UDC 004.412:519.237.5

DETERMINING OBJECT-ORIENTED DESIGN COMPLEXITY DUE TO THE IDENTIFICATION OF CLASSES OF OPEN-SOURCE WEB APPLICATIONS CREATED USING PHP FRAMEWORKS

Prykhodko A. S. – Post-graduate student of the Department of Mathematical Support of Computer Systems, Odesa I. I. Mechnikov National University, Odesa, Ukraine.

Malakhov E. V. – Dr. Sc., Professor, Head of the Department of Mathematical Support of Computer Systems, Odesa I. I. Mechnikov National University, Odesa, Ukraine.

ABSTRACT

Context. The problem of determining the object-oriented design (OOD) complexity of the open-source software, including Web apps created using the PHP frameworks, is important because nowadays open-source software is growing in popularity and using the PHP frameworks making app development faster. The object of the study is the process of determining the OOD complexity of the open-source Web apps created using the PHP frameworks. The subject of the study is the mathematical models to determine the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks.

Objective. The goal of the work is the build a mathematical model for determining the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks based on the three-variate Box-Cox normalizing transformation to increase confidence in determining the OOD complexity of these apps.

Method. The mathematical model for determining the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks is constructed in the form of the prediction ellipsoid equation for normalized metrics WMC, DIT, and NOC at the app level. We apply the three-variate Box-Cox transformation for normalizing the above metrics. The maximum likelihood method is used to compute the parameter estimates of the three-variate Box-Cox transformation.

Results. A comparison of the constructed model based on the *F* distribution quantile with the prediction ellipsoid equation based on the Chi-Square distribution quantile has been performed.

Conclusions. The mathematical model in the form of the prediction ellipsoid equation for the normalized WMC, DIT, and NOC metrics at the app level to determine the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks is firstly built based on the three-variate Box-Cox transformation. This model takes into account the correlation between the WMC, DIT, and NOC metrics at the app level. The prospects for further research may include the use of other data sets to confirm or change the prediction ellipsoid equation for determining the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks.

KEYWORDS: object-oriented design complexity, identification of classes, open-source software, Web app, prediction ellipsoid, Box-Cox transformation, depth of inheritance tree, number of children, weighted methods per class.

ABBREVIATIONS

BCT is the Box-Cox transformation;
CBO is coupling between object classes;
DIT is a depth of inheritance tree;
KLOC is a thousand lines of code;
LCOM is a lack of cohesion in methods;
NOC is a number of children;
OOD is the object-oriented design;
PHP is a hypertext preprocessor;
RFC is a response for a class;
RMSD is a root mean square deviation;
SMD is a squared Mahalanobis distance;
WMC is weighted methods per class.

NOMENCLATURE

k is a number of variables (metrics);
N is a number of data points;
$\mathbf{S_Z}$ is a sample covariance matrix for normalized data;
$\mathbf{X}$ is a non-Gaussian random vector;
$\overline{\mathbf{X}}$ is a vector of sample means of the X_j variables;
X_1 is a WMC metric at the app level;
X_2 is a DIT metric at the app level;
X_3 is a NOC metric at the app level;
X_j is a j-th non-Gaussian variable;
$\overline{X}_j$ is a sample mean of the X_j values;
$\mathbf{Z}$ is a Gaussian random vector;
$\overline{\mathbf{Z}}$ is a vector of sample means of the Z_j variables;
Z_j is a j-th Gaussian variable that is obtained by transforming variable X_j;
$\overline{Z}_j$ is a sample mean of the Z_j values;
α is a significance level;
β_1 is a multivariate skewness;
β_2 is a multivariate kurtosis;
ν is a number of degrees of freedom;
$\chi^2_{m,\alpha}$ is the Chi-Square distribution quantile with m degrees of freedom and significance level α;
$\boldsymbol{\psi}$ is a vector of multivariate normalizing transformation.

INTRODUCTION

It's known [1], that "Complexity is one of the basic problems that associated with software development tools and methods." And complexity is one of main components of quality. Today the creation of high-quality soft-

DOI 10.15588/1607-3274-2024-2-16

ware is one of the most main tasks in the software development industry.

The design in general, and the object-oriented design (OOD) in particular, is one of the important stages of software development [1]. Booch outlined four major steps involved in the OOD process [2]. The first step in OOD is the identification of classes. In this step, key abstractions in the problem space are identified and labeled as potential classes and objects.

Now many Web apps are created using the PHP frameworks making app development faster. The above also applies to open-source Web apps since nowadays, open-source software is growing in popularity [3]. This demands the construction of mathematical models for determining the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks.

The object of study is the process of determining the OOD complexity of the open-source Web apps created using the PHP frameworks.

The subject of study is the mathematical models to determine the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks.

The purpose of the work is the build a mathematical model for determining the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks.

1 PROBLEM STATEMENT

Suppose given the original sample as the three-dimensional non-Gaussian data set following metrics at the app level: WMC X_1, DIT X_2, and NOC X_3 from N open-source Web apps created using PHP frameworks. Suppose that there are invertible three-variate normalizing transformation of non-Gaussian random vector $\mathbf{X}=\{X_1,X_2,X_3\}^T$ to Gaussian random vector $\mathbf{Z}=\{Z_1,Z_2,Z_3\}^T$ is given by

$$\mathbf{Z}=\boldsymbol{\psi}(\mathbf{X}) \tag{1}$$

and the inverse transformation for (1)

$$\mathbf{X}=\boldsymbol{\psi}^{-1}(\mathbf{Z}). \tag{2}$$

It is required to build the mathematical model for determining OOD complexity due to the identification of classes of open-source Web apps created using PHP frameworks based on the transformations (1) and (2).

2 REVIEW OF THE LITERATURE

In [4] Chidamber and Kemerer proposed a set of software metrics for OOD. According to [4], the six metrics are designed to measure the three non-implementation steps in Booch's definition of OOD. These are the metrics WMC, DIT, NOC, RFC, CBO, and LCOM, which define the OOD complexity in the above steps. In particular, the metrics WMC, DIT, and NOC define the OOD complexity due to the identification of classes in the first step. According to [5], the metrics WMC, DIT, and NOC are related to object definition which is one of the fundamental elements of OOD as outlined by Booch [6].

The metric set of Chidamber and Kemerer (CK) serves as a generalized solution for other researchers to rely on for particular purposes [7–14], including the OOD complexity [1, 15, 16].

In paper [1] the authors fitted the multiple linear regression equation to determine the OOD complexity based on the minimal set of complexity metrics which are defined using the CK metrics. Although the proposed linear regression equation is fruitful for quantifying the complexity of the OOD hierarchy, it does not directly allow taking into account the correlation between the factors – software metrics. That can affect the result of such quantifying the complexity.

The research in [16] "focuses on two primary topics: (1) how indirect coupling measurements can aid developers with maintenance tasks and (2) how indirect coupling metrics can quantify software complexity and size, leveraging weighted differences across techniques. The study presents a comprehensive set of measures designed to assist developers and project managers with project management and maintenance activities." Also, the research in [16] does not directly take into account the correlation between the software metrics. Although other researchers indicate a significant correlation between certain software metrics, including the CK metrics [7–9, 15].

In the paper [15] the authors constructed a prediction ellipse equation for the normalized RFC and CBO metrics based on the bivariate Box-Cox transformation (BCT). They apply the above equation to evaluate complexity at the third step in the OOD of apps developed in Java. The authors proposed to use the squared Mahalanobis distance (SMD) from the prediction ellipse equation for the normalized RFC and CBO metrics as the complexity indicator of OOD of open-source apps in Java from the point of view of the relationships between classes.

But the ellipse allows you to take into account the correlation only for two variables. In the case of three non-Gaussian variables, we need to apply the prediction ellipsoid for the normalized data [17].

That is why to determine the OOD complexity due to the identification of classes we apply the approach proposed in [15] with the only difference that we are going to use the prediction ellipsoid for the normalized metrics WMC, DIT, and NOC.

3 MATERIALS AND METHODS

The equation for the prediction ellipsoid for the normalized WMC, DIT, and NOC metrics is defined as

$$\left(\mathbf{Z}-\overline{\mathbf{Z}}\right)^T \mathbf{S}_{\mathbf{Z}}^{-1}\left(\mathbf{Z}-\overline{\mathbf{Z}}\right)=\frac{3\left(N^2-1\right)}{N\left(N-3\right)}F_{3,N-3,\alpha}\,, \tag{3}$$

DOI 10.15588/1607-3274-2024-2-16

where $\mathbf{Z}$ is Gaussian random vector, $\mathbf{Z}=\{Z_1,Z_2,Z_3\}^T$; $\overline{\mathbf{Z}}$ is the sample mean vector, $\overline{\mathbf{Z}}=\{\overline{Z}_1,\overline{Z}_2,\overline{Z}_3\}^T$; N is the data point number; $F_{2,N-2,\alpha}$ is a quantile of the F distribution with 3 and $N-3$ degrees of freedom; α is a significance level; $\mathbf{S_Z}$ is the sample covariance matrix

$$\mathbf{S_Z}=\frac{1}{N}\sum_{i=1}^{N}(\mathbf{Z}_i-\overline{\mathbf{Z}})(\mathbf{Z}_i-\overline{\mathbf{Z}})^T . \quad (4)$$

We can write matrix (4) in the form

$$\mathbf{S}_Z=\begin{pmatrix} S_{Z_1Z_1} & S_{Z_1Z_2} & S_{Z_1Z_3} \\ S_{Z_2Z_1} & S_{Z_2Z_2} & S_{Z_2Z_3} \\ S_{Z_3Z_1} & S_{Z_3Z_2} & S_{Z_3Z_3} \end{pmatrix}, \quad (5)$$

where $S_{Z_qZ_r}=\sum_{i=1}^{N}\left[Z_{q_i}-\overline{Z}_q\right]\left[Z_{r_i}-\overline{Z}_r\right],\ q,r=1,2,3$.

To construct equation (3) for determining OOD complexity due to the identification of classes of open-source Web apps created using PHP frameworks, we used 121 apps hosted on GitHub for five well-known frameworks (CakePHP, CodeIgniter, Laravel, Symfony, and Yii). Moreover, we also used the apps from [18] for the CakePHP framework. We obtained the data set using the PhpMetrics tool [19] around the following software metrics at the app level: WMC, DIT, and NOC. Table 1 contains the descriptive statistics of that data set.

Table 1 – App Descriptive Statistics (N=121)

Metric Name	Min	Max	Mean	RMSD
App size in KLOC	1.008	339.674	30.982	47.667
Number of classes	25	7512	525.554	920.147
WMC	1.55	67.84	14.797	13.755
DIT	1.06	2.33	3.593	4.602
NOC	0.32	0.95	0.685	0.141

We checked the three-dimensional data for three-variate outliers. But before that, we tested the normality of three-variate data because well-known statistical methods are used to detect outliers in multivariate data under the assumption that the data is described by a multivariate Gaussian distribution [20]. We applied a multivariate normality test proposed by Mardia and based on measures of the multivariate skewness β_1 and kurtosis β_2 [21]. According to this test, the distribution of three-dimensional data of the above set is not Gaussian since the test statistic for multivariate skewness $N\beta_1/6$ of this data is greater than the quantile of the Chi-Square distribution, which is 25.19 for 10 degrees of freedom and 0.005 significance level.

That is why, as in [18], to detect multivariate outliers in the three-dimensional data (121 points), we use the statistical technique based on the multivariate normalizing transformations and the squared Mahalanobis distance (SMD) for normalized data. The SMD for normalized data is the left part of equation (3).

To normalize the data, we apply the three-variate BCT with components

$$Z_j=x(\lambda_j)=\begin{cases}\left(X_j^{\lambda_j}-1\right)/\lambda_j, & \text{if} \quad \lambda_j\neq 0;\\ \ln(X_j), & \text{if} \quad \lambda_j=0.\end{cases} \quad (6)$$

Here Z_j is a Gaussian variable; λ_j is a parameter of the Box-Cox transformation, $j=1,2,3$. We denoted metrics WMC, DIT, and NOC as X_1, X_2, and X_3, respectively.

The parameter estimates of the three-variate BCT for 121 data points are calculated by the maximum likelihood method according to [20] and are $\hat{\lambda}_1=-0.177882$, $\hat{\lambda}_2=-1.272192$, $\hat{\lambda}_3=1.811640$.

In our case, there are no multivariate outliers in three-dimensional non-Gaussian data since the SMD values for all 121 data points normalized using (6) are less than the statistic

$$\frac{3\left(N^2-1\right)}{N(N-3)}F_{3,N-3,\alpha}, \quad (7)$$

which equals 13.85 for the 0.005 significance level.

To calculate the SMD value for normalized data in equation (3) we have the sample mean vector $\overline{\mathbf{Z}}$ with components 1.869, 0.284, and –0.265, respectively, and matrix (5)

$$\mathbf{S}_Z=\begin{pmatrix} 0.30283 & -0.01204 & 0.00775 \\ -0.01204 & 0.01370 & 0.00726 \\ 0.00775 & 0.00726 & 0.00967 \end{pmatrix}. \quad (8)$$

In our case, the inverse matrix of matrix (8) is

$$\mathbf{S}_Z^{-1}=\begin{pmatrix} 3.877 & 8.398 & -9.411 \\ 8.398 & 139.516 & -111.477 \\ -9.411 & -111.477 & 194.605 \end{pmatrix}. \quad (9)$$

Equation (3) for the 0.005 significance level defines the prediction ellipsoid boundary beyond which three-variate outliers can appear.

To determine OOD complexity due to the identification of classes of open-source Web apps created using PHP frameworks by (3), we need to use in (3) statistic (7) for the 0.05 significance level. Statistic (7) equals 8.25 for the 0.05 significance level.

Like [15], we can use equation (3) for determining the OOD complexity due to the identification of classes of open-source Web apps created using PHP frameworks. To do this we need to calculate the SMD value for normal-

DOI 10.15588/1607-3274-2024-2-16

ized data. If the SMD value for normalized data is greater than 13.85, then this data point is the three-variate outlier. In this case, we cannot determine OOD complexity by (3). If the SMD value for normalized data is greater than 8.25 and less than 13.85, it means that the app has high complexity due to the identification of classes. Otherwise, there is no high complexity due to the identification of classes for the Web app created using the PHP framework.

4 EXPERIMENTS

For comparison of equation (3) with another equation for the prediction ellipsoid for the normalized WMC, DIT, and NOC metrics, we built the corresponding equation in the form

$$\left(\mathbf{Z}-\overline{\mathbf{Z}}\right)^T \mathbf{S}_{\mathbf{Z}}^{-1}\left(\mathbf{Z}-\overline{\mathbf{Z}}\right)=\chi^2_{3,\alpha}\,, \qquad (10)$$

where $\chi^2_{3,\alpha}$ is the quantile of the Chi-Square distribution with 3 degrees of freedom and a significance level α. Other notations are the same as in (3).

In our case, there are no multivariate outliers in three-dimensional non-Gaussian data since the SMD values for all 121 data points normalized using (6) are less than the quantile of the Chi-Square distribution, which equals 12.84 for the 0.005 significance level.

To determine OOD complexity due to the identification of classes of open-source Web apps created using PHP frameworks by (10), we need to use in (10) the quantile $\chi^2_{3,\alpha}$ for the 0.05 significance level. The quantile $\chi^2_{3,\alpha}$ equals to 7.81 for the 0.05 significance level.

Like applying (3), we can use equation (10) to determine the OOD complexity due to the identification of classes of open-source Web apps created using PHP frameworks. To do this we also need to calculate the SMD value for normalized data. If the SMD value for normalized data is greater than 12.84, then this data point is the three-variate outlier. In this case, we cannot determine OOD complexity by (10). If the SMD value for normalized data is greater than 7.81 and less than 12.84, it means that the app has high complexity due to the identification of classes. Otherwise, there is no high complexity due to the identification of classes for the Web app created using the PHP framework.

We developed the computer program implementing built equations (3) and (10) to conduct experiments. The program was written in Python.

5 RESULTS

The results of determining OOD complexity level (CL) due to the identification of classes for ten open-source Web apps created using various PHP frameworks are shown in Table 2. We selected two such apps for each framework, for which normalized metrics have SMD values of both maximum and minimum. We denoted the SMD value for normalized metrics as SMD_Z in Table 2. Also, we denoted metrics WMC, DIT, and NOC at the app level as X_1, X_2, and X_3, respectively.

The SMD_Z values from Table 2 indicate there is no high complexity due to the identification of classes for eight apps (rows 3–10) because its SMD_Z values are less than 7.81. These are the following apps: Wildflower, Croogo, AdaptCMS, Wallabag, Yii2-podium, Yupe, Classroombookings, and Electronic invoicing and warehouse management system.

Wildflower (https://github.com/klevo/wildflower) is a CakePHP Content Management System. Croogo (https://github.com/croogo/croogo) is a CakePHP-powered Content Management System. AdaptCMS (https://github.com/adaptcms/AdaptCMS) is an open-source CMS that is made using the Laravel framework for complete control of your website. Wallabag (https://github.com/wallabag/wallabag) is a web application that allows you to save web pages for later reading and that is created using the Symfony framework. Yii2-podium (https://github.com/bizley/yii2-podium) is a Yii2 forum module project. Yupe (https://github.com/yupe/yupe) is an open-source Yii-framework-based online e-commerce solution. Classroombookings (https://github.com/classroombookings/classroombookings) is an open source hassle-free room booking system for schools that is made using the CodeIgniter framework. The electronic invoicing and warehouse management system (https://github.com/kirilkirkov/Electronic-Invoicing-And-Warehouse-Management-System) is a CodeIgniter and Bootstrap self-hosted open-source app.

Table 2 – OOD complexity levels of ten apps

N	App Name	X_1	X_2	X_3	SMD_Z	CL
1	Apiato	1.55	2.33	0.83	9.63	high
2	Ilios	8.82	1.22	0.86	8.59	high
3	Wildflower	60.34	1.56	0.95	7.77	no high
4	Croogo	9.8	1.64	0.75	0.53	no high
5	AdaptCMS	6.93	1.46	0.71	0.21	no high
6	Wallabag	7.75	1.41	0.70	0.12	no high
7	Yii2-podium	16.41	2.17	0.79	4.86	no high
8	Yupe	10.07	1.59	0.83	1.23	no high
9	Classroombooking	16.34	1.35	0.79	1.55	no high
10	Electronic invoicing and warehouse management system	49.43	1.27	0.57	4.59	no high

In contrast, two apps (Apiato and Ilios) have high complexity due to the identification of classes because their SMD_Z values are greater than 7.81 and less than 12.84. Apiato (https://github.com/apiato/apiato) is a framework for building scalable and testable APi-centric apps with PHP, built on top of Laravel. Ilios (https://github.com/ilios/ilios) is the Curriculum Management System for Health Professions that is made using the Symfony framework.

6 DISCUSSION

We apply the three-variate BCT to build the prediction ellipsoid equation for the normalized WMC, DIT, and NOC metrics for determining OOD complexity due to the

DOI 10.15588/1607-3274-2024-2-16

identification of classes of open-source Web apps created using PHP frameworks since the distribution of the three-dimensional data is not Gaussian on that the Mardia multivariate normality test based on measures of the multivariate skewness and kurtosis indicates. This is also the reason we use the statistical technique based on the multivariate normalizing transformations and the SMD for normalized data to detect three-variate outliers in the three-dimensional non-Gaussian data. According to [17, 20], we apply the 0.005 significance level for three-variate outlier detection. Also, we use the 0.05 significance level for both equations as (3) and (10). The use of both equations has led to the following results. Only two apps have high complexity. These are apps Apiato and Ilios (respectively, rows 1 and 2 in Table 2). All other 119 apps have no high complexity because the SMD values for their normalized metrics are less than 7.82. Note, that we used the 0.05 significance level that is usually assigned, although this value may be discussed.

The advantages of the proposed prediction ellipsoid equations (3) and (10) include the possibility of determining OOD complexity due to the identification of classes of open-source Web apps created using PHP frameworks. Also, the above equations take into account, firstly, the correlation between the normalized WMC, DIT, and NOC metrics, and secondly, that their three-variate distribution is not Gaussian.

Concerning the considered prediction ellipsoid equations for the normalized WMC, DIT, and NOC metrics for determining OOD complexity due to the identification of classes of open-source Web apps created using PHP frameworks, two limitations should be acknowledged and addressed concerning the data sample from 121 open-source apps in PHP. The first limitation concerns the processing of the data sample for open-source apps developed using PHP frameworks only. The processing of other data samples, for example, for commercial apps may affect the volume of the prediction ellipsoids. In such cases, equations (3) and (10) remain to be confirmed or changed. The second limitation concerns the following restrictions on software metrics at the app level: the interval for WMC is from 1.55 to 67.84, the interval for DIT is from 1.06 to 2.33, and the interval for NOC is from 0.32 to 0.95. In addition to the above restrictions, the SMD value for normalized WMC, DIT, and NOC metrics at the app level cannot be greater than 13.85 for equation (3) and cannot be greater than 12.84 for equation (10).

CONCLUSIONS

The important problem of determining the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks is solved.

The scientific novelty of obtained results is that the mathematical model in the form of the prediction ellipsoid equation for the normalized WMC, DIT, and NOC metrics at the app level to determine the OOD complexity due to the identification of classes of the open-source Web apps created using the PHP frameworks is firstly built based on the three-variate Box-Cox transformation. This model takes into account the correlation between the WMC, DIT, and NOC metrics at the app level.

The practical significance of the obtained results is that the software realizing the constructed model is developed in Python. The empirical study allows us to recommend the built model for use in practice.

Prospects for further research may include the use of other data sets to confirm or change the prediction ellipsoid equation for determining the OOD complexity due to the identification of classes and implement the algorithms developed in [22] for the classification of mass problems of production subject domains to design the Web apps which are created using the PHP frameworks.

ACKNOWLEDGEMENTS

This work is proactive. The research was carried out within the framework of the scientific activity of the working hours of the authors according to their main positions.

REFERENCES

1. Khan S. A., Khan R. A. Object oriented design complexity quantification model, *Procedia Technology,* 2012, Vol. 4, pp. 548–554. DOI: 10.1016/j.protcy.2012.05.087.
2. Booch G. Object oriented design with applications. Redwood City, CA, Benjamin/Cummings, 1991, 580 p.
3. Madaehoh A., Senivongse T. OSS-AQM: An open-source software quality model for automated quality measurement, *Data and Software Engineering (ICoDSE) : the 2022 International Conference*. Denpasar, Indonesia, proceedings, IEEE, 2022, pp. 126–131. DOI: 10.1109/ICoDSE56892.2022.9972135
4. Chidamber S. R., Kemerer C.F. Towards a metrics suite for object oriented design, *ACM SIGPLAN Notices,* 1991, Vol. 26, Issue 11, pp. 197–211. DOI: 10.1145/118014.117970
5. Chidamber S. R., Kemerer C. F. A metrics suite for object-oriented design, *IEEE Transactions on Software Engineering,* 1994, Vol. 20, No. 6, pp. 476–493. DOI: 10.1109/32.295895
6. Booch G. Object-oriented development, *IEEE Transactions on Software Engineering*, 1986, Vol. 12, No. 1, pp. 211–221. DOI: 10.1109/TSE.1986.6312937
7. Barkmann H., Lincke R., Löwe W. Quantitative evaluation of software quality metrics in open-source projects, *Advanced Information Networking and Applications Workshops, 2009 International Conference.* Bradford, UK, 2009, proceedings, pp. 1067–1072. DOI: 10.1109/WAINA.2009.190
8. Sabahat N., Afzal Malik A., Azam F. Utility of CK metrics in predicting size of board-based software games, *Mehran University Research Journal of Engineering and Technology*, 2017, Vol. 36, No. 4, pp. 975–986.
9. Molnar AJ., Neamţu A., Motogna S. Evaluation of software product quality metrics, in E. Damiani, G. Spanoudakis, L. Maciaszek. Eds. *Evaluation of Novel Approaches to Software Engineering*. ENASE 2019. Communications in Computer and Information Science, Vol. 1172, Springer, Cham, 2020, pp. 163–187. DOI: 10.1007/978-3-030-40223-5_8

DOI 10.15588/1607-3274-2024-2-16

10. Rizwan M., Nadeem A., Sindhu M. A. Empirical evaluation of coupling metrics in software fault prediction, *Applied Sciences and Technology (IBCAST), 2020 17th International Bhurban Conference*. Islamabad, Pakistan, 2020, proceedings, IEEE, 2020, pp. 434–440. DOI: 10.1109/IBCAST47879.2020.9044489
11. Tapia V., Gaona C. Research opportunities in microservices quality assessment: A systematic literature review, *Journal of Advances in Information Technology,* 2023, Vol. 14, No. 5, pp. 991–1002. DOI: 10.12720/jait.14.5.991-1002
12. Wikantyasa I. M. A., Kurniawan A. P., Rochimah S. CK metric and architecture smells relations: Towards software quality assurance, *Information & Communication Technology and System (ICTS), 2023 14th International Conference.* Surabaya, Indonesia, 2023, proceedings, IEEE, 2023, pp. 13–17. DOI: 10.1109/ICTS58770.2023.10330874
13. Jin W. Zhang Y., Shang J. et al. Identifying code changes for architecture decay via a metric forest structure, *Technical Debt (TechDebt) : 2023 ACM/IEEE International Conference*. Melbourne, Australia, 2023, proceedings, pp. 62–71. DOI: 10.1109/TechDebt59074.2023.00014
14. Levasseur M., Badri M. Prioritizing unit tests using object-oriented metrics, centrality measures, and machine learning algorithms [Electronic resource], *Innovations in Systems and Software Engineering*, 2024. DOI: 10.1007/s11334-024-00550-9. Access mode: https://doi.org/10.1007/s11334-024-00550-9
15. Prykhodko S., Prykhodko N., Smykodub T. A joint statistical estimation of the RFC and CBO metrics for open-source applications developed in Java, *Computer Sciences and Information Technologies : the 2022 IEEE 17th International Conference (CSIT)*. Lviv, Ukraine, 10–12 November, 2022, proceedings. pp. 442–445. DOI: 10.1109/CSIT56902.2022.10000457
16. Navas-Su J., Gonzalez-Torres A., Hernandez-Vasquez M. et al. A metrics suite for measuring indirect coupling complexity, *Programming and Computer Software,* 2023, Vol. 49, Issue 8, pp. 735–761. DOI: 10.1134/S0361768823080157
17. Prykhodko S. Makarova L., Prykhodko K. et al. Application of transformed prediction ellipsoids for outliers detection in multivariate non-gaussian data, *Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET) : the 15th International Conference, IEEE,* Lviv-Slavske, 2020, proceedings, pp. 359–362. DOI: 10.1109/TCSET49122.2020.235454
18. Prykhodko S. B., Shutko I. S., Prykhodko A. S. A nonlinear regression model to estimate the size of web apps created using the CakePHP framework, *Radio Electronics, Computer Science, Control,* 2021, Vol. 59, No. 4, pp. 129–139. DOI: 10.15588/1607-3274-2021-4-12
19. PhpMetrics is a static analysis tool for PHP [Electronic resource]. Access mode: https://phpmetrics.org/
20. Johnson R. A., Wichern D. W. Applied multivariate statistical analysis. Pearson Prentice Hall, 2007, 800 p.
21. Mardia K.V. Measures of multivariate skewness and kurtosis with applications, *Biometrika*, 1970, Vol. 57, pp. 519–530. DOI: 10.1093/biomet/57.3.519
22. Malakhov E., Shchelkonogov D., Mezhuyev V.Algorithms of classification of mass problems of production subject domains, *Software and Computer Applications (ICSCA 2019) : 2019 8th International Conference, Feb. 19–21, 2019.* Penang, Malaysia, proceedings, pp. 149–153. DOI: 10.1145/3316615.3316676

Received 20.02.2024.
Accepted 01.04.2024.

УДК 004.412:519.237.5

ВИЗНАЧЕННЯ СКЛАДНОСТІ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОЕКТУВАННЯ ЗАВДЯКИ ІДЕНТИФІКАЦІЇ КЛАСІВ ВЕБ-ЗАСТОСУНКІВ З ВІДКРИТИМ КОДОМ, СТВОРЕНИХ ЗА ДОПОМОГОЮ PHP-ФРЕЙМВОРКІВ

Приходько А. С. – аспірант кафедри математичного забезпечення комп'ютерних систем Одеського національного університету імені І. І. Мечникова, Одеса, Україна.

Малахов Є. В. – д-р техн. наук, професор, завідувач кафедри математичного забезпечення комп'ютерних систем Одеського національного університету імені І. І. Мечникова, Одеса, Україна.

АНОТАЦІЯ

Актуальність. Проблема визначення складності об'єктно-орієнтованого проектування (ООП) програмного забезпечення з відкритим вихідним кодом, включаючи веб-програми, створені за допомогою фреймворків PHP, є важливою, оскільки сьогодні програмне забезпечення з відкритим кодом стає все популярнішим і використання фреймворків PHP робить розробку застосунків швидшою. Об'єктом дослідження є процес визначення складності ООП веб-застосунків з відкритим кодом, створених за допомогою фреймворків PHP. Предметом дослідження є математичні моделі для визначення складності ООП завдяки ідентифікації класів веб-застосунків з відкритим кодом, створених за допомогою фреймворків PHP.

Мета. Метою роботи є побудова математичної моделі для визначення складності ООП завдяки ідентифікації класів веб-застосунків з відкритим кодом, створених з використанням фреймворків PHP, на основі тривимірного нормалізуючого перетворення Бокса-Кокса для підвищення достовірності визначення складність ООП цих застосунків.

Метод. Математична модель для визначення складності ООП завдяки ідентифікації класів веб-застосунків з відкритим кодом, створених за допомогою фреймворків PHP, побудована у формі рівняння еліпсоїда прогнозування для нормалізованих метрик WMC, DIT і NOC на рівні застосунку. Ми застосовуємо тривимірне перетворення Бокса-Кокса для нормалізації наведених вище метрик. Метод максимальної правдоподібності використовується для обчислення оцінок параметрів тривимірного перетворення Бокса-Кокса.

Результати. Проведено порівняння побудованої моделі на основі квантиля F-розподілу з рівнянням еліпсоїда прогнозування на основі квантиля розподілу хі-квадрат.

Висновки. Математична модель у формі рівняння еліпсоїда прогнозування для нормалізованих метрик WMC, DIT та NOC на рівні програми для визначення складності ООП через ідентифікацію класів веб-застосунків з відкритим кодом, створених за допомогою фреймворків PHP, у перше побудована на основі тривимірного перетворення Бокса-Кокса. Ця мо-

DOI 10.15588/1607-3274-2024-2-16

дель враховує кореляцію між метриками WMC, DIT та NOC на рівні програми. Перспективи подальших досліджень можуть включати використання інших наборів даних для підтвердження або зміни рівняння еліпсоїда прогнозування для визначення складності ООП завдяки ідентифікації класів веб-застосунків з відкритим кодом, створених за допомогою фреймворків PHP.

КЛЮЧОВІ СЛОВА: складність об'єктно-орієнтованого проектування, ідентифікація класів, програмне забезпечення з відкритим кодом, веб-застосунок, еліпсоїд прогнозування, перетворення Бокса-Кокса, глибина дерева успадкування, кількість дітей, зважені методи на клас.

ЛІТЕРАТУРА

1. Khan S. A. Object oriented design complexity quantification model / Suhel Ahmad Khan, Raees Ahmad Khan // Procedia Technology. – 2012. – Vol. 4 – P. 548–554. DOI: 10.1016/j.protcy.2012.05.087.
2. Booch G. Object oriented design with applications / G. Booch. – Redwood City, CA: Benjamin/Cummings, 1991. 580 p.
3. Madaehoh A. OSS-AQM: An open-source software quality model for automated quality measurement / A. Madaehoh, T. Senivongse // Data and Software Engineering (ICoDSE) : the 2022 International Conference, Denpasar, Indonesia : proceedings. – IEEE, 2022. – P. 126–131. DOI: 10.1109/ICoDSE56892.2022.9972135
4. Chidamber S. R. Towards a metrics suite for object oriented design / S. R. Chidamber, C. F. Kemerer // ACM SIGPLAN Notices. – 1991. – Vol. 26, Issue 11. – P. 197–211. DOI: 10.1145/118014.117970
5. Chidamber S. R. A metrics suite for object-oriented design / S. R. Chidamber, C. F. Kemerer // IEEE Transactions on Software Engineering. – 1994. – Vol. 20, No. 6. – P. 476–493. DOI: 10.1109/32.295895
6. Booch G. Object-oriented development / G. Booch // IEEE Transactions on Software Engineering. – 1986. – Vol. 12, No. 1. – P. 211–221. DOI: 10.1109/TSE.1986.6312937
7. Barkmann H. Quantitative evaluation of software quality metrics in open-source projects / H. Barkmann, R. Lincke, W. Löwe // Advanced Information Networking and Applications Workshops : 2009 International Conference, Bradford, UK, 2009 : proceedings. – P. 1067–1072. DOI: 10.1109/WAINA.2009.190
8. Sabahat N. Utility of CK metrics in predicting size of board-based software games / N. Sabahat, A. Afzal Malik, F. Azam // Mehran University Research Journal of Engineering and Technology. – 2017. – Vol. 36, No. 4. – P. 975–986.
9. Molnar AJ. Evaluation of software product quality metrics / AJ. Molnar, A. Neamţu, S. Motogna // E. Damiani, G. Spanoudakis, L. Maciaszek. Eds. Evaluation of Novel Approaches to Software Engineering. ENASE 2019. Communications in Computer and Information Science, Vol. 1172. – Springer, Cham, 2020. – P. 163–187. DOI: 10.1007/978-3-030-40223-5_8
10. Rizwan M. Empirical evaluation of coupling metrics in software fault prediction / M. Rizwan, A. Nadeem, M. A. Sindhu // Applied Sciences and Technology (IBCAST) : 2020 17th International Bhurban Conference, Islamabad, Pakistan, 2020 : proceedings. – IEEE, 2020. – P. 434–440. DOI: 10.1109/IBCAST47879.2020.9044489
11. Tapia V. Research opportunities in microservices quality assessment: A systematic literature review / V. Tapia, C. Gaona // Journal of Advances in Information Technology. – 2023. – Vol. 14, No. 5. – P. 991–1002. DOI: 10.12720/jait.14.5.991-1002
12. Wikantyasa I. M. A. CK metric and architecture smells relations: Towards software quality assurance / I. M. A. Wikantyasa, A. P. Kurniawan, S. Rochimah // Information & Communication Technology and System (ICTS) : 2023 14th International Conference, Surabaya, Indonesia, 2023 : proceedings. – IEEE, 2023. – P. 13–17. DOI: 10.1109/ICTS58770.2023.10330874
13. Identifying code changes for architecture decay via a metric forest structure / [W. Jin, Y. Zhang, J. Shang et al.] // Technical Debt (TechDebt) : 2023 ACM/IEEE International Conference, Melbourne, Australia, 2023: proceedings. – P. 62–71. DOI: 10.1109/TechDebt59074.2023.00014
14. Levasseur M. Prioritizing unit tests using object-oriented metrics, centrality measures, and machine learning algorithms [Electronic resource] / M. Levasseur, M. Badri // Innovations in Systems and Software Engineering. – 2024. DOI: 10.1007/s11334-024-00550-9. – Access mode: https://doi.org/10.1007/s11334-024-00550-9
15. Prykhodko S. A joint statistical estimation of the RFC and CBO metrics for open-source applications developed in Java / S. Prykhodko, N. Prykhodko, T. Smykodub // Computer Sciences and Information Technologies : the 2022 IEEE 17th International Conference (CSIT), Lviv, Ukraine, 10–12 November, 2022 : proceedings. – P. 442–445. DOI: 10.1109/CSIT56902.2022.10000457
16. Navas-Su J. A metrics suite for measuring indirect coupling complexity / [J. Navas-Su, A. Gonzalez-Torres, M. Hernandez-Vasquez et al.] // Programming and Computer Software. – 2023. – Vol. 49, Issue 8. – P. 735–761. DOI: 10.1134/S0361768823080157
17. Application of transformed prediction ellipsoids for outliers detection in multivariate non-gaussian data / [S. Prykhodko, L. Makarova, K. Prykhodko et al.] // Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET) : the 15th International Conference, IEEE, Lviv-Slavske, 2020 : proceedings. – P. 359–362. DOI: 10.1109/TCSET49122.2020.235454
18. Prykhodko S.B. A nonlinear regression model to estimate the size of web apps created using the CakePHP framework / S. B. Prykhodko, I. S. Shutko, A. S. Prykhodko // Radio Electronics, Computer Science, Control. – 2021. – Vol. 59. No. 4. – P. 129–139. DOI: 10.15588/1607-3274-2021-4-12
19. PhpMetrics is a static analysis tool for PHP [Electronic resource]. – Access mode: https://phpmetrics.org/
20. Johnson R. A. Applied multivariate statistical analysis / R. A. Johnson, D. W. Wichern. – Pearson Prentice Hall, 2007. – 800 p.
21. Mardia K. V. Measures of multivariate skewness and kurtosis with applications / K. V. Mardia // Biometrika. – 1970. – Vol. 57. – P. 519–530. DOI: 10.1093/biomet/57.3.519
22. Malakhov E. Algorithms of classification of mass problems of production subject domains / E. Malakhov, D. Shchelkonogov, V. Mezhuyev // Software and Computer Applications (ICSCA 2019) : 2019 8th International Conference, Feb. 19–21, 2019, Penang, Malaysia : proceedings. – P. 149–153. DOI: 10.1145/3316615.3316676

DOI 10.15588/1607-3274-2024-2-16