THE INTELLIGENT DECISION SUPPORT SYSTEM FOR CHOICE OF SOFTWARE PROJECT

Oksana POMOROVA, Tetyana HOVORUSHCHENKO

Khmelnitsky National University, Faculty of Computer System and Programming, Ukraine

e-mail: o.pomorova@gmail.com, tat_yana@ukr.net

Abstract

The analysis of the software evaluation methods was conducted. The perspective research direction to improve the software quality is development of intelligent systems (IS). IS will analyze and process the design stage metrics. Also IS will evaluate the project and provide the prediction of the characteristics of designed software. The developed intelligent decision support system for choice of software project is named as Intelligent System of Software Complexity and Quality Evaluation and Prediction (ISCQEP). ISCQEP is designed for the evaluation of design stage results and prediction of software complexity and quality characteristics on the basis of processing of design stage metrics with exact and predicted values. The conclusions about project and designed software complexity and quality are the results of the system functioning. Today the main parameters in the choice of software project are design cost, design time and designing company reputation. But the decisions on the basis of these parameters are not always guarantee the proper software quality. ISCQEP conclusions allow comparing the different project versions, when design cost and design time is approximately equal. The proposed IS provides the motivated and grounded decision about choice of software project taking into account the complexity and quality of project and designed software.

Keywords**:** *software, software metric, software metric at the design stage, software quality, software complexity, Safety Case methodology, artificial neural network.*

1 INTRODUCTION

1.1 Safety Case Methodology

Safety Case methodology (Safety computer-aided software engineering) is developed over 20 years [1]. The primary object of Safety Case methodology is to minimize the software security and commercial risks by constructing a report, which

should provides evidences, reasons and arguments that software is safe, and all requirements for the software is properly implemented. Currently, this methodology is generally accepted, but the level of its automation is still low.

The process of software developing for the Safety Case methodology depends on a large number of documents (requirements, standards, project specification), source code, software evaluation methods and analysis of their results, software testing results and the degree of its documentation.

The basic parts of Safety Case model: 1) software requirements profile - functional, inverse and non-functional requirements for software systems are analyzed; requirements for software safety and security are researched; completeness of all kinds of requirements is estimated; 2) software analysis results profile - *metric analysis results*, source code and software test results are analyzed; 3) evaluation of obtained software (results profile) accordance to its requirements (requirements profile).

1.2 Analysis of Software Metrics

One of the main tools of analysis and evaluation of software quality is metric analysis. Software metric provides obtaining of numerical values of some software properties or its specifications.

In spite of numerous researches of software metrics many unsolved tasks remain in this domain: 1) the lack of unified standards for metrics - over a thousand metrics were created, each developer of "measurement" system offers own measures of software quality evaluation and proper metrics; 2) difficult interpretation of metrics values - for most users the metrics and their values are not informative; 3) quality measurement technology has not yet reached maturity - only 1,5% software companies are trying to evaluate the quality of processes and ready product using metrics, and only 0,5% software companies are trying to improve their work on the basis of quantitative criteria of software quality; 4) at the software design stage the most attention at the project choice is paid to design cost, design time, software company reputation and software engineering technologies, but the decisions, taken on the basis of these parameters, are not guarantee software quality.

1.3 Task Formulation

The analysis of the software evaluation methods was conducted. The perspective research direction to improve the software quality is development of intelligent systems (IS). IS will analyze and process the design stage metrics. Also IS will evaluate the project and provide the prediction of the characteristics of designed software. IS conclusions allow comparing the different project versions, when design cost and design time of several projects is approximately equal. So IS will provide the motivated and grounded decision about choice of software project taking into account the complexity and quality of project and designed software.

2 SOFTWARE DESIGN STAGE METRICS

Analysis of software quality metrics in terms of possible of their application at the design stage was conducted [2, 3]. The software metrics, values of which can calculate exactly or approximately at the design stage, were selected from the set of basic metrics. 9 software metrics of design stage with the exact values and 15 software metrics of design stage with the predicted values (Table 1) were chosen [2, 3] as the basic metrics for processing by intelligence system of software complexity and quality evaluation and prediction (ISCQEP).

Table 1

№	Design Stage Metrics with the Exact Values		Design Stage Metrics with the Predicted Values	
	Complexity Metrics	**Quality Metrics**	**Complexity Metrics**	**Quality Metrics**
1	Chepin's metric	Cohesion metric	Expected Lines Of Code	Software design total time
2	Jilb's metric (absolute)	Coupling metric	Halstead's metric	Design stage time
3	McClure's metric	Metric of the global variables calling	McCabe's metric	Software design expected cost (USD)
4	Kafur's metric	Time of models modification	Jilb's metric (logical)	Software quality audit expected cost (USD)
5		Quantity of found bugs during the models inspection	Expected quantity of program statements	Software realization productivity (minutes for 1 line)
6			Expected estimate of interfaces complexity	Program code realization expected cost (USD)
7				Expected functional points FP
8				Effort applied by Boehm's model (man-months)
9				Expected development time by Boehm's model (months)

3 THE STRUCTURE OF ISCQEP

ISCQEP is developed for evaluation of design stage results and prediction of software complexity and quality characteristics. These estimates are formed on the basis of processing exact and predicted values of design stage metrics. Quantitative values of design stage metrics are given to the ISCQEP inputs. The conclusions about complexity and quality of the project and designed software are the results of the system functioning. Structure of ISCQEP is represented on Fig.1.

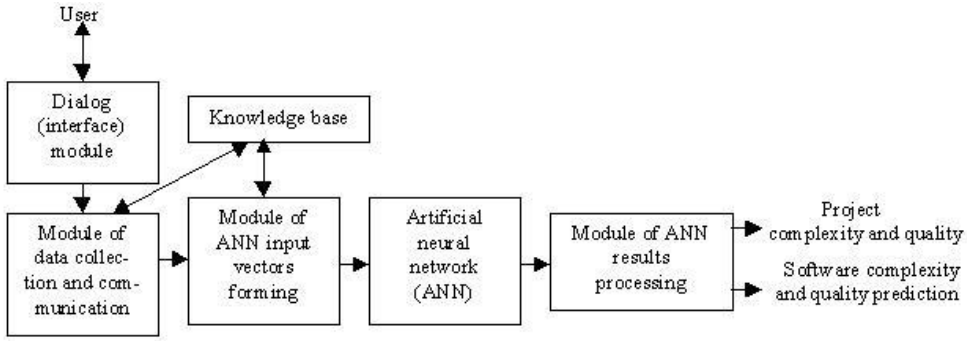

Figure 1-ISCQEP Structure

ISCQEP consists of next modules: interface module; module of data collection and communication; knowledge base; artificial neural network (ANN); module of ANN input vectors forming; module of ANN results processing.

The *interface module* visualizes the functioning of module of data collection and communication, displays the system functioning and publishes the messages to user. The *module of data collection and communication* reads the user information about the quantitative values of metrics of software design stage, saves this information in the knowledge base and transmits its to the module of ANN input vectors forming. *Knowledge base* contains the quantitative values of metrics of software design stage, the ANN input vectors and the rules of ANN results processing.

The *artificial neural network (ANN)* provides the approximation of software metrics of design stage. ANN evaluates the complexity and quality of software project. Also ANN predicts the complexity and quality characteristics of designed software. Input data for ANN are the set of the design stage metrics with the exact values $TMP = \{tmp_a \mid a = 1..9\}$ and the set of the design stage metrics with the predicted values $PMP = \{pmp_b \mid b = 1..15\}$. If a certain metric was not determined, the proper element of set is equal -1.

The *module of ANN input vectors forming* prepares the metrics values from the knowledge base for the ANN inputs. ANN has 9 inputs x' and 15 inputs x. The values of exact metrics are given on inputs x', and the values of predicted metrics are

given on inputs x. The value of i-th TMP element is the value of input x_i' $(i=1..9)$, the value of j-th PMP element is the value of input x_j $(j=1..15)$.

Multilayer perceptron is ANN for solving of task of the metrics analysis and prediction of software quality characteristics [2]. This ANN has 24 neurons of the input layer, 14 neurons of approximating layer, 8 neurons of the adjusting layer and 4 neurons of the output layer. The ANN layers structural scheme is shown on Fig.2.

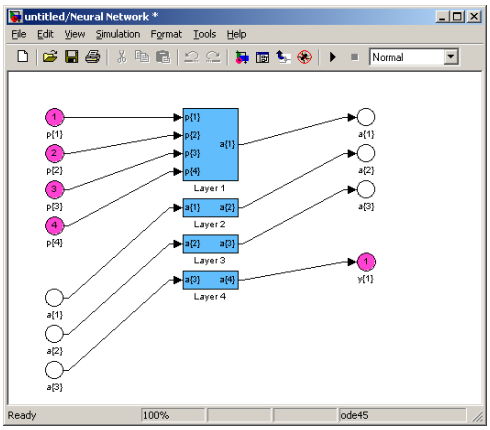

Figure 2-ANN Layers Structural Scheme in Simulink

Realized neural network was trained with training sample of 1935 vectors by one step secant backpropagation method (OSS). The training performance is $\xi = 0{,}102197$. ANN was tested with testing sample of 324 vectors.

The ANN processes the set of input vectors and gives 4 output values from the range [0, 1]: OSP - project complexity estimate; OQP - project quality evaluation; $PSPZ$ - designed software complexity prediction; $PQPZ$ - designed software quality prediction.

The *module of ANN results processing* makes the conclusions about the project quality and complexity and the expected quality and complexity of designed software on the basis of the following rules: 1) if $OSP = 0$, then exact complexity metrics were not determined; 2) if $OSP \to 0$, then the project is complicated to realization; 3) if $OSP \to 1$, then the project is simple to realization; 4) if $OQP = 0$, then exact quality metrics were not determined; 5) if $OQP \to 0$, then project is a low quality; 6) if $OQP \to 1$, then the project satisfies the customer requirements in quality; 7) if $PSPZ = 0$, then predicted complexity metrics were not determined; 8) if $PSPZ \to 0$, then designed software will has significant complexity; 9) if $PSPZ \to 1$, then simple software is expected; 10) if $PQPZ = 0$, then predicted quality metrics were not determined; 11) if $PQPZ \to 0$, then designed software will has low quality; 12) if $PQPZ \to 1$, then high quality software is expected.

4 EXAMPLE OF FORMING OF CONCLUSIONS ABOUT PROJECT AND SOFTWARE COMPLEXITY AND QUALITY BY ISCQEP

The metric analysis results of 3 projects (Table 2) of software company "STU-Electronics" (Khmelnitsky, Ukraine) will be used as the ISCQEP input data.

Table 2

№ pr.	Stage Design Metrics with the Exact Values		Stage Design Metrics with the Predicted Values	
	Complexity Metrics	**Quality Metrics**	**Complexity Metrics**	**Quality Metrics**
(1)	(2)	(3)	(4)	(5)
1	Chepin's metric - 22750 Jilb's metric (absolute) - 1730 McClure's metric - 84050 Kafur's metric - was not determined	Cohesion - 3 Coupling - 7 Metric of the global variables calling - was not determined Time of models modification - 33 Quantity of found bugs during the models inspection - was not determined	Expected LOC - 35300 Halstead's - was not determined McCabe's - 1680 Jilb's (logical) - 0,7 Quantity of program statements - 35000 Interfaces complexity - was not determined	Total time - 364 Design time - 127 Design cost - 17500 Quality audit cost - was not determined Productivity - was not determined Realization cost - 6125 Expected FP - 2100 Effort applied - 283 Development time - was not determined
2	Chepin's metric - 16250 Jilb's metric (absolute modular complexity) - 1250 McClure's metric - 60050 Kafur's metric - 257000	Cohesion - 5 Coupling - 4 Metric of the global variables calling - 0,5 Time of models modification - 24 Quantity of found bugs during the models inspection - 2500	Were not determined	Were not determined
3	Were not determined	Were not determined	Expected LOC - 10800 Halstead's - 312501 McCabe's - 480 Jilb's (logical) - 0,2	Total time - 104 Design time - 37 Design cost - 5000 Quality audit cost - 500 Productivity - 1,1

(1)	(2)	(3)	(4)	(5)
			Quantity of program statements - 10000 Interfaces complexity - 0,2	Realization cost - 1750 Expected FP - 560 Effort applied - 80 Development time - 5

After receiving of metric analysis data the module of data collection and communication transmits this information to the module of ANN input vectors forming and saves in the knowledge base.

ANN processes the input vectors and gives the results. On the basis of these results and rules from knowledge base the module of ANN results processing makes certain conclusions (Table 3).

Table 3

Project	**Value *OSP* and ISCQEP conclusion**	**Value *OQP* and ISCQEP conclusion**	**Value *PSPZ* and ISCQEP conclusion**	**Value *PQPZ* and ISCQEP conclusion**
1	0.32	0.29	0.33	0.31
	The project is sufficiently complicated to realization	The project has low quality	The designed software will has significant complexity	The designed software will has low quality
2	0.50	0.51	0	0
	The project has medium complexity	The project has medium quality	The predicted complexity metrics were not determined	The predicted quality metrics were not determined
3	0	0	0.83	0.80
	The exact complexity metrics were not determined	The exact quality metrics were not determined	The designed software will has significant complexity	High quality software is expected

The data in Table 3 shows, that for the project 2 the predicted complexity and quality metrics were not determined, therefore ISCQEP can not make the conclusion about complexity and quality of designed software. For the project 3 the exact complexity and quality metrics were not determined, therefore ISCQEP can not evaluate the complexity and quality of software project. The research of project 1 and other projects (wih all groups metrics) shows that the evaluations of quality and complexity of project and designed software are approximately equal. So the project 3 is the simplest and most high-quality project among the projects from Table 3. The software designed for project 3 will be a simple and high quality.

5 ISCQEP FUNCTIONING RESULTS USING

Today the main parameters in the choice of software project are the design cost, design time and designing company reputation, but the decisions on the basis of these parameters are not always guarantee the proper software quality. ISCQEP conclusions allow comparing the different projects, when the cost and time is approximately equal. For example, data about two projects of software company "STU-Electronics" (Khmelnitsky, Ukraine) are considered in Table 4.

Table 4

Project	Values Y_1, Y_2	Values Y_3, Y_4	Design Cost	Design Time
1	Y_1=0,85; Y_2=0,86	Y_3=0,81; Y_4=0,87	12500 USD	250 working days
2	Y_1=0,22; Y_2=0,25	Y_3=0,28; Y_4=0,27	13125 USD	260 working days

The characteristics of projects 1 and 2 (Table 4) evidence that both projects have approximately the same design cost and time, but significantly different estimates of complexity and quality of project and designed software. On the basis of only cost and time software company can make the false choice of software project. ISCQEP conclusions help to make the right choice in favor of project 1. Project 1 (Table 4) has the best complexity and quality characteristics among two proposed projects.

Acknowledgement: The necessity and actuality of scientific research in domain of software quality evaluation and prediction follow from the results of the analysis of methods of software metric evaluation. The proposed intelligent system of software complexity and quality evaluation and prediction provides the motivated and grounded decision about choice of software project taking into account the complexity and quality of project and designed software.

This work has been supported by the grant 158886-TEMPUS-2009-UK-JPCR "National Safeware Engineering Network of Centres of Innovative Academia-Industry Handshaking".

REFERENCES

[1] P. BISHOP, R. BLOOMFIELD: A Methodology for Safety Case Development. http://www.adelard.com/papers/sss98web.pdf - Feb. 1998.

[2] O.POMOROVA, T.HOVORUSHCHENKO, O.ONISHCHUK: Evaluation of Design Results and Prediction of Software Quality Characteristics" // Transactions of Khmelnitsky National University, N2, 2011 - pp.168-178 http://library.tup.km.ua/pdf/visnyk_tup/2011/(174)%202011-2-t.pdf

[3] O.POMOROVA, T.HOVORUSHCHENKO: The Intelligent Method of Design results Evaluation and Software Quality Characteristics Prediction // Radioelectronic and Computer Systems, N6, 2010 - pp.211-218 http://nbuv.gov.ua/portal/natural/Rks/2010_6/Pomorova.pdf