Software Maintainability Metrics Model: An Improvement in the Coleman-Oman Model

Aldo Liso
AIPA

This article analyzes the contribution of the number-of-comments parameter used in the Coleman-Oman regression model, a model aimed at determining the maintainability index of a software system. Some characteristics of this parameter can prove to be unsuitable for application to many software systems. A change to the number-of-comments parameter of the Coleman-Oman model will therefore be proposed, as well as a change of the way to obtain the evaluation of the parameter. These changes do not touch the structure of the parameter and are likely to give better results, which can be widely used. This article does not include research that results in a new parameter, but is intended to stimulate thought within the software community that results in the required research and new parameter.

Among the various metrics used to measure software system maintainability, the class of regression models defined by Coleman–Oman and others [1] have raised particular interest in the academic, industrial, and public administration circles. These models are based on a combination of variables (independent metrics) in a polynomial expression that allow calculation of the maintainability index. The studies documented by several software laboratories, the University of Idaho, and Hewlett Packard have shown the high significance, in the software system maintainability evaluation, of the regression models based on Halstead metrics, McCabe's Cyclomatic Complexity, Lines of Code, and Number of Comments.

It is known that the software system maintainability is of the utmost importance, since through it the following is possible:

- Monitor changes to the software system and its quality characteristic.
- Make decisions about the most appropriate maintenance strategy for software procedures by locating the source code that could cause the greatest risks.
- Compare the quality of various software systems and supply information for the best choice among these systems.

The Coleman–Oman Model

The most used model for determining the maintainability index (MI) of a software system is the Coleman-Oman model described in [2]:

Model 1:

1) $$MI = 171 - 5.2\ln(aveV) - 0.23aveV(g') - 16.2\ln(aveLOC) + 50\sin\sqrt{2.46\,perCM}$$

where:

- *aveV* is the average Halstead Volume per module.
- *aveV(g')* is the average extended cyclomatic complexity per module.
- *aveLOC* is the average lines of code per module.
- *perCM* is the average percent of lines of comment per module.

It is known that the first version of Model 1 contained the *aveCM* variable (average lines of comment per module) rather than the *perCM* variable (average percent of lines of comment per module) [2]. However, this first version was not satisfactory because it was too sensitive to the presence of a large number of comments. This means that the presence of large comment blocks, especially inside small modules, results in an increase of the maintainability values. In order to correct such a behavior, the *aveCM* variable was replaced with the *perCM* variable and a threshold value of 50 for this variable was determined and applied [2].

Welker and Oman suggested choosing between Model 1 and the following Model 2:

2) $$MI = 171 - 5.2\ln(aveV) - 0.23aveV(g') - 16.2\ln(aveLOC)$$

which is derived from Model 1 by removing the *comments* variable due to the considerations on the comments arising from code reading. They suggested using Model 2 when the following hypotheses are verified:

- The comments are not closely related to the code. In such situations the comments can become out-of-sync with the code and therefore make the code less maintainable.
- At the beginning of each module there are large blocks of comments (company-standard comments, for example) that bring a negligible benefit to the maintainability of the software. In such situations biased estimates of the maintainability are highly probable.
- There are large sections of code that have been commented out, which creates maintenance difficulties.

In short, the choice between the two regression models depends on the evaluation of the contribution of the comment variable to the software maintainability. Only when this contribution has been demonstrated is it advisable to introduce the number-of-comments variable in the model. In a major research effort due to Hewlett Packard [3], the following thresholds for the evaluation of the maintainability index, calculated by means of the previous models, have been determined:

$MI < 65$	poor maintainability
$65 \le MI < 85$	fair maintainability
$85 \le MI$	excellent maintainability

Comments Component Analysis

If we analyze the contribution of the

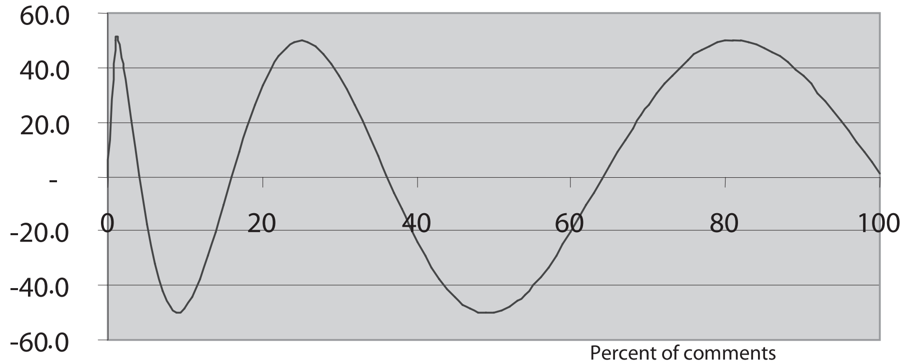


Figure 1: *Values of the Function:* $50\, sin\sqrt{2.46\, perCM}$

perCM factor as indicated in Model 1

$$50\, sin\sqrt{2.46\, perCM}$$

we can notice a sinusoidal path with maximum height equal to 50. The function must be considered in the [0, 100] interval, closed on the left and opened to right, therefore excluding "100" value. In fact, while it is possible to consider a module without comments, it is senseless to consider a module made up by comments only.

By analyzing the path of this function in relation to the maintainability metric, we can notice that there are some drawbacks to this function:

- The function has three peaks at 1 percent, 25 percent, and 81 percent. Now, while it is commonly accepted that a normal value for the comment variable is around 25 percent, the other two values are not justified.
- The decreasing path of the function in the interval from the 1 percent to 9 percent is inexplicable, since the trend would have to be positive in this interval.
- It is difficult to explain the function path in the interval to the right of the 49 percent *perCM* value.

It is also important to make the following remarks. If the comments are significant, as required by the Oman model, the function must be increasing between 0 percent and a fixed maximum value (for example 20 percent); then it must decrease gradually for the successive values. Also we believe that a situation in which the comment value is around 80 percent is to be preferred to the total absence of comments. Therefore the value of the function, there, must be positive, even if its contribution is insignificant. In conclusion it is important to search a function with a new path similar to the following one in Figure 2.

The Proposed Model Modification

There are various solutions to this problem. In this article, a modification to the model of Coleman-Oman is proposed that leaves the structure of the original metric unchanged.

$$A\ sin\sqrt{K \cdot perCM}$$

The change is made by means of the following procedure. First of all it is necessary to determine the multiplicative coefficient K that better represents the contribution to the maintainability of the comment variable in the specific software system to be evaluated. Then, once the set of the values of the variable

$$sin\sqrt{K \cdot perCM}$$

has been calculated, it is possible to determine the parameter A = β_5 by Equation 3:

$$MI = \beta_1 + \beta_2 \ln(aveV) + \beta_3\, aveV(g') + \beta_4 \ln(aveLOC) + \beta_5\, sin\sqrt{K \cdot perCM} \qquad (3)$$

The problem of the parameters estimation consists in obtaining numerical values for the coefficients, i.e., values for the parameters multiplying each of the variables of the Equation 3, and for the intercept.

The estimate of β_5 (A) parameter must be obtained through the same calculation process used to determine the estimates of the other parameters of the model: the intercept, the parameters of ln(*aveV*), *aveV(g')*, and ln(*aveLOC*).

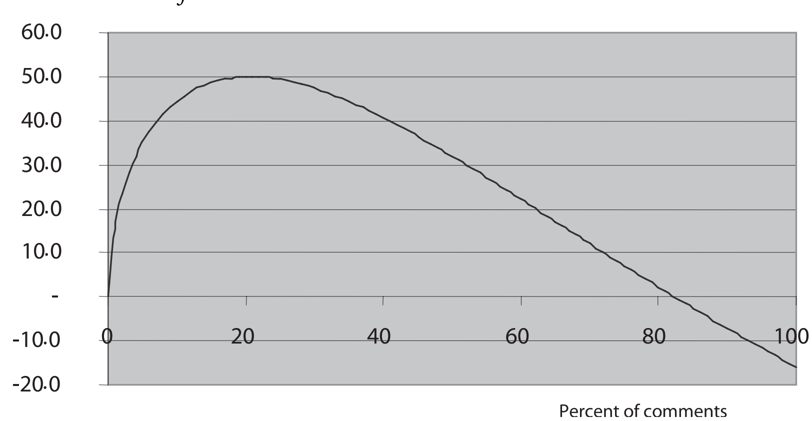
As for the determination of the coefficient *K* in the hypothesis, we neglect the *open reengineering* criterion[1]; various remarks can be made, and a great number of different factors can influence its calculation (language, programming support environment, etc.). We believe that the parameter *K* is mainly dependent on the used language. High-level languages generally require a greater percentage of comments with an equal number of developed lines of code. The diagram in Figure 3 shows the value of the component

$$\beta_5\, sin\sqrt{K \cdot perCM}$$

for values of *K* = 2.46, *K* = 0.12, *K* = 0.16, *K* = 0.20, in the case that the value of the parameter β_5 (amplitude) is equal to 50, which is the value set by Coleman-Oman in their model. Every language has its own characteristic curve; the curves shown in the diagram are obviously given by way of example, because investigation in this direction has not been carried out yet.

The main characteristics of these curves are that they differ slightly in the optimal value of the percentage of the comments; however, they differ greatly regarding the different values of comment percentage for which it is assumed that to exceed such threshold would have a negative influence on the maintainability.

Figure 2: *Ideal Path of Comments Contribution*



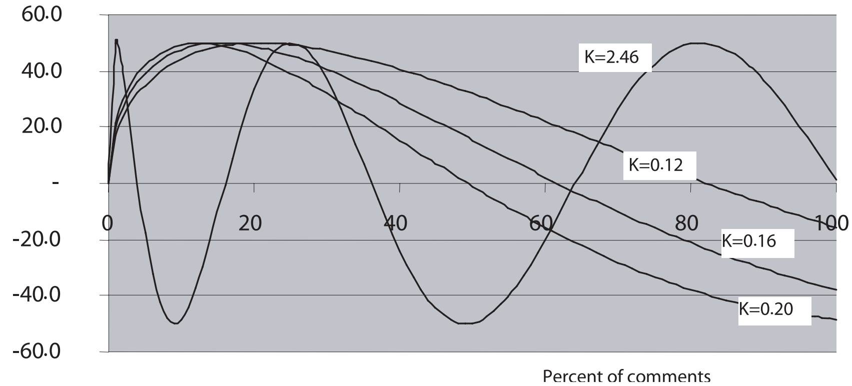


Figure 3: *Values of the Function:* $50 \sin \sqrt{K \cdot perCM}$

Conclusion

A method has been proposed to improve the Coleman-Oman model by modifying the contribution given by the comment component. The contribution to the improvement of the model of the β_5 parameter calculation is particularly significant since this parameter, with the new method proposed, will no longer be set from the outside. Instead it will be determined simultaneously by means of the same process of estimation adopted to determine the estimates of the other parameters of the model.

Future developments will therefore have to regard the estimation of the coefficient K, the estimation of the parameters of the modified Coleman-Oman model, and the experimentation of such model in industrial and public administration software systems.

References

1. Welker, K.D. and Oman, P., Software Maintainability Metrics Models in Practice, CROSSTALK, Vol. 8, pp. 19-23 Nov./Dec. 1995, www.stsc.hill.af.mil/crosstalk/1995/nov/maintain.asp
2. Coleman, D.; Lowther, B.; and Oman, P.; Using Metrics to Evaluate Software System Maintainability, *IEEE Computer*, Vol. 27(8), pp. 44-49, Aug. 1994.
3. Coleman, D.; Lowther, B.; and Oman, P.; The Application of Software Maintainability Models on Industrial Software Systems, University of Idaho, Software Engineering Test Lab, Report No. 93-03 TR, Nov. 1993.
4. Watson, A. and McCabe, T., Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric, NIST, Sept. 1996, www.itl.nist.gov/div897/sqg/pubs/publications.htm

Note

1. The open reengineering concept is similar in that the abstract models used to represent software systems should be as independent as possible of implementation characteristics such as source code formatting and programming languages. The objective is to be able to set model standards and interpret the resultant numbers uniformly across software systems [4].

Additional Reading

1. Coleman, D., Assessing Maintainability, 1992 Software Engineering Productivity Conference Proceedings, Hewlett-Packard, pp. 525-532, 1992.
2. Oman, P. and Hagemeister, J., Constructing and Testing of Polynomials Predicting Software Maintainability, *Journal of Systems and Software*, Vol. 24(3), March 1994.

About the Author

Aldo Liso received the Laurea degree in Physics from the University of Bari. He is a senior member of the technical staff of Autorità per l'informatica nella Publica Amministrazione (AIPA) the Authority for I.T. in the Italy Public Administration. Liso has 20 years of experience working on software development projects and he is Professor of information systems at the DUSIT - University of Rome "La Sapienza."

AIPA
Via Isonzo, 21
00198 Roma, Italy
Phone: +39 6 85264 305
Fax: +39 6 85264 255
E-mail: liso@aipa.it