



## Review paper: A Study on Metric For Code Readability

Deepa Dhabhai\*

Amity University Jaipur,  
Rajasthan, India

Dr. A. K. Dua

HOD (CSE) Amity University Jaipur,  
Rajasthan, India

Anil Saroliya

Amity University Jaipur,  
Rajasthan, India

**Abstract**— *In this paper we have presented the comparative analysis of the papers related to metric for the code readability. The critical factors in maintaining the software quality is readability and the readability of the program is related to its maintainability. The Code readability is very important in the software development process. In this paper we define the new readability metric that can be used to measures the readability of the given source code. In the present day's software industry is using the software metrics to estimate the complexity of software systems to find the software cost estimation, the software development control, the software testing, the software assurance and the software maintenance.*

*The readability of the program is related to its maintainability, and is thus the key factor in overall software quality. And typically, the maintenance will consume over the 70% of the total lifecycle cost of the software product.*

**Keywords** — *software quality, Software metrics, Code Readability, maintainability.*

### I. INTRODUCTION

Readability can be defined as the human judgment of understanding the text. And the critical factor in maintaining the software quality's is the readability and the readability of the program is related to its maintainability. And where the cost of the software product in the total life cycle of its maintenance will consume around 70%. And in the maintenance of the software both the source code readability and the documentation readability play the critical role.

On other hand some of the researchers have noted that the act of reading the code is the most time-consuming component of all the maintenance activities. And as of the modern software engineering, the maintaining software often means evolving the software and then modifying existing code. And the readability is also the another important attributes of the software systems that gives substantial affect on the software maintainability. Maintenance of the less readable source code is much more difficult than the source code which has more readable source code And the readability Metrics are the family of the software metrics that measure software complexity with taking the readability into the considerations. And there are several uses from this automated readability metric like, which helps in writing more of the readable software to the developers by quickly identifying code which scores poorly and also it can also monitor and maintain the readability of a code which support the project managers. And it can even assist inspections by helping to target effort at parts of the program that may need improvement. It can serve as the requirement for acceptance. And according to the Dijkstra, the readability of the program depends on many of the factors such as the simplicity of the control sequences comments, top down approach and so on.

### II. IMPORTANCE AND RELEVANCE OF THE STUDY

Dijkstra, as an example, claimed that the readability of a program depends for the most part upon the simplicity of its sequencing management (e.g., he conjectured that goto unnecessarily complicates program understanding), and utilized that notion to assist encourage his top-down approach to system style.

According to paper “ Learning a Metric for Code Readability” by Raymond P.L. Buse, Westley Weimer

In this study, they need chosen to focus on readability directly each as a result of it's an inspiration that's severally valuable, and additionally as a result of developers have nice management over it. They show that there's so a significant correlation between readability and quality.

The main contributions of this paper are:

- a method for the development of associate degree automatic code readability metric supported native code options.
- A survey of a hundred and twenty human annotators on one hundred code snippets that forms the idea of the metric bestowed and evaluated during this paper. they're unaware of any printed code readability study of comparable size (12,000 human judgments). They directly valuate the performance of our model on this knowledge set.
- a collection of experiments that reveal significant correlations between our metric for readability and external notions of code quality as well as defect density.
- A discussion of the options concerned in our metric and their relevance code engineering and artificial language style.

This article additionally includes some previous work that:

- an extra experiment correlating our readability metric with a a lot of natural notion of code quality and defect density specific human mentions of bug repair. mistreatment code version management repository data they coarsely separate changes created to deal with bugs from alternative changes. They find that low readability correlates a lot of powerfully

with this direct notion of defect density than it will with the previous approach of mistreatment potential bugs according to static analysis tools.

- an extra experiment that compares readability to cyclomatic quality. This experiment serves to validate our claim that our notion of readability is basically freelance from ancient measures of code quality.
- an extra longitudinal experiment showing however changes in readability will correlate with changes in defect density as a program evolves.

On the opposite hand several major comes like UNIX system, Java , MySQL and a few fashionable compilers has gained unimaginable visibility and validation as open supply model of code. However, however specifically (if at all) do shoppers of open supply live the standard and security of any piece of code to see ,if it's a decent suitable their stack? Few would disagree that a lot of eyes reviewing code could be a excellent thanks to cut back the quantity of defects. However, no effective yardstick has been out there to live however sensible the standard extremely is.

software quality attributes outlined by ISO-9126, maintainability is recognized by several researchers as having the most important impact on code quality (Troy, 1995). At the 1992 code Engineering Productivity conference, a Hewlett- Packard government explicit that sixty – eightieth of their analysis and development workers were committed maintaining forty – fifty million SLOC (Troy, 1995). Glass (2002) states that computer code maintenance consumes from forty – eightieth of the overall computer code value, with a mean of hour. Jakob Bohme and Basili (2001) report a mean of seventieth.

### **Readability and quality**

It is vital to notice that readability isn't identical as quality, that some existing metrics are by trial and error shown helpful. whereas computer code quality metrics generally take under consideration the dimensions of categories and ways, and also the extent of their interactions, the readability of code is sbased totally on native, line-by-line factors. Our notion of readability arises directly from the judgments of actual human annotators United Nations agency don't have context for the code they're judgment. quality factors, on the opposite hand, might have very little relevancy what makes code apprehensible to humans.

## **III. BASIC TECHNIQUES AND PROCEDURES**

To understand why Associate in Nursing empirical and objective model of computer code readability is beneficial, take into account the employment of readability metrics in natural languages. The Flesch-Kincaid Grade Level, the Gunning-Fog Index, the smogginess Index, and also the machine-driven Readability Index area unit simply some samples of readability metrics for standard text. These metrics area unit all supported easy factors like average syllables per word and average sentence length.

Despite this simplicity, they need every been shown to be quite helpful in apply. Flesch-Kincaid, that has been in use for over fifty years, has not solely been integrated into well-liked text editors as well as Microsoft Word, however has conjointly become a us governmental customary. Agencies, as well as the Department of Defense, need several documents and forms, internal and external, to fulfill have a Flesch readability grade of ten or below (DOD MIL-M-38784B).Defense contractors are oftenrequired to use it after they write technical manuals.

Some existing readability matrices-

### **A. The machine-driven Readability Index (ARI)**

In this metric word problem and sentence problem ratios ar used. The word problem refers to range of letters per word and sentence problem refers to the amount of worlds gift in a very sentence. the primary step is to ascertain the factors used and so relate them to different indices. The syntax associated with factoris same thereto of presently used indices. within the method the verification of relationship between the factors is self- evident. There ar 2 factors related to most readability factors. the primary issue is expounded to syntax that is formed from range of words. The second issue is expounded to morphology that is formed upof letters. glossary has benefits, and it's slow and comparatively inaccurate once it's applied to reading materialof adults. The linguistic unit count isn't reliable.

The equation to work out readability with ARI is

$$4.71(\text{characters}) + \text{zero}.5 (\text{words}) - \text{twenty one}.43$$

Words sentences.

### **B. SMOG**

In 1969 G Harry McLaughlin created the air pollution readability metric. It estimates the the amount of years of education one must comprehend a bit of text. this is often AN improvemnet over different readability formulae. air pollution stands for easy live of jargon. Some believe that it stands for Robert Gunning's FOG. air pollution formula is as shown below.

$$\text{SMOG grade} = \text{three} + \text{root of word Count}$$

### **C. The Gunning's Fog Index**

This readability metric is thought as FOG Index developed by Robert Gunning. per him the readability formula is

$$\text{Grade Level} = \text{zero}.4 (\text{ASL} + \text{PHW}) (3)$$

where

(ASL = Average Sentence Length (i.e., range of words divided by the amount of sentences)

PHW = share of onerous Words ).

In this paper they outline a replacement metric based mostly on a group of rules collected from package engineers. the foundations square measure unbroken a part of formula of the new metric. This will mean that the new metric is developed victimization individual rules computations and so substituted into the most formula. once implementing the new metric, they evaluated the metric by tantalizing fifty package engineers asking them to supply their readability share (from human perspective). Then the results square measure averaged and compared with the results of the example application for constant supply files. The comparison results square measure encouraging and therefore the new metric are often employed in world package development communities. Readability metric helps in understanding the share of readability of given ASCII text file.

This paper aims at methodology wont to outline the new metric is employed here. 1st of all some rules that may be wont to live readability square measure obtained from senior package engineers. It will mean that we have a tendency to asked some senior package engineers to supply criteria for readability metric and picked up their responses. once creating associate initial list, the list is valid and eventually seven rules square measure thought of to be a part of new metric. The seven rules and their representative notations square measure given in table one. this technique is impressed by

Table 1 – Rules involved in new readability metric

RULE	NOTATION
Lines of Code	LOC
Line Length	LL
Presence of comment lines in the program	NOCL
No. of Blank Lines	NOBL
Breaking the line after semicolon	NLAS
Blank space after directive statements	BSAD
No. of Methods	NOM

As are often seen in table one, the notations for all seven rules square measure given. These notations square measure wont to deduce associate equation for the new metric. The new code readability metric is painted by

$$CR = LOC + LL + NOCL + NOBL + NLAS + BSAD + NOM$$

where Cr stands for Code Readability. alternative notations square measure provided in table one. As a part of the methodology to outline new metric and implement it, a example net application is made victimization Visual Studio that takes a supply file as input and extracts values for LOC, LL, NOCL, NOBL, NLAS, BSAD and NOM. later on those values square measure substituted within the equation four that is employed to reason the readability of given ASCII text file. The result are in share of readability. The additional during this share, the additional the ASCII text file readability is. once implementing the new metric, we have a tendency to evaluated the metric by tantalizing package engineers asking them to supply their readability percentage(from human perspective). Then the results square measure averaged and compared with the results of the example application for constant supply files.

The planned readability metric provides result that is incredibly nearer to the results given by human participants. This reveals that the new metric enforced is helpful find readability of ASCII text file.

#### IV. CONCLUSIONS

The In this paper we've got given readability live for modeling code readability supported the judgments of human annotators And here, we have a tendency to given that it's potential to form a metric that agrees with these annotators the maximum amount as they believe one another by solely considering a comparatively easy set of low-level code options. package engineers as a part of their development life cycle will use the projected metric to reason readability. Thus readability will be improved and successively this improves the standard of merchandise.

we've got conjointly determined that readability provides a major Level of correlation with additional typical metrics of package quality, like defects, code churn, and self rumored Stability.

#### REFERENCES

- [1] K. Aggarwal, Y. Singh, and J.K. Chhabra, "An Integrated Measure of Software Maintainability," Proc. Reliability and Maintainability Symp., pp. 235-241, Sept. 2002
- [2] L.E. Deimel, Jr., "The Uses of Program Reading," ACM SIGCSE Bull., vol. 17, no. 2, pp. 5-14, 1985.
- [3] D.R. Raymond, "Reading Source Code," Proc. Conf. Center for Advanced Studies on Collaborative Research, pp. 3-16, 1991.
- [4] S. Rugaber, "The Use of Domain Knowledge in Program Understanding," Ann. Software Eng., vol. 9, nos. 1-4, pp. 143-192, 2000.
- [5] B. Boehm and V.R. Basili, "Software Defect Reduction Top 10 List," Computer, vol. 34, no. 1, pp. 135-137, Jan. 2001.
- [6] J.L. Elshoff and M. Marcotty, "Improving Computer Program Readability to Aid Modification," Comm. ACM, vol. 25, no. 8, pp. 512-521, 1982
- [7] J.C. Knight and E.A. Myers, "Phased Inspections and Their Implementation," ACM SIGSOFT Software Eng. Notes, vol. 16, no. 3, pp. 29-35, 1991.

- [8] N.J. Haneef, "Software Documentation and Readability: A Proposed Process Improvement," ACM SIGSOFT Software Eng. Notes, vol. 23, no. 3, pp. 75-77, 1998.
- [9] P.A. Relf, "Tool Assisted Identifier Naming for Improved Software Readability: An Empirical Study," Proc. Int'l Symp. Empirical Software Eng., Nov. 2005.
- [10] E.W. Dijkstra, A Discipline of Programming. Prentice Hall PTR, 1976.
- [11] The Automated Readability Index (ARI): <http://www.readabilityformulas.com/automatedreadability-index.php>
- [12] The smog readability formula: <http://www.readabilityformulas.com/smog-readability-formula.php>
- [13] The Gunning's Fog Index (or FOG) Readability Formula: <http://www.readabilityformulas.com/gunning-fog-readability-formula.php>
- [14] Raymond P.L. Buse and Westley R. Weimer, "Learning a Metric for Code Readability", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 36, NO. 4, JULY/AUGUST 2010.
- [15] Rajendar Namani<sup>1</sup>, Kumar J<sup>2</sup> "A New Metric for Code Readability" *ISSN: 2278-0661, ISBN: 2278-8727* Volume 6, Issue 6 (Nov. - Dec. 2012), PP 44-48.