UDC 004.9:004.41 **doi: 10.32620/reks.2022.2.06**

Amrita SHARMA, Neha CHAUDHARY

Manipal University Jaipur, Jaipur, India

THE COMBINED MODEL FOR SOFTWARE DEVELOPMENT EFFORT ESTIMATION USING POLYNOMIAL REGRESSION FOR HETEROGENEOUS PROJECTS

Subject matter*: Estimating the software work is a crucial job of persons participating in software project management. The difficulty in predicting effort is compounded by the fact that software development is always changing. In the past, researchers used one form of development methodology in their work to estimate effort and time. Estimations of the software projects are estimated with different size matrices. The lines of code, story point and use case point are required for the estimation using algorithmic models for procedural, agile, and object-oriented development approaches. Currently, the companies use these three types of size matrices for estimating projects. Not any one model present estimates the effort for different development approaches with different size metrics. This paper proposes a combined software estimation model for three types of development methodologies with regression analysis. The estimation can be done with the proposed model for a software project developed using the procedural, agile, and object-oriented approach.* ***Method****: The input for the model is the size of the software, such as lines of code, story point, and use case point. The model is developed using the polynomial regression. The model is developed with the four constant parameters that are based on the procedural, agile, and object-oriented projects. A dataset of python projects for procedural, zia dataset for agile, company dataset for object-oriented methodology is used to propose the model.* ***Conclusion****: The effort is predicted for the procedural, agile, and object-oriented projects with the polynomial regression model and compare the results to existing models to validate the work. The R^2 is used to measure accuracy and the MMRE is used to determine error. The accuracy of the proposed model was higher than 90% and the error was found to be less than 0.05. The results are compared with case-based reasoning and an ensemble model for the procedural approach, linear regression and Bayesian network for the agile approach, and linear and log-linear regression for object-oriented approach. The minimum error and maximum accuracy is achieved compared to these techniques.*

Keywords: *software cost estimation; lines of code; user story point; use case point; polynomial regression; combined mode.*

Introduction

In software engineering, an effort is used to quantify workforce usage and is defined as the total time spent by members of a development team to perform a task. Effort estimation is the process of estimating how much effort will be required to develop or maintain a software application. This effort is usually measured in terms of the number of hours a person works or the amount of money required to pay for the work. Effort estimating is used in the early phases of the software development life cycle to aid in the creation of project plans and budgets. This strategy can be used by a project manager or product owner to accurately forecast cost and assign resources.

Although effort estimate can be applied to any software development approach, it is most commonly associated with Agile. The product owner is in charge of keeping track of project deliverables. They'll figure out how long it'll take them to complete each task. Instead than looking at time or cost estimates, they'll focus on user stories and story points. Except the individual member of team, agile require entire team for software development. In the agile approach, the workload is assigned to the entire team except for the individual member of the team. A team of best specialists require to provide the project in a decided scope and budget [1].

A widely acknowledged and widely used technique for capturing the business processes and requirements of a software application project is use case modelling. Because use cases define the project's functional scope, assessing their contents provides significant insight into the time and resources required to develop and implement a project. The UCP technique, which is based on Gustav Karner's work [2], abstracts the use case actors, scenarios, and other technological and environmental elements into an equation.

Software estimation approaches have improved during the previous three decades. The bulk of software effort estimation tools, approaches, and model-based procedures rely on historical data from earlier completed projects to generate a mathematical formula for estimating software expenditures. COCOMO [3], SEER-SEM, PUTNAM's model, PRICE-S, SLIM, [4], agile model [5], use case model [6], and other researchers have discussed a number of software effort estimating models. These models leverage size of the software as the primary

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

input for estimating software cost, along with other elements and characteristics including personal and project attributes, complexity, environmental factors, and so on. Learning-oriented models [7] are based on past estimating experience. The models are created by using the prior estimating techniques to train them. Artificial intelligence, neural networks [6], machine learning [8], case-based reasoning [9], optimization approaches [10], fuzzy logic [11], and other learning-oriented techniques are examples. When there is a lack of quantifiable and empirical data, model-based techniques come in handy. The quantifiable and empirical facts, on the other hand, are not necessary for the expert-based approach [12]. This method calculates the software cost of a new project by comparing it to a similar project that was completed previously. The analogy-based method leverages data from a before finished project to estimate software costs [13]. In software engineering, a multitude of development approaches are now used to create software. The estimations of these strategies are based on a variety of input parameters. These development approaches employ several size estimation matrices. To improve the accuracy of effort estimation, researchers applied a number of machine learning methodologies and evolutionary algorithms.

1. Related work

Sequential development approach: Shahpar et al. [14] used the PSA-SA (Particle swarm optimization-Simulated annealing) approach for feature weighting in their analogy-based estimation for software work estimation. On the Albrecht dataset, the performance of this approach was compared to that of the MMRE, MdMRE, and PRED. A Taguchi-based artificial neural network with two different activation functions was proposed by Nevena et al. [15]. In this study, six different datasets based on lines of code were used. The input values are applied using the clustering method. The results were validated using the mean magnitude relative error. This work minimizes the execution time by executing a small number of iterations. Support vector machine (SVM), Multilayer Perceptron (MLP), and Generalized Linear Models (GLM) are three machine learning algorithms suggested by Pospieszny et al. [16]. (GLM). Using the ISBSG dataset, the described approaches are utilized to estimate software effort and duration. The author validates and compares their previous work to the MMRE and PRED, concluding that the SVM and ensemble model generate superior results.

Agile development approach: Panda et al. [17] employ the General Regression Neural Network (GRNN), the Probabilistic Neural Network (PNN), the Cascade-Correlation Neural Network (CCNN), and the Group Method of Data Handling (GMDH) Polynomial Neural Network to evaluate the effort in agile development. These neural networks were taken into account in the user narrative point analysis. The cost of software development was calculated using neural network models. The neural networks were compared based on a few common factors. Based on story size, velocity, narrative complexity, friction, and dynamic aspects, Zia et al. [5] estimated development time and cost using an agile methodology dataset, resulting in lower MMRE values for time and cost. Khuat and Le [18] presented a hybrid algorithm using the PSO and ABC to improve accuracy by suggesting the parameters of an estimating model for the agile approach using story points and velocity. The Zia dataset was used to generate a model, which was subsequently validated with the MMRE, MdMRE, PRED, R2, and MAR.

Object-oriented development approach: Nassif et al. [6] used a multilayer perceptron neural network to estimate development effort. The size of the program in the use case point and the team's productivity were used as inputs in this study. The network was trained using the back-propagation method, and the model's result was a software development effort. The results were validated using the mean relative error method. This study used a total of 160 ISBSG projects, academics, and small software development businesses. The UCP, which is based on expert guesses, has been used in several research to calculate effort using productivity. To evaluate effort, productivity is paired with the size measure UCP in various research. Most productivity estimates, on the other hand, are based on expert predictions, which might result in inconsistencies. Azzeh and Nassif [19] developed a hybrid model that combined the SVM and RBNN machine learning algorithms to close this gap. These approaches were created for the categorization and forecasting stages, as well as the model, which was created with the assistance of industry and student initiatives. This model was evaluated using the SA, MAE, MBRE, and MIBRE.

2. Proposed work with Polynomial regression

In the present work, the multiple linear regression is applied on the data that contains the dependent and independent variables. The stepwise approach is used for the regression analysis for the procedural, agile, and object-oriented development approaches. The stepwise approach is applied to find out the most correlated variables and by choosing the selected parameters, a software effort estimation model is given. For the estimation of effort required for software development, a regression model is given where the estimation is done with the selected parameters of three development approaches. In all the three development approaches, the most correlated variable is the size which is the lines of code, story point,

and use case point for the procedural, agile, and object-oriented projects respectively. Therefore, the size is the main input for the effort estimation model.

With only one independent variable, x, polynomial regression is a particular instance of multiple regression [20]. The model of one-variable polynomial regression can be written as presenting in equation 1.

$$y_i = \beta_o + \beta_1 x_i + \beta_2 x^2{}_i + \cdots + \beta_k x^k{}_i + e, \qquad (1)$$

where k is the degree of the polynomial. The degree of the polynomial is the order of the model.

2.1. Experiment strategy

The research is carried out by gathering data and determining the relationship between the response and predictors. After that, regression model is built using the most correlated variables of the dataset. The size, measured in lines of code, story point, and use case point, found the most correlated variable for procedural, agile, and object-oriented approaches, respectively. With the use of evaluation metrics, the model is assessed. Finally, for each of the three development methodologies, the suggested model is compared to earlier software effort estimating methods.

The proposed polynomial regression model is used as a measure in this study. For the evaluation of the effort necessary for software development, the regression required historical data. The story point, initial velocity (Vi), declaration, which is the product of friction and dynamic factors (FF*DF), total velocity (V), and workdays are the independent variables and effort is the dependent variable in the Agile dataset. The dependent variable in the procedural dataset is effort, while the independent variables are lines of code, as well as the fifteen components from the four categories of project, product, personal, and platform. One dependent parameter, effort, and four independent parameters, unadjusted weight, unadjusted use case weight, technical variables, and environmental factors, make up the object-oriented dataset. The stepwise principle was followed in the construction of regression model utilizing computational tools. The significance level of p=0.05 was used to examine the association.

2.2. Dataset

A dataset of python projects is used for procedural development in this study [21] that has 9 projects. The effort is a dependent attribute; the lines of code is an independent variable. The Zia dataset [22] was used as the basis for the agile development approach. The software projects from the six software houses were collected in [5] to build this dataset. The agile methodology is used to construct these projects. Twenty-one projects are included in the dataset. This dataset has one dependent variable, effort, and one independent variable, story point. The dataset for the estimation of use case point is taken from [23]. The real 20 column in the dataset contains the use case point (UCP) determined by dividing the real effort by 20 (the productivity threshold utilized in prior research). One dependent variable is the amount of effort required to develop software programs.

2.3. Accuracy measure

We opted to utilize the mean magnitude relative error (MMRE), which is extensively used and suggested by researchers, to compare the prediction accuracy of the estimate model. MMRE assesses the differences between expected and actual values. The formula for the MMRE is given in equation 2. The mean square error (MSE) and coefficient of determination (R^2) are also used to find out the error and accuracy. These accuracy measures are also used to validate the planned work. The MSE and R^2 are calculating as showing in the equation 3 and 4, respectively.

$$\mathrm{MMRE} = \frac{(\mathrm{ActVal} - \mathrm{EstVal})}{\mathrm{ActVal}}, \qquad (2)$$

$$\mathrm{MSE} = (\mathrm{ActVal} - \mathrm{EstVal})^2, \qquad (3)$$

$$R^2 = 1 - \frac{(\mathrm{ActVal} - \mathrm{EstVal})^2}{\left(\mathrm{ActVal} - \widehat{\mathrm{ActVal}}\right)^2}, \qquad (4)$$

where ActVal stands for the actual value, EstVal stands for the estimated value, and $\widehat{\mathrm{ActVal}}$ stands for the mean of the actual value.

3. Polynomial regression model evaluation

With polynomial regression analysis, numerous multiple linear regression models can be built for different levels of degree. From these models, the best model is chosen that give the best results for procedural, agile, and object-oriented projects. Equations 5 represent the polynomial model for the three development methodologies. This model is created with the one variable which is the size of project. Since in earlier section, the size finds most correlated variable, the polynomial model used this variable to create the model.

$$\mathrm{Effort} = a + b(s) + c(s^2) + d(s^3). \qquad (5)$$

In this model, the s is the size of project which is measured in lines of code for procedural project, story point for agile project, and use case point for object-oriented project. The, b, c, and d are the coefficients that

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

have the constant values which are different for each development methodologies. The values of these coefficients are given in table 1.

Table 1

Coefficients of proposed model based on procedural, agile, and object-oriented projects

Development approach	a	b	c	d
Procedural	0	0.5533	0.0183	0.0003
Agile	37.68	-0.4709	0.0054	0.00001
Object-oriented	0.001	19	0.0002	0.00006

4. Results and Discussion

The research identifies the correlation coefficient between the response and predictor. The size of the project has been determined to be the most correlated parameter for all three development methodologies, where the effort and other input characteristics are correlated. The size is the most significant variable to consider when estimating the project's effort. The size of procedural projects is measured in lines of code, while agile projects are measured in story points, and object-oriented projects are measured in use case points. Figure 1 depicts the relationship between code size (lines of code, story points, and use case points) and effort in terms of polynomial regression. The correlated variables used for the effort estimation model in the procedural dataset is the line of code. The story point is used in the effort estimation for agile because it is the most correlated variable to effort. In case of object-oriented projects, the use case point, the most correlated variable, is used for developing the effort estimation model. The R^2, MMRE, and MSE, as shown in table 2, are used to determine the model's accuracy.

Table 2

Accuracy matrices for three development methodologies using the proposed model

Development approach	R^2	MMRE	MSE
Procedural	91 %	0.048	1.747091
Agile	91 %	0.039	101.9186
Object-oriented	99 %	-0.012	7235.187

Table 3

Comparison of proposed model for procedural projects

Techniques	MMRE
CBR [9]	0.45
Ensemble model [16]	0.17
Proposed model	0.04

The best model for the procedural, agile, and object-oriented datasets, which have the highest accuracy for effort estimation, is shown in table 2. Figure 2, demonstrate the estimation findings for the three development methodologies. In figure 1, (a) is showing the actual and predicted effort for procedural projects, (b) is showing the actual and predicted effort for agile projects, and (c) is showing the actual and predicted effort for object-oriented projects. The results are also validated with the previous results. The comparison for the proposed model using three procedural, agile, and object-oriented is given in table 3, 4, and 5 respectively.

Table 4

Comparison of proposed model for agile projects

Techniques	MMRE
Linear regression [5]	0.07
Bayesian network	0.06
Proposed model	0.03

Table 5

Comparison of proposed model for object-oriented projects

Techniques	MMRE	R^2
Log-linear [6]	0.39	--
Linear regression [23]	--	72 %
Proposed model	-0.01	94 %

Conclusions

In the present work, the software effort estimation is done for the procedural, agile, and object-oriented development approaches. For estimating effort, the parameters are identified by finding the correlation between the dependent and independent parameters of the three development approaches. The four new coefficients are created by regression analysis. The software effort estimation model is created with the polynomial regression for the procedural, agile, and object-oriented projects. The multiple linear regression and correlation is done by using the stepwise approach for finding the suitable parameter for effort estimation. This research work conclude that the linear regression model for the agile projects gives the best results for estimating the effort required for software development.

Contributions of authors: purpose and task formulation of the software cost estimation based on different development methodologies, concept of size matrix of software using agile, traditional, and object-oriented projects, development of software effort estimation model by using polynomial regression – **Amrita Sharma**; review and analysis of references, suggesting the concept of combined model for effort estimation, development of mathematical model, and analysis and presentation of results – **Neha Chaudhary**. All the authors have read and agreed to the published version of the manuscript.

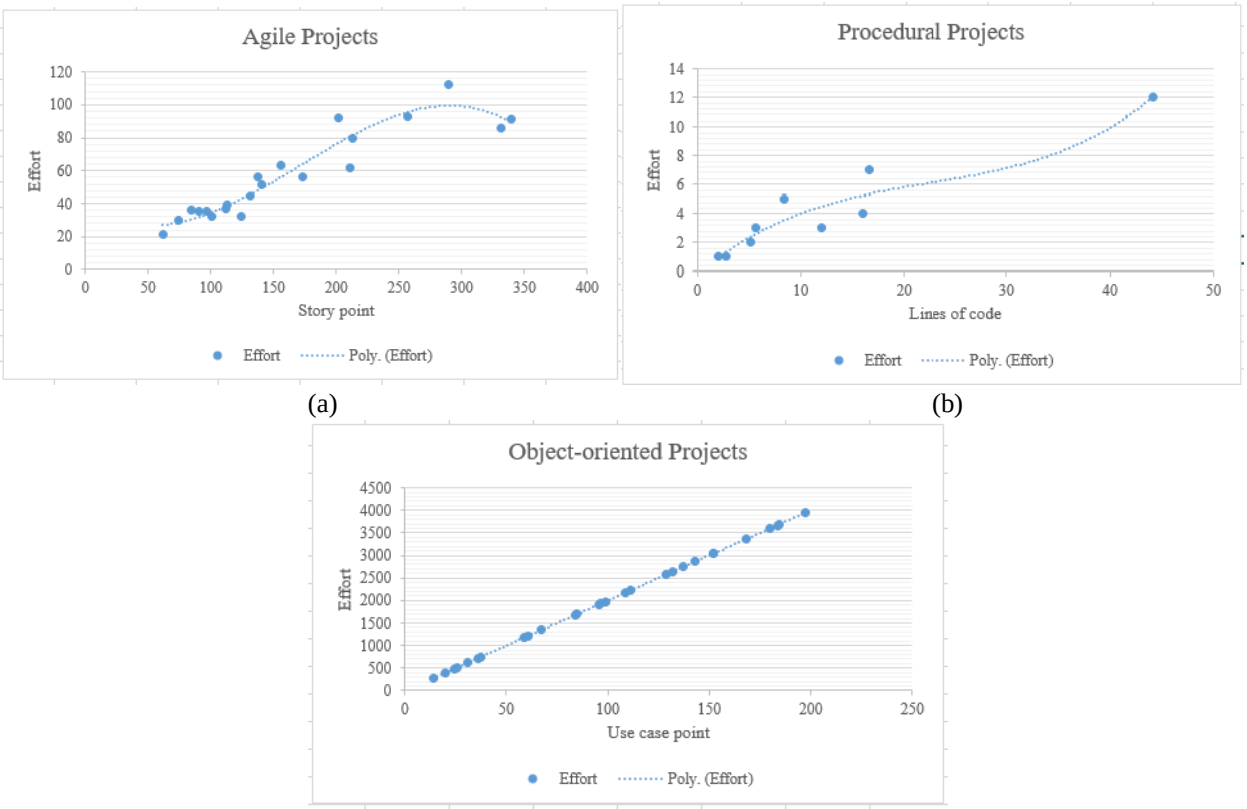

Figure 1. Polynomial regression between dependent and independent variables for (a) agile projects, (b) Procedural projects, and (c) object-oriented projects

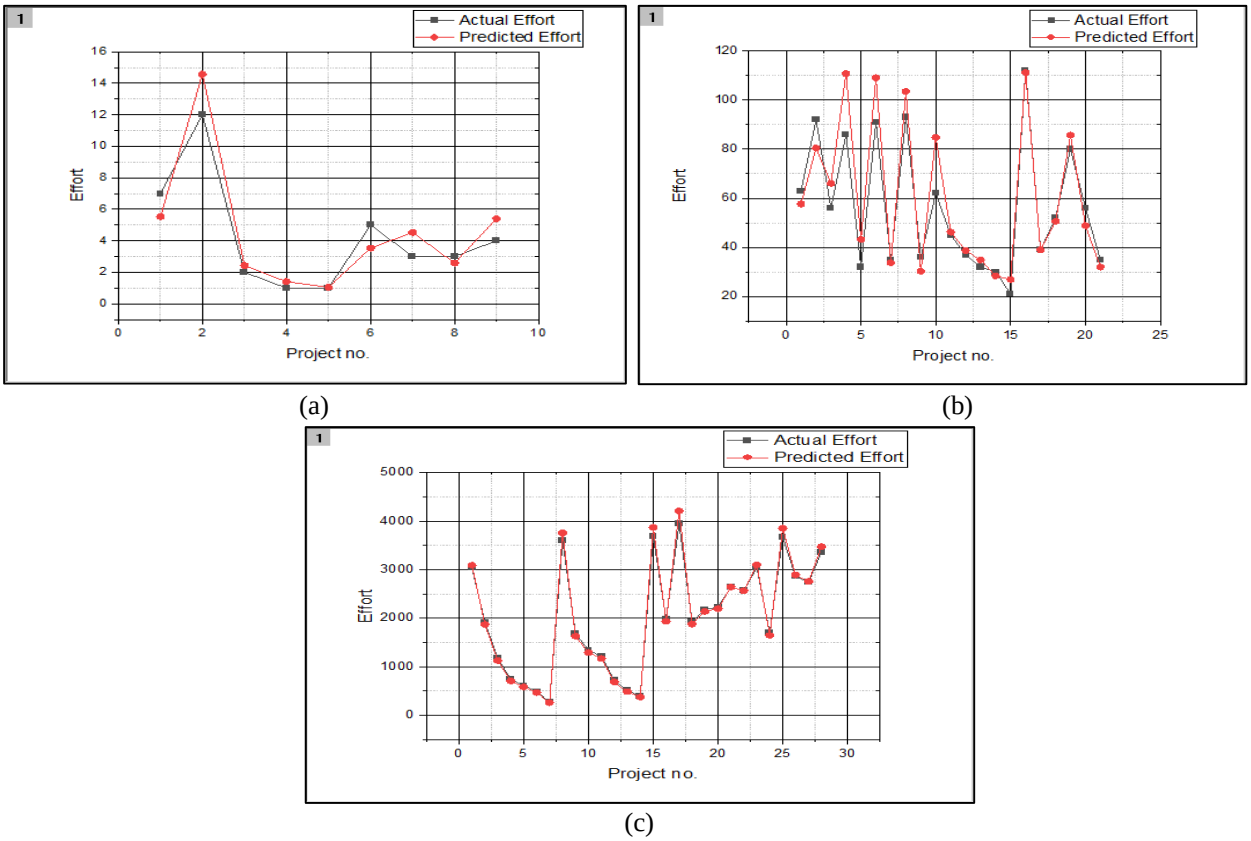

Figure 2. Actual and predicted effort for (a) procedural, (b) agile and, (c) object-oriented projects

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

References (GOST 7.1:2006)

1. Kononenko, I. Mathematical model of software development project team composition optimization with fuzzy initial data [Text] / I. Kononenko, H. Sushko // Radioelectronic and computer systems. – 2021. – Vol. 3. – P. 149-159. DOI: 10.32620/reks.2021.3.12.

2. Karner, G. Resource Estimation for Objectory Projects [Text] / Gustav Karner // Objective Systems SF AB. – 1993. – P. 1-9.

3. Boehm, B. Software Engineering Economics [Text] / B. Boehm. – Prentice Hall, Englewood Cliffs, 1981. – P. 200-217.

4. Putnam, L. H. A general empirical solution to the macro software sizing and estimating problem [Text] / L. H. Putnam // IEEE transactions on Software Engineering. – 1978. – Vol. 4. – P. 345-361.

5. Ziauddin, S. K. T. An effort estimation model for agile software development [Text] / S. K. T. Ziauddin, S. Zia // Advances in computer science and its applications (ACSA). – 2012. – Vol. 2, No. 1. – P. 314-324.

6. Nassif, A. B. Towards an early software estimation using log-linear regression and a multilayer perceptron model [Text] / A. B. Nassif, D. Ho, L. F. Capretz // Journal of Systems and Software. – 2013. – Vol. 86, No 1. – P. 144-160. DOI: 10.1016/j.jss.2012.07.050.

7. Boehm, B. W. Software Development Cost Estimation Approaches – A Survey [Text] / B. W. Boehm, C. Abts, S. Chulani // Annals of software engineering. – 2000. – Vol. 10 No. 1. – P. 177-205. DOI: 10.1023/A:1018991717352.

8. Sharma, A. Analysis of Software Effort Estimation Based on Story Point and Lines of Code using Machine Learning [Text] / A. Sharma, N. Chaudhary // International Journal Of Computing and Digital System. – 2022. – Vol. 12, No. 1. – P. 131-140. DOI: 10.12785/ijcds/120112.

9. Wu, D. Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation [Text] / D. Wu, J. Li, C. Bao // Soft Computing. – 2018. – Vol. 22. – P. 5299-5310. DOI: 10.1007/s00500-017-2985-9.

10. Dhiman, A. Optimization of COCOMO II effort estimation using genetic algorithm [Text] / A. Dhiman, C. Diwaker // American International Journal of Research in Science, Technology, Engineering & Mathematics. – 2013. – Vol. 3, No. 2. – P. 208-212.

11. Lopez-Martin, C. A fuzzy logic model for predicting the development effort of short scale programs based upon two independent variables [Text] / C. Lopez-Martin // Appl. Soft Computer. – 2011. – Vol. 11, Iss. 1. – P. 724-732. DOI: 10.1016/j.asoc.2009.12.034.

12. Briand, L. C. COBRA: Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment [Text] / L. C. Briand, K. El. Emam, F. Bomarius // Proceedings of the 20th International Conference on Software Engineering. – 1998. – P. 390-399. DOI: 10.1109/ICSE.1998.671392.

13. Angelis, L. Building a Software Cost Estimation Model Based on Categorical Data [Text] / L. Angelis, I. Stamelos, M. Morisio // Proceedings of the 7th International Software Metrics Symposium. – 2001. – P. 4-15. DOI: 10.1109/METRIC.2001.915511.

14. Shahpar, Z. Hybrid PSO-SA approach for feature weighting in analogy-based software project effort estimation [Text] / Z. Shahpar, V. Khatibi, A. Khatibi Bardsiri // Journal of AI and Data Mining. – 2021. – Vol. 9, Iss. 3. – P. 329-340. DOI: 10.22044/JADM.2021.10119.2152.

15. Improved Effort and Cost Estimation Model Using Artificial Neural Networks and Taguchi Method with Different Activation Functions [Text] / N. Rankovic, D. Rankovic, M. Ivanovic, L. Lazic // Entropy. – 2021. – Vol. 23, Iss. 7. – Article No. 854. DOI: 10.3390/e23070854.

16. Pospieszny, P. An effective approach for software project effort and duration estimation with machine learning algorithms [Text] / P. Pospieszny, B. Czarnacka-Chrobot, A. Kobylinski // Journal of Systems and Software. – 2018. – Vol. 137. – P. 184-196. DOI: 10.1016/j.jss.2017.11.066.

17. Panda, A., Empirical validation of neural network models for agile software effort estimation based on story points. [Text] / A. Panda, S. M. Satapathy, S. K. Rath // Procedia Computer Science. – 2015. – Vol. 57. – P. 772-781. DOI: 10.1016/j.procs.2015.07.474.

18. Khuat, T. T. A Novel Hybrid ABC-PSO Algorithm for Effort Estimation of Software Projects Using Agile Methodologies [Text] / T. T. Khuat, M. H. Le // Journal of Intelligent Systems. – 2018. – Vol. 27, No. 3. – P. 489-506. DOI: 10.1515/jisys-2016-0294.

19. Azzeh, M. A hybrid model for estimating software project effort from Use Case Points [Text] / M. Azzeh, A. B. Nassif // Applied Soft Computing. – 2016. – Vol. 49. – P. 981-989. DOI: 10.1016/j.asoc.2016.05.008.

20. Ostertagová, E, Modelling using polynomial regression [Text] / E. Ostertagová // Procedia Engineering. – 2012. – Vol. 48. – P. 500-506, DOI: 10.1016/j.proeng.2012.09.545.

21. Sharma, A. Software Cost Estimation for Python Projects Using Genetic Algorithm [Text] / A. Sharma, N. Chaudhary // Communication and Intelligent Systems. ICCIS 2019. Lecture Notes in Networks and Systems, Springer, Singapore. – 2019. – Vol. 120. – P. 137-148. DOI: 10.1007/978-981-15-3325-9_11.

22. Sharma, A. Linear Regression Model for Agile Software Development Effort Estimation [Text] / A. Sharma, N. Chaudhary // 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE). – 2020. – P. 1-4. DOI: 10.1109/ICRAIE51050.2020.9358309.

23. Silhavy, R. Analysis and selection of a regression model for the Use Case Points method using a stepwise approach / [Text] / R. Silhavy, P. Silhavy, Z. Prokopova // Journal of Systems and Software. – 2017. – Vol. 125. – P. 1-14. DOI: 10.1016/j.jss.2016.11.029.

References (BSI)

1. Kononenko, I., Sushko, H., Mathematical model of software development project team composition optimization with fuzzy initial data, *Radioelectronic and computer systems*, 2021, vol. 3, pp. 149-159. DOI: 10.32620/reks.2021.3.12.

2. Karner, G, Resource Estimation for Objectory Projects. *Objective Systems SF AB*, 1993, pp. 1-9.

3. Boehm, B. *Software Engineering Economics.* Prentice Hall, Englewood Cliffs, 1981, pp. 200-217.

4. Putnam, L. H. A general empirical solution to the macro software sizing and estimating problem. *IEEE transactions on Software Engineering*, 1978, vol. 4, pp. 345-361.

5. Ziauddin, S. K. T., Zia, S. An effort estimation model for agile software development. *Advances in computer science and its applications (ACSA)*, 2012, vol. 2, no. 1, pp. 314-324.

6. Nassif, A., Ho, B. D., Capretz, L. F. Towards an early software estimation using log-linear regression and a multilayer perceptron model. *Journal of Systems and Software*, 2013, vol. 86, no 1, pp. 144-160. DOI: 10.1016/j.jss.2012.07.050.

7. Boehm, B. W., Abts, C., Chulani, S. Software Development Cost Estimation Approaches – A Survey. *Annals of software engineering*, 2000, vol. 10, no. 1, pp. 177-205. DOI: 10.1023/A:1018991717352.

8. Sharma, A., Chaudhary, N. Analysis of Software Effort Estimation Based on Story Point and Lines of Code using Machine Learning. *International Journal Of Computing and Digital System*, 2022, vol. 12, no. 1, pp. 131-140. DOI: 10.12785/ijcds/120112.

9. Wu, D., Li, J., Bao, C. Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation. *Soft Computing*, 2018, vol. 22, no. 16, pp. 5299-5310. DOI: 10.1007/s00500-017-2985-9.

10. Dhiman, A. Optimization of COCOMO II effort estimation using genetic algorithm. *American International Journal of Research in Science, Technology, Engineering & Mathematics*, 2013, vol. 3, no. 2, pp. 208-212.

11. Lopez-Martin, C. A fuzzy logic model for predicting the development effort of short scale programs based upon two independent variables. *Appl. Soft Computer*, 2011, vol. 11, iss. 1, pp. 724-732. DOI: 10.1016/j.asoc.2009.12.034.

12. Briand, L. C., Emam, K. El., Bomarius, F. CORBA: Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment. *Proceedings of the 20th International Conference on Software Engineering*, 1998, pp. 390-399. DOI: 10.1109/ICSE.1998.671392.

13. Angelis, L., Stamelos, I., Morisio, M. Building a Software Cost Estimation Model Based on Categorical Data. *Proceedings of the 7th International Software Metrics Symposium*, 2001, pp. 4-15. DOI: 10.1109/METRIC.2001.915511.

14. Shahpar, Z., Khatibi, V., Khatibi Bardsiri, A. Hybrid PSO-SA approach for feature weighting in analogy-based software project effort. *Journal of AI and Data Mining*, 2021, vol. 9, iss. 3, pp. 329-340, DOI: 10.22044/JADM.2021.10119.2152.

15. Rankovic, N., Rankovic, D., Ivanovic, M., Lazic, L. Improved Effort and Cost Estimation Model Using Artificial Neural Networks and Taguchi Method with Different Activation Functions. *Entropy*, 2021, vol. 23, iss. 7, article no. 854. DOI: 10.3390/e23070854.

16. Pospieszny, P., Czarnacka-Chrobot, B. Kobylinski, A. An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*, 2018, vol. 137, pp. 184-196. DOI: 10.1016/j.jss.2017.11.066.

17. Panda, A. Satapathy, S. M., Rath, S. K. Empirical validation of neural network models for agile software effort estimation based on story points. *Procedia Computer Science*, 2015, vol. 57, pp. 772-781. DOI: 10.1016/j.procs.2015.07.474.

18. Khuat, T. T., Le, M. H. A Novel Hybrid ABC-PSO Algorithm for Effort Estimation of Software Projects Using Agile Methodologies. *Journal of Intelligent Systems*, 2018, vol. 27, no. 3, pp. 489-506. DOI: 10.1515/jisys-2016-0294.

19. Azzeh, M., Nassif, A. B. A hybrid model for estimating software project effort from Use Case Points. *Applied Soft Computing*, 2016, vol. 49, pp. 981-989. DOI: 10.1016/j.asoc.2016.05.008.

20. Ostertagová, E. Modelling using polynomial regression. *Procedia Engineering*, 2012, vol. 48, pp. 500-506. DOI: 10.1016/j.proeng.2012.09.545.

21. Sharma, A., Chaudhary, N. Software Cost Estimation for Python Projects Using Genetic Algorithm. *Communication and Intelligent Systems. ICCIS 2019. Lecture Notes in Networks and Systems*, Springer, Singapore, 2019, vol. 120, pp. 137-148. DOI: 10.1007/978-981-15-3325-9_11.

22. Sharma, A., Chaudhary, N. Linear Regression Model for Agile Software Development Effort Estimation. *5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 2020, pp. 1-4. DOI: 10.1109/ICRAIE51050.2020.9358309.

23. Silhavy, R., Silhavy, P., Prokopova, Z. Analysis and selection of a regression model for the Use Case Points method using a stepwise approach. *Journal of Systems and Software*, 2017, vol. 125, pp. 1-14. DOI: 10.1016/j.jss.2016.11.029.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Надійшла до редакції 21.12.2022, розглянута на редколегії 15.04.2022.

КОМБИНИРОВАННАЯ МОДЕЛЬ ДЛЯ ОЦЕНКИ ЗАТРАТ НА РАЗРАБОТКУ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ ПОЛИНОМИАЛЬНОЙ РЕГРЕССИИ ДЛЯ ГЕТЕРОГЕННЫХ ПРОЕКТОВ

Амрита Шарма, Неха Чаудхари

Тема: Оценка работы программного обеспечения является важной задачей лиц, участвующих в управлении проектами программного обеспечения. Сложность прогнозирования усилий усугубляется тем фактом, что разработка программного обеспечения постоянно меняется. В прошлом исследователи использовали одну форму методологии разработки в своей работе для оценки усилий и времени. Оценка программных проектов выполняется с помощью матриц разного размера. Строки кода, сюжетная точка и точка варианта использования в основном требуются для оценки с использованием алгоритмических моделей для процедурных, гибких и объектно-ориентированных подходов к разработке. В настоящее время компании используют эти три типа размерных матриц для оценки проектов. **Цель**: В данной работе предложены комбинированные программные модели оценки для трех типов методологий разработки. Оценка может быть сделана с помощью предложенной модели программного проекта, разработанного с использованием процедурного, гибкого и объектно-ориентированного подхода. **Метод**: входными данными для моделей является размер программного обеспечения, такой как строки кода, сюжетная точка и точка варианта использования. Модели разработаны с использованием множественной линейной регрессии с пошаговым подходом. Модели разрабатываются с четырьмя постоянными параметрами, которые основаны на процедурных, гибких и объектно-ориентированных проектах. Набор данных проектов Python для процедурных, набор данных zia для Agile, набор данных компании для объектно-ориентированной методологии используются для предложения моделей. **Заключение**: прогнозируются усилия для процедурных, гибких и объектно-ориентированных проектов с моделями линейной регрессии и сравниваются результаты с существующими моделями для проверки работы. В работе достигается максимальная точность оценки программных проектов с минимальной ошибкой.

Ключевые слова: оценка стоимости программного обеспечения; строки кода; пользовательская история; точка варианта использования; полиномиальная регрессия; комбинированный режим.

КОМБІНОВАНА МОДЕЛЬ ДЛЯ ОЦІНКИ ВИТРАТ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ ПОЛІНОМІАЛЬНОЇ РЕГРЕСІЇ ДЛЯ ГЕТЕРОГЕННИХ ПРОЄКТІВ

Амріта Шарма, Неха Чаудхарі

Тема: Оцінка роботи програмного забезпечення є важливою роботою осіб, які беруть участь в управлінні програмним проєктом. Труднощі в прогнозуванні зусиль ускладнюються тим фактом, що розробка програмного забезпечення постійно змінюється. Раніше дослідники використовували одну з форм методології розробки у своїй роботі для оцінки зусиль і часу. Оцінка для програмних проєктів виконується за допомогою матриць різного розміру. Рядки коду, точки історії та точки використання в основному необхідні для оцінки з використанням алгоритмічних моделей для процедурних, гнучких та об’єктно-орієнтованих підходів до розробки. Зараз компанії використовують ці три типи матриць розмірів для оцінки проєктів. **Мета**: У цій роботі запропоновано комбіновані моделі оцінки програмного забезпечення для трьох типів методологій розробки. Оцінку можна здійснити за допомогою запропонованої моделі програмного проєкту, розробленого з використанням процедурного, швидкого та об'єктно-орієнтованого підходу. **Метод**: вхідними параметрами для моделей є розмір програмного забезпечення, наприклад рядки коду, сюжетна точка та точка використання. Моделі розроблені за допомогою множинної лінійної регресії з поетапним підходом. Моделі розроблені з чотирма постійними параметрами, які базуються на процедурних, гнучких та об’єктно-орієнтованих проєктах. Набір даних проєктів Python для процедурних, набір даних zia для Agile, набір даних компанії для об’єктно-орієнтованої методології використовуються для пропонування моделей. **Висновок**: зусилля прогнозуються для процедурних, гнучких та об’єктно-орієнтованих проєктів з моделями лінійної регресії та порівнюються результати з існуючими моделями для перевірки роботи. Робота досягає найкращої точності для оцінки програмних проєктів з мінімальною похибкою.

Ключові слова: оцінка вартості програмного забезпечення; рядки коду; точка історії користувача; точка випадку використання; поліноміальна регресія; комбінований режим.

Амріта Шарма – PhD в галузі програмної інженерії та машинного навчання, Департамент комп'ютерних програм Маніпальського університету в Джайпурі, Джайпур, Індія.

Неха Чаудхарі – доцент кафедри комп’ютерних наук та інженерії в Маніпальському університеті в Джайпурі, Джайпур, Індія.

Amrita Sharma – PhD Scholar in Software engineering and Machine learning, Department of Computer Applications at Manipal University Jaipur, Jaipur, India,
e-mail: amritasharma9468@gmail.com, ORCID: 0000-0002-3513-5031.

Neha Chaudhary – Associate Professor in Department of Computer Science & Engineering at Manipal University Jaipur, Jaipur, India,
e-mail: Chaudhary.neha@jaipur.manipal.edu.