligent system of predicting the characteristics and evaluating the success of software projects implementation, which should be design.

The neuronet model of predicting the software projects characteristics based on the SRS analysis was developed for evaluation the software projects characteristics based on the processing of SRS indicators [11]. The basis of this model is the artificial neural network (ANN), which performs the approximation of SRS indicators and provides the predicted relative evaluation of the characteristics of the software, which will be developed by the analyzed specification. The input data for ANN are three sets of indicators: the set of indicators of section 1 of the SRS R1={Tv, Qv, Sa, Qcs, Sc}, where Tv – predicted realization time, Qv – quantity of performers, Sa – predicted quantity of users, Qcs – quantity of software components, Sc – predicted size (LOC); the set of indicators of section 2 of the SRS R2={Cos, Cdb, Cc, Cdt, Cud, Sud}, where Cos – cost of used operating systems, Cdb - cost of used databases, Cc – cost of used compilers, Cdt – cost of development tools, Cud – quantity of user documentation pages, Sud – cost of user documentation; the set of indicators of section 1 of the SRS R3={Qfr, Cfr, Qa, Cb, Cui, Qmi, Cmi, Qai, Cai, Qci, Cci, Qnfr, Cnfr}, where Qfr – quantity of functional requirements, Cfr – cost of functional requirements, Qa – quantity of algorithms, Cb – average predicted cost of bug, Cui – cost of user interfaces, Qmi – quantity of intermodule interfaces, Cmi – cost of intermodule interfaces, Qai - quantity of hardware interfaces, Cai – cost of hardware interfaces, Qci - quantity of communication interfaces, Cci – cost of communication interfaces, Qnfr – quantity of non-functional requirements, Cnfr – cost of non-functional requirements. The result of ANN functioning is the set of the predicted relative evaluation of the software project characteristics SCH={Cs, Dsp, Cx, Cp, Ub, Qs}, where Cs – software project cost, Dsp – duration, Cx – complexity, Cp – cross-platform, Ub – usability, Qs – quality [11]. These characteristics provide the comprehensively analysis of the possible success of software projects implementation – in terms not only quality of developed software products (quality, cross-platform, usability), but quality of software projects management (cost, duration, complexity). ANN was realized in Matlab, was trained with training sample of 6030 vectors by different training methods and was tested with testing sample of 610 vectors [10]. The analysis of charts of the ANN training and testing led to the conclusion that the ANN was trained with high accuracy and precision. In [10] the analysis of ANN training results (by different training functions with different performance functions) was also conducted. The performance function `msereg` and the training functions `OSS`, `SCG`, `RPROP` were selected on the basis of the following criteria: training performance, training time, and number of epochs.

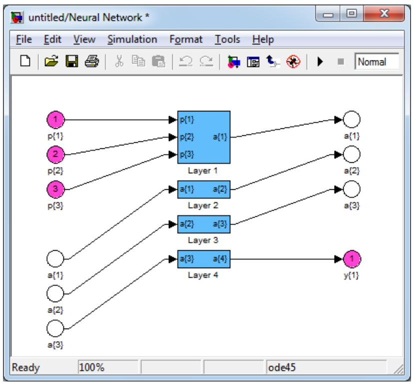
The structure chart of the ANN layers in Simulink is shown on Fig. 3.

The developed *method of evaluating the success of software project implementation based on analysis of SRS (MESSPI)* consists of the next stages [18]:

1. neuronet prediction of characteristics of software project based on the analysis of specification (the basis of which is the neuronet model of predicting the software projects characteristics based on the SRS analysis [10, 11]). The result of this stage is the set of the predicted relative evaluations of the software project characteristics SCH={Cs, Dsp, Cx, Ub, Cp, Qs}, Cs € [0..1], Dsp € [0..1], Cx € [0..1], Ub € [0..1], Cp € [0..1], Qs € [0..1], where 0 – insufficient data for prediction of the characteristics (in this case MESSPI does not work), 0.08 – characteristic negative affects on the success of software project implementation (high cost, duration, complexity, low usability, cross-platform, quality), 1 – characteristic positive impacts on the success of software project implementation (low cost, duration, complexity, high usability, cross-platform, quality);
2. interpretation of the received relative values of the software project characteristics – criteria for this interpretation is the integrative indicator of software project (Fig. 4):

   $$Iip_{Sp}=0.5*0.866*(Cs*Cx+Cx*Dsp+ Dsp*Ub+ Ub*Cp+Cp*Qs+ Qs*Cs) \quad (1)$$
3. evaluation of the degree of success of the software project implementation on the basis of the integrative indicator:

   $$P_{Iip}=Iip_{Sp}/Iip_{max}=Iip_{Sp}/2.598=0.385*Iip_{Sp} \quad (2)$$
4. testing of the stability and acceptability of compensations of software project characteristics: the indicator Ace_{Sp} of stability and acceptability of compensatory effects of the characteristics has the value “True”, if the hexagon (Fig. 4) is convex (if the sum of the angles of hexagon is 720° and sines of angles have the same signs).



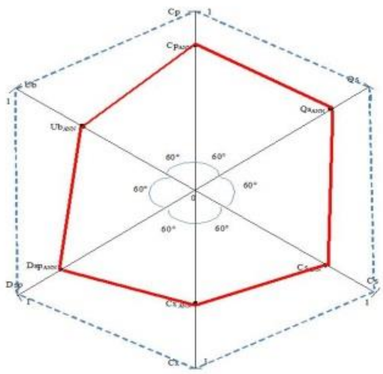


Fig. 3. The structure chart of ANN layers **Fig. 4.** The graphical representation of Iip_{Sp} and Iip_{max}

Thus, the input data for MESSPI is the set of SRS indicators, and the result of the method is the evaluation of the degree of success of the software project implementation [18], which provides to perform the reasonable choice of SRS for the further implementation of the project.

For the completion of ITPCES we need to develop *the intelligent system of predicting the characteristics and evaluating the success of software projects implementation* based on the developed method MESSPI [18].

3 Intelligent System of Predicting the Characteristics and Evaluating the Success of Software Projects Implementation

The input of the intelligent system of predicting the characteristics and evaluating the success of software projects implementation (SPCES) are the selected in [10] 24 SRS indicators, and the result of its work are: the relative values of the software project characteristics, the conclusion about stability and acceptability of compensatory effects of the software project characteristics,the integrative indicator of software project (graphical representation and value), the value of the degree of success of the software project implementation and the conclusion about category of success of software project implementation (the successful, the challenged or the failed project is expected).

The structure (algorithmic-focused vision with elements of architectural solutions) of the intelligent system of predicting the characteristics and evaluating the success of software projects implementation is represented on Fig. 5.

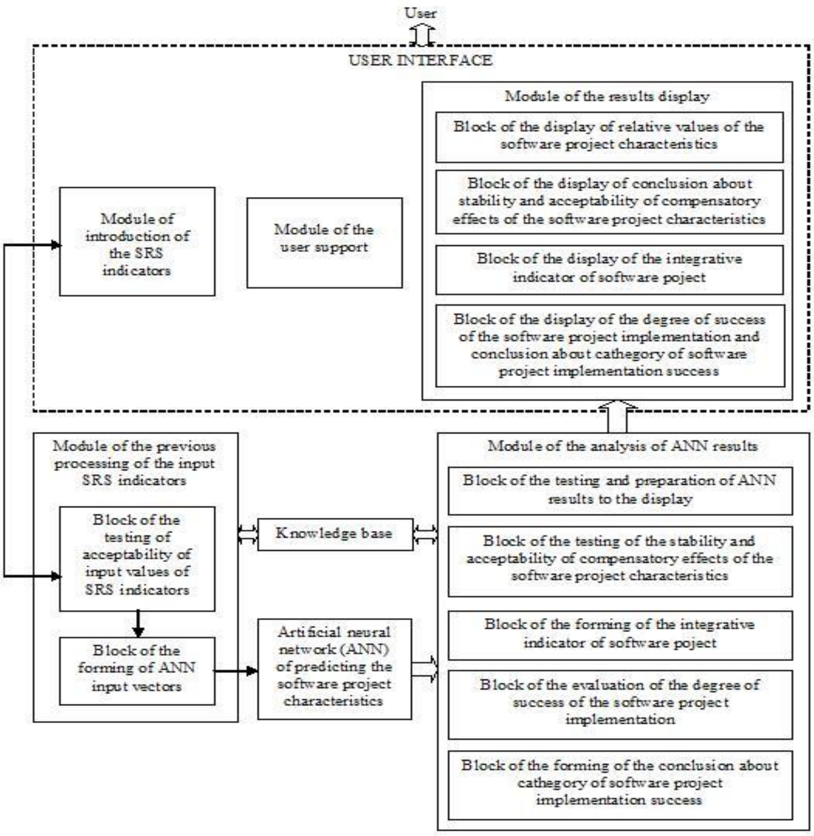


Fig. 5. The structure (algorithmic-focused vision with elements of architectural solutions) of intelligent system of predicting the characteristics and evaluating the success of software projects implementation (SPCES)

SPCES consists of the next components:

1. *module of introduction of the SRS analysis* – is the part of the user interface; reads the user information about the quantitative values of 24 SRS indicators, which are necessary for prediction of the software project characteristics;
2. *module of the user support* – is the part of the user interface; provides to the user the information about the structure of the software requirements specification, about the SRS indicators (which are required for prediction of the software projects characteristics), about the valid (for the system) ranges of the SRS indicators values (defined in [10] based on the analysis of the above-described 410 SRS), about the process of forming the results of the system (SPCES) functioning;
3. *module of the previous processing of the input SRS indicators* – tests the acceptability of the input values of the SRS indicators under the rules of the knowledge base; forms the input vector for ANN: ANN has 5 inputs x′, 6 inputs x″ and 13 inputs x; on the inputs x′ the indicators of the section 1 of the SRS are submitted, on the inputs x″ - the indicators of the section 2 of the SRS, on the inputs x – the indicators of the section 3 of the SRS under the rules of the knowledge base;
4. *knowledge base* – consists of the data section and rules section; in the data section accumulates the values of the SRS indicators and results of the system SPCES functioning; the rules section contains: rules for the testing of acceptability of input values of SRS indicators, rules for the forming of ANN input vectors, rules for the testing and preparation of ANN results to the display, rules for the testing of the stability and acceptability of compensatory effects of the software project characteristics, rules for the forming of the conclusion about category of success of the software project implementation;
5. *artificial neural network (ANN)* of predicting the software project characteristics – detailed described in paragraph 2 and in the papers [10, 11];
6. *module of the analysis of ANN results* – tests and prepares of ANN results to the display, calculates the value of indicator Ace_{Sp} of stability and acceptability of compensatory effects of the characteristics according to the 4-th stage of the method MESSPI [18] and forms the conclusion about stability and acceptability of compensatory effects of the software project characteristics, forms the graphical representation and calculates the value of the integrative indicator Iip_{Sp} of the software project according to the 2-nd stage of the method MESSPI [18], evaluates the degree P_{Iip} of success of software project implementation according to the 3-rd stage of the method MESSPI [18] and forms the conclusion about category of software project implementation success (conclusions are formed using the rules from the knowledge base);
7. *module of the results display* – is the part of the user interface; provides to the user the relative values of the software projects characteristics, the conclusion about stability and acceptability of compensatory effects of the software project characteristics, the graphical representation and the value of the integrative indicator of the software project, the degree of success of software project implementation and the conclusion about category of software project implementation success (successful, challenged or failed project is expected).

The rules for the testing of acceptability of input values of SRS indicators (are substantiated by the valid (for the SPCES) ranges of the SRS indicators values, that were defined in [10] based on the analysis of the above-described 410 SRS) have the form:

1. if Tv € [1..24] (months), then flag=true, else flag=false;
2. if Qv € [1..10] (persons), then flag=true, else flag=false;
3. if Sa € [1..1000] (persons), then flag=true, else flag=false;
4. if Qcs € [1..50] (components), then flag=true, else flag=false;
5. if Sc € [50..50000] (lines of code), then flag=true, else flag=false;
6. if Cos € [0..1250] (USD), then flag=true, else flag=false;
7. if Cdb € [0..1250] (USD), then flag=true, else flag=false;
8. if Cc € [0..1250] (USD), then flag=true, else flag=false;
9. if Cdt € [0..1250] (USD), then flag=true, else flag=false;
10. if Cud € [1..50] (pages), then flag=true, else flag=false;
11. if Sud € [50..2500] (USD), then flag=true, else flag=false;
12. if Qfr € [5..300] (requirements), then flag=true, else flag=false;
13. if Cfr € [50..4750] (USD), then flag=true, else flag=false;
14. if Qa € [1..500] (algorithms), then flag=true, else flag=false;
15. if Cb € [10..960] (USD), then flag=true, else flag=false;
16. if Cui € [50..3000] (USD), then flag=true, else flag=false;
17. if Qmi € [50..2450] (interfaces), then flag=true, else flag=false;
18. if Cmi € [25..2500] (USD), then flag=true, else flag=false;
19. if Qai € [5..100] (interfaces), then flag=true, else flag=false;
20. if Cai € [25..1500] (USD), then flag=true, else flag=false;
21. if Qci € [5..125] (interfaces), then flag=true, else flag=false;
22. if Cci € [25..1750] (USD), then flag=true, else flag=false;
23. if Qnfr € [1..9] (requirements), then flag=true, else flag=false;
24. if Cnfr € [50..4000] (USD), then flag=true, else flag=false;
25. if flag=true, then the input values of the SRS indicators are acceptable, else if flag=false the input values of the SRS indicators are not acceptable, in this case the method MESSPI and the system SPCES cannot be used for this SRS and project.

The rules for the forming of ANN input vectors (are substantiated by the quantities of the elements of the above-described sets R1-R3) have the form:

1. on the input x'_i (i=1..5) the value of i-th element of set R1 of the indicators of the section 1 of the SRS is submitted;
2. on the input x''_k (k=1..6) the value of k-th element of set R2 of the indicators of the section 2 of the SRS is submitted;
3. on the input x_j (j=1..13) the value of j-th element of set R3 of the indicators of the section 3 of the SRS is submitted;
4. if the user doesn’t enter the value of indicator, then corresponding input of ANN is -1.

The rules for the testing and preparation of ANN results to the display (are substantiated by the above-described approach to ANN training) have the form:

1. if Cs=0 or Dsp=0 or Cx=0 or Cp=0 or Ub=0 or Qs=0, then insufficient data for prediction of the software project characteristics, in this case the method MESSPI and the system SPCES cannot be used for this SRS and project;
2. output y_1 - Cs – the relative value of the software project cost, output y_2 – Dsp – the relative value of the software project duration, output y_3 - Cx – the relative value of the software project complexity, output y_4 – Ub – the relative value of the software project usability, output y_5 – Cp – the relative value of the software project cross-platform, output y_6 – Qs – the relative value of the software project quality.

The rules for the testing of the stability and acceptability of compensatory effects of the software project characteristics (are substantiated by the above-described method MESSPI) have the form:

1. if the hexagon (Fig. 4) is convex (if the sum of the angles of hexagon is 720° and sines of angles have the same signs), then the indicator Ace_{Sp} of stability and acceptability of compensatory effects of the characteristics has the value “True”;
2. if the hexagon (Fig. 4) isn’t convex (if the sum of the angles of hexagon isn’t 720° or sines of angles have the different signs), then the indicator Ace_{Sp} of stability and acceptability of compensatory effects of the characteristics has the value “False”;
3. if Ace_{Sp}=True, then the chararcteristics are stable, the compensations of characteristics are acceptable, the method MESSPI and the system SPCES are suitable for this software project and this SRS, else if Ace_{Sp}=False, then the chararcteristics are unstable, the compensations of characteristics are unacceptable, the method MESSPI and the system SPCES are not suitable for this software project and this SRS.

The degree of success of the software project implementation, which is defined under the 3-rd stage of method MESSPI, is uninformative to the developers and to the customers through the complexity and ambiguity of interpretation of its value in the predicting the category of the success of the software project. For the facilitation of the interpretation of the value of the degree of success of the software project implementation we define thresholds values of this degree, which provide the conclusion about the category of the project success. For establishment of these thresholds values (for creation of the rules for the forming of the conclusion about category of success of the software project implementation) we have analyzed 410 above-described SRS, for which the degrees P_{Iip} of success of the software project implementation were determined according to method MESSPI [18], and 410 finished applications, for which the categories of the success are known. In general, based on the proposed definition of the success of software project implementation and current reports [3], there are three categories of success of the software projects implementation: 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). The results of this analysis are shown in Table 1.

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

	Web-applications	Mobile applications	E-learning	Applications for statistics, accounting	Automated systems	Information systems
Failed software projects	$P_{Iip} \leq 0.17$	$P_{Iip} \leq 0.19$	$P_{Iip} \leq 0.15$	$P_{Iip} \leq 0.16$	$P_{Iip} \leq 0.17$	$P_{Iip} \leq 0.18$
Challenged software projects	$0.17 < P_{Iip} \leq 0.62$	$0.19 < P_{Iip} \leq 0.60$	$0.15 < P_{Iip} \leq 0.61$	$0.16 < P_{Iip} \leq 0.62$	$0.17 < P_{Iip} \leq 0.61$	$0.18 < P_{Iip} \leq 0.59$
Successful software projects	$P_{Iip} > 0.62$	$P_{Iip} > 0.60$	$P_{Iip} > 0.61$	$P_{Iip} > 0.62$	$P_{Iip} > 0.61$	$P_{Iip} > 0.59$

The rules for the forming of the conclusion about category of success of the software project implementation (considering the empirical estimates from Table 1, which in general correspond to the statistical evaluations [3] from Fig. 1) have the form:

1. if the value of the degree of success of the software project implementation $P_{Iip} \leq 0.19$, then software project is predictably failed;
2. if the value of the degree of success of the software project implementation $0.19 < P_{Iip} \leq 0.62$, then software project is predictably challenged;
3. if the value of the degree of success of the software project implementation $P_{Iip} > 0.62$, then software project is predictably successful.

4 Experiments with SPCES

The input data for the SPCES are the SRS indicators for five software projects, that were developed by the different groups of developers for the solution of the one task – the development of the automated system for large-format photo print – to the order by LLC «Deymos», Khmelnitsky (Table 2).

Table 2. The values of the indicators of five SRS, that developed by the different groups of developers for the solution of the one task

№ pr.	The set of indica-tors of section1 of SRS	The set of indica-tors of section2 of SRS	The set of indicators of section3 of SRS	
1	Tv=6, Qv=3, Sa=220, Qcs=13, Sc=10900	Cos=260, Cdb=324, Cc=216, Cdt=270, Cud=11, Sud=690	Qfr=83, Cfr=1075, Qa=108, Cb=216, Cui=705, Qmi=680, Cmi=563,	Qai=29, Cai=415, Qci=30, Cci=400, Qnfr=3, Cnfr=910
2	Tv=13, Qv=6, Sa=495, Qcs=25, Sc=26090	Cos=624, Cdb=640, Cc=648, Cdt=639, Cud=25, Sud=1329	Qfr=160, Cfr=2510, Qa=258, Cb=505, Cui=1687, Qmi=1310, Cmi=1315,	Qai=53, Cai=805, Qci=65, Cci=925, Qnfr=5, Cnfr=2110
3	Tv=18, Qv=8, Sa=770, Qcs=37, Sc=36940	Cos=249, Cdb=300, Cc=219, Cdt=283, Cud=10, Sud=650	Qfr=149, Cfr=2530, Qa=247, Cb=499, Cui=1683, Qmi=1302, Cmi=1319,	Qai=52, Cai=811, Qci=67, Cci=918, Qnfr=5, Cnfr=2100
4	Tv=24, Qv=10, Sa=1000, Qcs=50, Sc=50000	Cos=0, Cdb – not defined, Cc – not defined, Cdt – not defined, Cud – not defined, Sud – not defined	Qfr=300, Cfr=4750, Qa=500, Cb=960, Cui=3000, Qmi=2450, Cmi=2500,	Qai=53, Cai=805, Qci=65, Cci=925, Qnfr=5, Cnfr=2110
5	Tv – not defined, Qv=1, Sa – not defined, Qcs – not defined, Sc – not defined	Cos=620, Cdb=641, Cc=645, Cdt=653, Cud=27, Sud=1326	Qfr=167, Cfr=2498, Qa=262, Cb=509, Cui=1691, Qmi=1313, Cmi=1310,	Qai=56, Cai=798, Qci=62, Cci=929, Qnfr=5, Cnfr=2125

After the introduction of the SRS indicators values the module of the previous processing of the input SRS indicators saves this data in the data section of the knowledge base and tests the acceptability of the input values of the SRS indicators under the rules of the rules section of the knowledge base - if input values are not valid, then the system gives the message to the user: "The input values of SRS indicators are unacceptable, the system of predicting the characteristics and evaluating the success of software projects implementation cannot be used for such SRS". For the proposed software projects the valid values of the SRS indicators were entered, so the ANN input vector is formed for each software project. Block of the forming of ANN input vectors generates the vectors for the appropriate ANN inputs - Table 3. The ANN of predicting the software project characteristics of the project processes the input vector and gives the results (the predicted relative evaluations of the software project characteristics), that also is shown in Table 3.

Table 3. ANN input and output vectors for five software projects

№ pr.	Input 1 (x′)	Input 2 (x″)	Input 3 (x)	Output (Y)
1	[6;3;220;13; 10900]	[260;324;216; 270;11;690]	[83;1075;108; 216;705;680; 563;29;415;30; 400;3;910]	[0.789;0.782; 0.792;0.79; 0.795;0.792]
2	[13;6;495;25; 26090]	[624;640;648; 639;25;1329]	[160;2510;258; 505;1687;1310; 1315;53;805;65; 925;5;2110]	[0.518;0.521; 0.521;0.53; 0.53;0.518]
3	[18;8;770;37;369 40]	[249;300;219; 283;10;650]	[149;2530;247; 499;1683;1302; 1319;52;811;67; 918;5;2100]	[0.539;0.537; 0.542;0.533; 0.54;0.542]
4	[24;10;1000;50; 50000]	[0;-1;-1;-1;-1;-1]	[300;4750;500; 960;3000;2450; 2500;53;805;65; 925;5;2110]	[0.389;0.082; 0.39; 0.097; 0.093;0.389]
5	[-1;1;-1;-1;-1]	[624;648;648; 648;25;1329]	[167;2498;262; 509;1691;1313; 1310;56;798;62; 929;5;2125]	[0.68;0.522; 0.681;0.52; 0.52;0.68]

Block of the testing and preparation of ANN results to the display tests the ANN results – if the value of even one ANN output is 0, the system gives the message to the user: "The data for predicting the software project characteristics are insufficient, so the system of predicting the characteristics and evaluating the success of software projects implementation cannot be used for such SRS". For the proposed software projects the input data were sufficient, so the ANN results were prepared to the display according to the above rules.

Block of the testing of the stability and acceptability of compensatory effects of the software project characteristics calculates the indicator Ace_{Sp} of stability and acceptability of compensatory effects of the characteristics (Table 4) and forms the conclusion about stability and acceptability of compensatory effects of the software project characteristics. If Ace_{Sp} = False, the user gets the message: "The software projects characteristics are unstable, the compensations of characteristics are unacceptable, so the system of predicting the characteristics and evaluating the success of software projects implementation is not suitable for this project and for this SRS". If Ace_{Sp} = True, then the system calculates the integrative indicator of software project, the degree of success of the software project implementation and forms the conclusion about category of software project implementation success. For the proposed software projects №1, №2, №3, №5 the characteristics are stable, the compensations of characteristics are acceptable, so the obtained predicted relative values of the characteristics are processed according to the method MESSPI. For the proposed software project №4 the system gives the message to the user: "The software projects characteristics are unstable, the compensations of characteristics are unacceptable, so the system of predicting the characteristics and evaluating the success of software projects implementation is not suitable for this project and for this SRS".

Block of the forming of the integrative indicator of software project forms the graphical representation and calculates the value of integrative indicator Iip_{Sp} of software project (Table 4).

Block of the evaluation of the degree of success of the software project implementation estimates the value of the degree of success P_{Iip} of the software project implementation (Table 4).

Block of the forming of the conclusion about category of software project implementation success uses the rules of the knowledge base and forms the conclusion about category of software project implementation success (Table 4).

Table 4. The values of the indicator Ace_{Sp} of stability and acceptability of compensatory effects of the characteristics, integrative indicator Iip_{Sp}, the degree of success P_{Iip} of the software project implementation and the conclusion about category of software project implementation success for five above software projects

№ pr.	Ace_{Sp}	Iip_{Sp}	P_{Iip}	Conclusion about category of software project implementation success
1	True	1,621	0,624	Successful project
2	True	0,711	0,274	Challenged project
3	True	0.754	0.290	Challenged project
4	False	-	-	The software projects characteristics are unstable, the compensations of characteristics are unacceptable, so this system is not suitable for this project and for this SRS
5	True	0.942	0.363	Challenged project

Let's analyze of the results: the category of software project implementation success was defined for the projects №1, №2, №3, №5, for which the predicted values of characteristics are stable, their compensations are acceptable. The software project №1 has the best characteristics, it predictably belongs to the category of successful projects. The software projects №2, №3, №5 have the worst characteristics and predictable are classified as challenged projects. For the project №4 the system SPCES cannot determine the category of implementation success because the predicted characteristics are unstable and their compensations are unacceptable. So the conclusion of the intelligent system of predicting the characteristics and evaluating the success of software projects implementation recommends to LLC "Deymos" to order the implementation of software project №1 (to the development of the automated system for large-format photo print) that will be successful with the greatest probability.

Nowadays the developer and customer select the software project based on only own intuition and the cost and duration that predicted in the SRS. But SRS developers cannot always correctly predict the oriented cost and duration of software project during development of the SRS. Predicted (in the SRS) values of cost and duration for the four examined alternative software projects (for which SPCES determines the category of implementation success) are represented in Table 5.

Table 5. Predicted (in the SRS) values of cost and duration for the four software projects

Characteristics of software project	Values for Project №1	Values for Project №2	Values for Project №3	Values for Project №5
Predicted cost, specified in the SRS	11875 USD	11125 USD	10625 USD	10812 USD
Predicted duration, specified in the SRS	6 months	13 months	18 months	Not defined

The values of the characteristics of software projects from Table 5 show, that all four software projects have the different duration but the same cost that predicted in the SRS. But the results of Table 3 show, that projects have significantly different relative values of all characteristics, including the cost, which were calculated taking into account all significant SRS indicators. Thus, the relative cost ranges from 0.518 (for Project №2) to 0.789 (for Project №1). So, if we evaluate the cost, taking into account all significant SRS indicators, then it values are not the same for the four examined projects. As for the value of project duration, this value isn't defined for project №5, for example, then software project in this case will be evaluated solely on the basis of its cost value. Therefore, the customer and developer can make the wrong conclusion about choice of project on the basis of solely cost and time that predicted in the SRS. In addition, such conclusion is difficult in the real conditions. For example, according to Table 5, the lowest cost has the software project №3, and the lowest duration has the software project №1 (but the value is unknown for the software project №5), i.e. the customer and the developer must make the choice of software project in this case on the basis of one criterion - or by cost, or by the duration.

In addition, the success of software projects implementation depends not only on the cost and duration, but also on the functionalities of developed software, i.e. on the rest of the main characteristics of the software project - complexity, usability, cross-platform and quality, which aren't defined in the SRS explicitly in the quantitative form. In addition, Table 4 shows that the examined software projects have the different category of software project implementation success. Therefore, the values of main characteristics, provided by ANN and the conclusions of SPCES about the category of software project implementation success will help to make the right choice and to implement the software project which will be successful with the greatest probability (among from four examined software projects is Project №1). But if the developer and the customer made the choice of the software project on the basis of the only duration, they probably would choose the project №3, which really has a low degree of success of the implementation and with the high probability will be challenged software project (wrong choice).

5 Conclusions

In the article the structure of information technology of predicting the characteristics and evaluating the success of software projects implementation (ITPCES) is first time proposed. The basic components of ITPCES are the previously developed by the author the neuronet model of predicting the software projects characteristics and the method of evaluating the success of software projects implementation based on analysis of SRS and also (yet not developed) the intelligent system of predicting the characteristics and evaluating the success of software projects implementation, to the designing of which this research is dedicated.

The structure of the intelligent system of predicting the characteristics and evaluating the success of software projects implementation (SPCES) are proposed. SPCES consists of the next components: module of introduction of the SRS analysis; module of the user support; module of the previous processing of the input SRS indicators; knowledge base; artificial neural network (ANN); module of the analysis of ANN results; module of the results display. This system gives the conclusion about the probably category of success of the software project implementation based on analysis of the SRS (at the early stages of the life cycle).

The practical significance of the proposed information technology ITPCES is this fact, that system's conclusions about the category of the success of software project implementation provide to the customers the comparison of the proposed software projects and the data for the reasoned and informed choice of the most successful software project (not just on the basis of the project cost and duration, as is currently).

The authors' following perspective for future researches are: 1) development of DEF0-block diagram and UML component/deployment diagrams for the SPCES; 2) realization of the intelligent system of predicting the characteristics and evaluating the success of software projects implementation for prediction of characteristics and evaluation of success of software project implementation based on analysis of the

SRS; 3) realization of the information technology of predicting the characteristics and evaluating the success of software projects implementation.

References

1. ISO 10002:2014 Quality management – Customer satisfaction – Guidelines for complaints handling in organizations (2014)
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. The Standish Group International: CHAOS Manifesto – Think big, act small. Technical report, CHAOS Knowledge Center (2013)
4. Levenson, N.G.: Systemic factors in software-related spacecraft accidents. In: AIAA Space Conference and Exposition, pp.1-11 (2001)
5. Levenson, N.G.: Software challenges in achieving space safety. Journal of the British Interplanetary Society. 62, 265-272 (2009)
6. 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)
7. Yourdon, E.: Death March: The complete software developer's guide to surviving "Mission impossible" projects: 2nd edition. Prentice Hall (2003)
8. IEEE 1031-2011 – IEEE Guide for the Functional Specification (2011)
9. IEEE 29148-2011 - Systems and software engineering - Life cycle processes - Requirements engineering (2011)
10. Krasiy, A.: Research of artificial neural network's component of method of characteristics predicition and evaluating the success of software projects implementation. Transactions of Khmelnitsky National University. 1, 24-31 (2016) [in Ukrainian]
11. 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]
12. Fenton, N.: Software metrics: A rigorous approach (3rd edition). CRC Press (2014)
13. Chen, A., Beatty, J.: Visual models for software requirements. MS Press, Washington (2012)
14. Fatwanto, A.: Software requirements specification analysis using natural language processing technique In: International Conference on Quality in Research, pp.105-110 (2013)
15. 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)
16. Leon, A., Leon, M.: Fundamentals of Information Technology: 2nd Edition. Vikas Publishing (2009)
17. Korneev, I., Ksandopulo, G., Mashurtsev, V.: Information Technologies: Tutorial. Publishing house "Prospekt" (2009) [in Russian]
18. Hovorushchenko, T., Krasiy, A.: Method of Evaluating the Success of Software Project Implementation Based on Analysis of Specification Using Neuronet Information Technologies. Proceedings of the 11th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer. CEUR-WS. 1356, 100-107 (2015)