Code Readability Metric for PHP

Cho Thet Mon[1]
Myanmar Institute of Information Technology

Mandalay, Maynmar
chothetmonucsm@gmail.com

Su Su Hlaing[2]
Myanmar Institute of Information Technology

Mandalay, Maynmar

Mie Mie Tin[3]
Myanmar Institute of Information Technology

Mandalay, Myanmar

Mie Mie Khin[4]
Myanmar Institute of Information Technology

Mandalay, Myanmar

Tin Moh Moh Lwin[5]
Myanmar Institute of Information Technology

Mandalay, Myanmar

Khin Mar Myo[6]
Central Institute of Civil Service
Upper Myanmar

***Abstract*—PHP is one of the most popular languages that millions of websites and web applications are developed every month using PHP. In this paper, software metric related to code readability is developed and suitable set of attributes are identified for the given system to establish a software quality observatory for PHP language. Correlation criterion was used to empirically validate the developed metric and to find out the nature of relationship between the metric and other well-known metric. Moreover, the other validation, such as validation by using PHP Mess Detector (PHPMD) tool was also evaluated in the system. According to these analysis results, the proposed metric is applicable and usable to the field of software engineering metrics by providing a practical summary of the metric validation criteria found in the academic literature.**

Keywords—software metric, readability, software quality, Correlation, PHP Mess Detector

I. INTRODUCTION

Software metrics are important and useful in several activities of the software development life cycle to make assessment and to determine the quality of software product or process [1]. The main contribution of the proposed system is to develop PHP software quality measuring tool that can measure readability of PHP language. The objectives are to contribute new metric for PHP codes, to develop a tool that can automatically collect different readability attributes and to investigate the relation between the proposed software metric and other indicator of software quality.

II. SOURCE CODE READABILITY ($SCREA_{PHP}$)

According to PHP coding guidelines and standards, readability related features will be extracted from the PHP source codes to measure the quality of readability. To enhance readability, variable names should be camel-case started with a lower case letter. Readable class names should be descriptive and should be avoided using abbreviations where possible and must be started with an upper case letter [6, 7]. Method names should be camel-case started with a lowercase letter. Private and protected attributes have an underscore prefix. Constants should be all uppercase and words are separated by an underscore. Moreover, it is needed to detect when a field has a very short name or very long name.

According to the above factors, in the study, the number of readable variable names and method names are defined as names that are declared with lower camel-case letter and the length of variable or method names are between 3 and 20 letters. The number of readable class names is defined as the class name started with upper camel-case letter and the length of it is greater than 3 letters. The rules and usage of line length in source code and documentation are also important aspects of coding standards for many years. A common line length should be 80 characters in PHP to maximize readability [5]. According to these factors, readable code line is defined as the number of line length that is less than 80 characters.

Having blank lines in source code can increase user's readability quality, but they are needed to be in the reasonable places. The number of blank lines before functions, return, comment or traits can also lead to readable code [6, 8]. In that respect, blank lines before functions, blank lines before comment, return and traits are considered as readable blank lines of program source code in the proposed system. Therefore, it is important to consider the above factors and they become the base attributes to calculate proposed readability metric ($SCREA_{php}$) of PHP codes. The required base attributes for readability metric are described in Table 1. They are important factors for achieving a high code quality.

TABLE 1. BASE ATTRIBUTES FOR READABILITY

Acronyms	Description	Acronyms	Description
RMN	Readable Method Names	MN	numbers of methods defined in a file
RBL	Readable Blank Lines	BL	numbers of blank lines in a file
RCM	Readable Class Names	CM	numbers of class names defined in a file
RVN	Readable Variable Names	VN	numbers of variables defined in a file
RL	Readable Lines	LN	numbers of lines defined in a file

$$SCREA_{php} = \frac{\sum_{c=1}^{d} RMN_c}{\sum_{a=1}^{b} MN_a} * w1 + \frac{\sum_{g=1}^{h} RBL_g}{\sum_{e=1}^{f} BL_e} * w2 + \frac{\sum_{k=1}^{l} RCN_k}{\sum_{i=1}^{j} CN_i} * w3 + \frac{\sum_{o=1}^{p} RVN_o}{\sum_{m=1}^{n} VN_m} * w4 + \frac{\sum_{s=1}^{t} RL_s}{\sum_{q=1}^{r} LN_q} * w5 \quad (1)$$

Equation (1) is readability metric ($SCREA_{php}$) to measure the quality of PHP code and w1, w2, w3, w4 and w5 are corresponding weights that are assigned using rank-order weight method [3]. To assign attribute weight, the process of ranking attributes was performed based on their importance and then weight values that are computed with rank-order method have been assigned.

III. Empirical Validation of Proposed Metrics

Software metric validation is needed to indicate how a new metric is valid, that is, suitable and acceptable for its intended use. In order to demonstrate the applicability of the new metric according to objective evaluation based on facts or theory, IEEE Standard 1061 validation criteria were mainly emphasized [4]. From these criteria, correlation is reasonable and applicable for the proposed metric. The purpose is to externally validate proposed metric by investigating its statistical relationship to external software quality including maintainability index (MI) that is validated metric [2].

To check the validity of proposed metric, the processes are as follows. First, 210 PHP files in different natures are used that were extracted from the web. Then, the required attributes of these files were extracted and readability metric was calculated by using Equation (1). After that a statistical analysis process is used to discover the relationship between source code metric and the maintainability of the software.

The other task is using the PHPMD tool to find possible violations of 210 PHP files [8]. Each PHP code is run from command line window using this tool and export xml file. After that the required violation count are parsed by using DOM parser. And then, the relations are investigated among the violation **results** given by PHPMD tool and the proposed metric results.

The following hypotheses based on many research papers and evaluation results are proposed to identify whether there is a consistent relation between the proposed metric and other quality indicators. The study tested two hypotheses which relate the selected metric to the MI and code violation and an empirical study have been carried out and tried to find out the nature of their relationships.

Hypothesis 1: The readability is positively related to code maintainability. The software systems with less readable source code are recognized as more difficult to maintain than those with more readable source code.

Hypothesis 2: The codes that are difficult to read will also produce more violations.

In the research, proposed metric is calculated by using 210 PHP files and maintainability index for these programs are also calculated. Then, the relation between them is studied by applying correlation indicators, PEARSON coefficient being one of them. The correlations among the proposed metric, MI and Rule Violation-Based Measures using PHPMD Tool are presented in below.

TABLE 2: CORRELATION BETWEEN PROPOSED METRICS AND MI

Proposed Metrics	MI	PHPMD Violation
Readability	0.15098	-0.0651

Table 2 summarizes statistical analysis results among software metric, well-known MI and violation. According to the critical value table, the relation among calculated readability and MI was greater than the critical value from the table of the p-values 0.05 [9]. This means that there is a significant relationship between the proposed metrics and MI that supported Hypothesis 1 that readable codes are easy to maintain than less readable codes. The scatter plot for correlation between Readability and MI is shown in Figure 1.1.

The other relation is investigated among the violation results given by PHPMD tool and the proposed metric results. From the table, the only noticeable feature is the negative relationship between readability and possible code violation. This means that as readability increases, the number of violation found decreases. This supports Hypothesis 2.

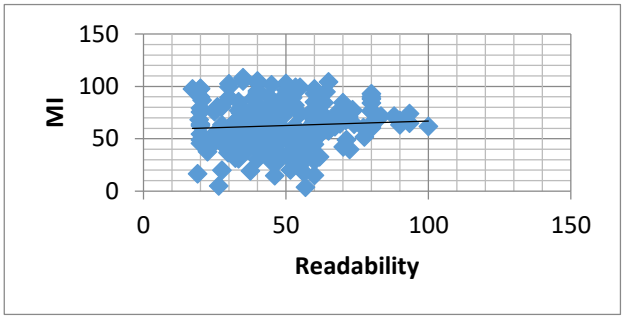

Figure 1.1 Scatter Plot for Correlation between Readability and MI

IV. Conclusion

For assessing PHP quality, readability metric is valuable measure for software developers to support as a quality measuring tool that can automatically collect readability attributes.The proposed software quality metric can also measure the quality of codes written in a multi-paradigm programming especially object-oriented features codes. Therefore, the proposed metric is valuable and usable for measuring and evaluating software quality and helps software developers to provide useful feedback and to facilitate private assessment of PHP based applications. Future work may involve extending the existing set of proposed metrics to develop more features so that they become more practical.

References

[1] J. Bansiya and C.G. Davis, "*A Hierarchical Model for Object-Oriented Design Quality Assessment*", IEEE Transactions on Software Engineering, Vol. 28, No. 1, 2002.

[2] P. Oman and J. Hagemeister, "*Construction and Testing of Polynomials Predicting Software Maintainability*," The Journal of Systems and Software, Vol. 24, no. 3, September 1994, p. 251-266.

[3] F. Barron and E. Barrett, "*Decision Quality Using Rank Attribute Weights*," Computer Journal of Management Science, Vol. 42, no. 11, 1996, p. 256-259.

[4] IEEE Std 1061-1998 (R2009), "*IEEE Standard for a Software Quality Metrics Methodology*", Software Engineering Standards Committee of the IEEE Computer Society, 2009.

[5] https://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code--net-8118

[6] https://github.com/joomla/coding-standards/wiki/Php

[7] http://flowframework.readthedocs.io/en/stable/TheDefinitiveGuide/PartV/CodingGuideLines/PHP.html

[8] http://phpmd.org

[9] http://www.statisticssolutions.com/table-of-critical-values-pearson-correlation/