© 2016 IJSRSET | Volume 2 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

# Comparative Analysis of Line of Code Metric Tools

**Anureet Kaur**

Department of Computer Science & Applications, Khalsa College, Amritsar, Punjab, India

## ABSTRACT

LOC (Line of code) is considered as one of the easiest measure for various purposes by stakeholders. It is a software metric used to measure the size of a software program by counting the number of lines in the program's source code. It is also used as an estimation method for calculating effort involved in software development process. This paper shows the existing software LOC metric tools. An experiment is conducted on a same java project for different set of freely available LOC count tools. Results thus obtained have high level of variability in one of the tool and makes them difficult in reaching some assessments. So, the stakeholder must consider other aspects as well to reach at correct elucidation.

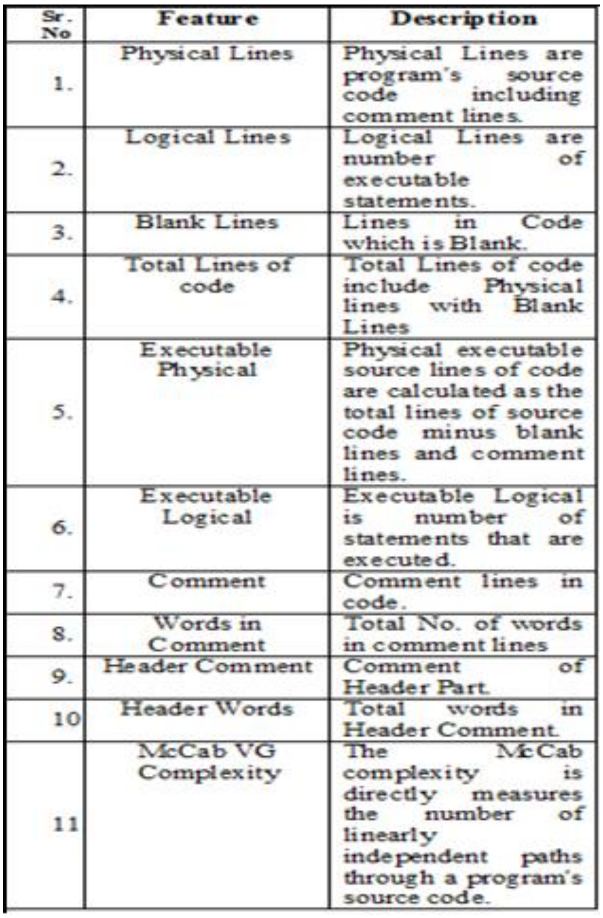
**Keywords:** LOC, Software Metrics, Software engineering, Analysis

## I. INTRODUCTION

Size is considered as one of the most important attributes of a software product. Using this one can not only find the software cost and time but also a base unit to obtain other metrics for project status and software quality measurement. According to a survey on cost estimation approaches, size metric is used as an essential input for most of cost estimation models [1].A standard metric for counting size of software is Lines of code (LOC). LOC is used as a measure for counting lines of code in a program. The blank lines, white spaces, and comments lines are excluded in the total count. The count will be different for different programming languages. There are various tools freely available to provide the count. The tool is expected to compute some other aspects also of the source program. These aspects are shown in figure1 [2] [3]

In the first section of this paper a brief review is done on the freely available tools used for counting LOC. At the end of this section a summarized view of each tool is given. In the Next section an experiment is shown on a java project using the five different tools and results of each tool is compared against the same java project.

## II. METHODS AND MATERIAL



| Sr. No | Feature | Description |
|---|---|---|
| 1. | Physical Lines | Physical Lines are program's source code including comment lines. |
| 2. | Logical Lines | Logical Lines are number of executable statements. |
| 3. | Blank Lines | Lines in Code which is Blank. |
| 4. | Total Lines of code | Total Lines of code include Physical lines with Blank Lines |
| 5. | Executable Physical | Physical executable source lines of code are calculated as the total lines of source code minus blank lines and comment lines. |
| 6. | Executable Logical | Executable Logical is number of statements that are executed. |
| 7. | Comment | Comment lines in code. |
| 8. | Words in Comment | Total No. of words in comment lines |
| 9. | Header Comment | Comment of Header Part. |
| 10 | Header Words | Total words in Header Comment. |
| 11 | McCab VG Complexity | The McCab complexity is directly measures the number of linearly independent paths through a program's source code. |

**Figure 1:** Features of LOC metric Tool

## Review of Existing LOC count tools

There are numerous tools available online for counting line of code.In this paper five freely available LOC count tool are used compare the results. The tools used are as follows:

### A. LocMetrics

LocMetrics counts total lines of code (LOC), blank lines of code (BLOC), comment lines of code (CLOC), lines with both code and comments (C&SLOC), logical source lines of code (SLOC-L), McCabe VG complexity (MVG), number of comment words (CWORDS) and header comments (HCLOC), and header words (HCWORD). Physical executable source lines of code (SLOC-P) are calculated as the total lines of source code minus blank lines and comment lines. It’s a simple and easy to use application that provides detailed information on your project metrics .Counts are calculated on a per file basis and accumulated for the entire project. LocMetrics also generates a comment word histogram. [4]

### B. LineTally

LineTally is a freeware application designed to allow programmers to count the number of lines in their source code. It supports about 61 languages. An unlimited number of user defined languages can be defined as well. It can tell the programmer how many code lines, comment lines, mixed lines (code and comments on the same line), blank lines, and total lines that were found as well as their respective percentages. [5]

LineTally supports a wide range of programming languages (more than 60), including Visual Basic, Visual Basic.NET, VBScript, B++, C/C++, C#, J#, SQL/Oracle SQL, Informix, AWK, Haskell, Delphi, HTML, Pascal, Cold Fusion, Fortran 77, Fortran 90/95, XML, Batch, Objective-C, Java, PHP, ASP, Forth, and lot more. LineTally works with high accuracy and produces super-fast results. Apart from that, reports can be exported to CSV, HTML, or TXT format.

### C. SourceMonitor

SourceMonitor tracks source code quality and quantity. For results SourceMonitor Version 3.5 is used. SourceMonitor is free monitoring software for the Microsoft Windows operating system that collects and compares source code metrics. [6]

The program supports a variety of programming languages that include C++, C#, Java, Visual Basic, Delphi and HTML. Depending on the selected programming language advanced features like method and function level metrics may be available for a more detailed analysis and comparison.

### D. Universal Code Lines Counter

Universal Code Lines Counter is a free application that counts the exact number of source code lines in a software project. It’s a handy and reliable tool that separately counts the lines of code, blank lines, and comments in multiple files in one go. Universal Code Lines Counter comes with an intuitive program interface that not only counts the source code lines, but also gives you detailed report of your software project as table or as graphics (in pie chart form).

By default it supports many programming languages including C#, C/C++, AutoIt, CSS, Delphi/Pascal, Java, JavaScript, HTML, Modula 2/3, MySQL, PHP, Prolog, Python, Perl, Ruby, VBScript/ASP, XML and Visual Basic. Plus it gives you options to configure it for more programming or script languages. [7]

### E. Any Code Counter

Any Code Counter Inc. is a software company that develops and markets code counter with rich functions and flexibilities. With Any Code Counter, one can easily count any source files accurately and export counting results in multiple formats. Any Code Counter supports all the programming languages: C Ada, Assembly, ASP, Basic, BAT, C, C++, C#, CSS, Delphi, HTML, INI, Java, JavaScript, Makefile, Pascal, Perl, PHP, Python, Ruby, SQL, Text, XML, and even self-defined languages.[ 8]

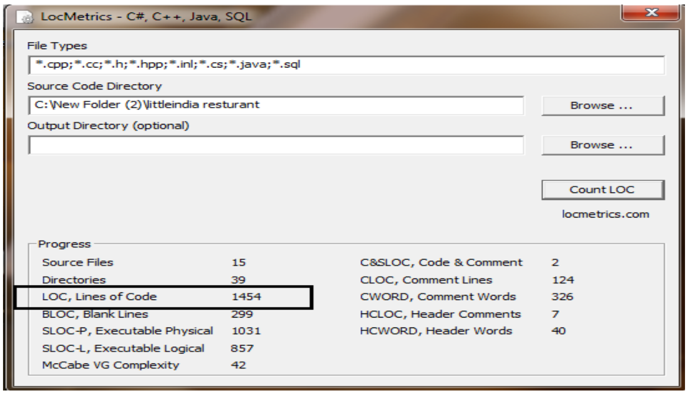
Various features of all the above said five tools are summaries in the table1.In this table the version of the specific tool, interface through which the tool can be used, the programming language whose source code is recognised in the tool and finally the format in which the report can be saved is shown.

Table 1: Features of the Tools

| LOC Metric Tool/Features | Version | Interface | Programming Languages Support | Report Export Format |
|---|---|---|---|---|
| LocMetrics | - | GUI | C/C++, Visual C#.NET,SQL/Oracle SQL | CSV |
| LineTally | 1.7 | GUI | Visual Basic, Visual Basic.NET, VBScript, B++, C/C++, C#, J#, SQL/Oracle SQL, Informix, AWK, Haskell, Delphi, HTML, Pascal, Cold Fusion, Fortran 77, Fortran 90/95, XML, Batch, Objective-C, Java, PHP, ASP, Forth, and 60 more | CSV, HTML, or TXT format |
| SourceMonitor | 3.5 | GUI | C++, C#, Java, Visual Basic, Delphi and HTML | XML or CSV files. |
| Universal Code Lines Counter | 1.1.6 | GUI | C#, C/C++, AutoIt, CSS, Delphi/Pascal, Java, JavaScript, HTML, Modula 2/3, MySQL, PHP, Prolog, Python, Perl, Ruby, VBScript/ASP, XML and Visual Basic. | CSV,HTML |
| Any Code Counter | 1.0.1 | GUI | C Ada, Assembly, ASP, Basic, BAT, C, C++, C#, CSS, Delphi, HTML, tINI, Java, JavaScript, Makefile, Pascal, Perl, PHP, Python, Ruby, SQL, Text, XML, and even self-defined languages. | CSV, HTML, XML.XLS, format |

## III. RESULTS AND DISCUSSION

The results are obtained on a java project. All tools are given same project as an input. The results obtained in shown in the following figures.


LocMetrics - C#, C++, Java, SQL

File Types: *.cpp;*.cc;*.h;*.hpp;*.inl;*.cs;*.java;*.sql

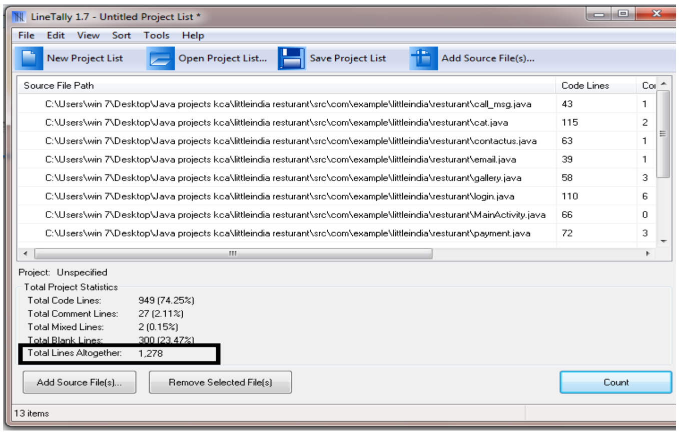
Source Code Directory: C:\New Folder (2)\littleindia resturant

Output Directory (optional):

Progress

| | | | |
|---|---|---|---|
| Source Files | 15 | C&SLOC, Code & Comment | 2 |
| Directories | 39 | CLOC, Comment Lines | 124 |
| LOC, Lines of Code | 1454 | CWORD, Comment Words | 326 |
| BLOC, Blank Lines | 299 | HCLOC, Header Comments | 7 |
| SLOC-P, Executable Physical | 1031 | HCWORD, Header Words | 40 |
| SLOC-L, Executable Logical | 857 | | |
| McCabe VG Complexity | 42 | | |

**Figure 2** : Screen shot of LocMetric


LineTally 1.7 - Untitled Project List *

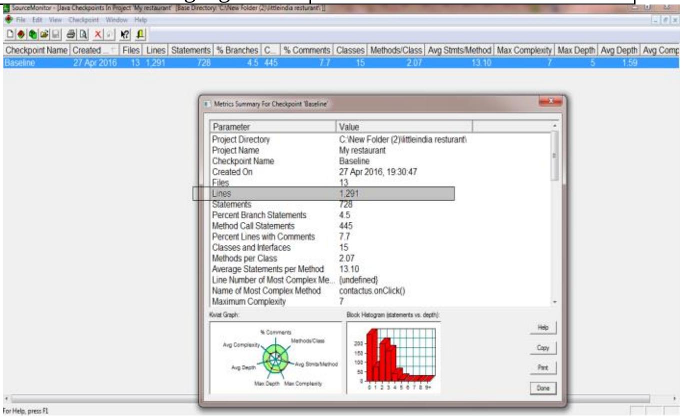
| Source File Path | Code Lines | Co |
|---|---|---|
| C:\Users\win 7\Desktop\Java projects kca\littleindia resturant\src\com\example\littleindia\resturant\call_msg.java | 43 | 1 |
| C:\Users\win 7\Desktop\Java projects kca\littleindia resturant\src\com\example\littleindia\resturant\cat.java | 115 | 2 |
| C:\Users\win 7\Desktop\Java projects kca\littleindia resturant\src\com\example\littleindia\resturant\contactus.java | 63 | 1 |
| C:\Users\win 7\Desktop\Java projects kca\littleindia resturant\src\com\example\littleindia\resturant\email.java | 39 | 1 |
| C:\Users\win 7\Desktop\Java projects kca\littleindia resturant\src\com\example\littleindia\resturant\gallery.java | 58 | 3 |
| C:\Users\win 7\Desktop\Java projects kca\littleindia resturant\src\com\example\littleindia\resturant\login.java | 110 | 6 |
| C:\Users\win 7\Desktop\Java projects kca\littleindia resturant\src\com\example\littleindia\resturant\MainActivity.java | 66 | 0 |
| C:\Users\win 7\Desktop\Java projects kca\littleindia resturant\src\com\example\littleindia\resturant\payment.java | 72 | 3 |

Project: Unspecified

Total Project Statistics

| | |
|---|---|
| Total Code Lines: | 949 (74.25%) |
| Total Comment Lines: | 27 (2.11%) |
| Total Mixed Lines: | 2 (0.15%) |
| Total Blank Lines: | 300 (23.47%) |
| Total Lines Altogether: | 1,278 |

**Figure 3:** Screen shot of LineTally

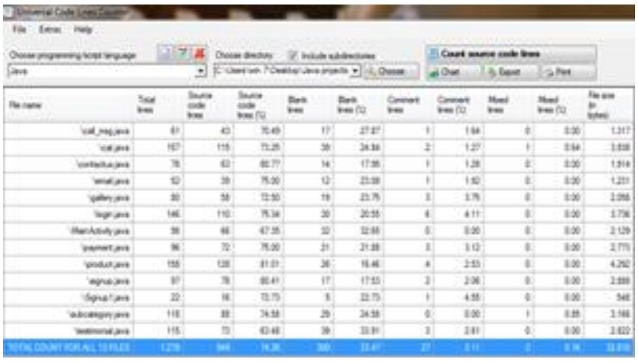

SourceMonitor - [Java Checkpoints In Project 'My restaurant' [Base Directory: 'C:\New Folder (2)\littleindia resturant\']]

| Checkpoint Name | Created | Files | Lines | Statements | % Branches | C... | % Comments | Classes | Methods/Class | Avg Stmts/Method | Max Complexity | Max Depth | Avg Depth | Avg Comp |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 27 Apr 2016 | 13 | 1,291 | 728 | 4.5 | 445 | 7.7 | 15 | 2.07 | 13.10 | 7 | 5 | 1.59 | |

Metrics Summary For Checkpoint 'Baseline'

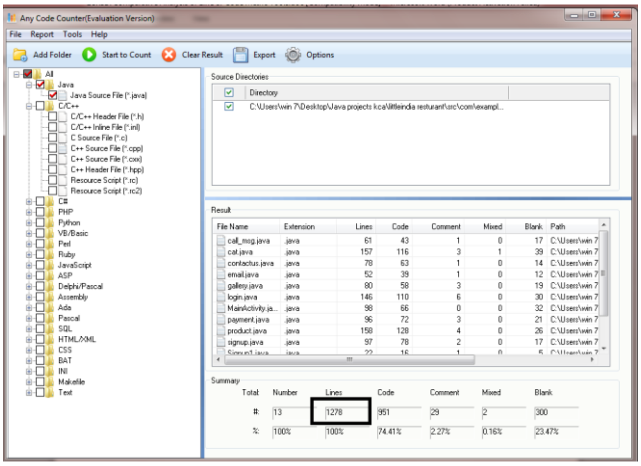
| Parameter | Value |
|---|---|
| Project Directory | C:\New Folder (2)\littleindia resturant\ |
| Project Name | My restaurant |
| Checkpoint Name | Baseline |
| Created On | 27 Apr 2016, 19:30:47 |
| Files | 13 |
| Lines | 1,291 |
| Statements | 728 |
| Percent Branch Statements | 4.5 |
| Method Call Statements | 445 |
| Percent Lines with Comments | 7.7 |
| Classes and Interfaces | 15 |
| Methods per Class | 2.07 |
| Average Statements per Method | 13.10 |
| Line Number of Most Complex Me... | {undefined} |
| Name of Most Complex Method | contactus.onClick() |
| Maximum Complexity | 7 |

**Figure 4:** Screen shot of SorceMonitor


Universal Code Lines Counter

| File name | Total lines | Source code lines | Source code lines (%) | Blank lines | Blank lines (%) | Comment lines | Comment lines (%) | Mixed lines | Mixed lines (%) | File size (in bytes) |
|---|---|---|---|---|---|---|---|---|---|---|
| \call_msg.java | 61 | 43 | 70.49 | 17 | 27.87 | 1 | 1.64 | 0 | 0.00 | 1,317 |
| \cat.java | 157 | 115 | 73.25 | 39 | 24.84 | 2 | 1.27 | 1 | 0.64 | 3,838 |
| \contactus.java | 78 | 63 | 80.77 | 14 | 17.95 | 1 | 1.28 | 0 | 0.00 | 1,914 |
| \email.java | 52 | 39 | 75.00 | 12 | 23.08 | 1 | 1.92 | 0 | 0.00 | 1,231 |
| \gallery.java | 80 | 58 | 72.50 | 19 | 23.75 | 3 | 3.75 | 0 | 0.00 | 2,056 |
| \login.java | 146 | 110 | 75.34 | 30 | 20.55 | 6 | 4.11 | 0 | 0.00 | 3,736 |
| \MainActivity.java | 98 | 66 | 67.35 | 32 | 32.65 | 0 | 0.00 | 0 | 0.00 | 2,129 |
| \payment.java | 96 | 72 | 75.00 | 21 | 21.88 | 3 | 3.13 | 0 | 0.00 | 2,775 |
| \product.java | 158 | 128 | 81.01 | 26 | 16.46 | 4 | 2.53 | 0 | 0.00 | 4,292 |
| \signup.java | 97 | 78 | 80.41 | 17 | 17.53 | 2 | 2.06 | 0 | 0.00 | 3,888 |
| \Signup1.java | 22 | 16 | 72.73 | 5 | 22.73 | 1 | 4.55 | 0 | 0.00 | 548 |
| \subcategory.java | 118 | 88 | 74.58 | 29 | 24.58 | 0 | 0.00 | 1 | 0.85 | 3,166 |
| \testimonial.java | 115 | 73 | 63.48 | 39 | 33.91 | 3 | 2.61 | 0 | 0.00 | 3,822 |
| TOTAL COUNT FOR ALL 13 FILES | 1,278 | 949 | 74.26 | 300 | 23.47 | 27 | 2.11 | 2 | 0.16 | [illegible] |

**Figure 5:** Screen shot of Universal Code Lines Counter



| ☑ | Directory |
|---|---|
| ☑ | C:\Users\win 7\Desktop\Java projects kca\littleindia resturant\src\com\exampl... |

| File Name | Extension | Lines | Code | Comment | Mixed | Blank | Path |
|---|---|---|---|---|---|---|---|
| call_msg.java | .java | 61 | 43 | 1 | 0 | 17 | C:\Users\win 7 |
| cat.java | .java | 157 | 116 | 3 | 1 | 39 | C:\Users\win 7 |
| contactus.java | .java | 78 | 63 | 1 | 0 | 14 | C:\Users\win 7 |
| email.java | .java | 52 | 39 | 1 | 0 | 12 | C:\Users\win 7 |
| gallery.java | .java | 80 | 58 | 3 | 0 | 19 | C:\Users\win 7 |
| login.java | .java | 146 | 110 | 6 | 0 | 30 | C:\Users\win 7 |
| MainActivity.ja... | .java | 98 | 66 | 0 | 0 | 32 | C:\Users\win 7 |
| payment.java | .java | 96 | 72 | 3 | 0 | 21 | C:\Users\win 7 |
| product.java | .java | 158 | 128 | 4 | 0 | 26 | C:\Users\win 7 |
| signup.java | .java | 97 | 78 | 2 | 0 | 17 | C:\Users\win 7 |
| Signup1.java | .java | 22 | 16 | 1 | 0 | 5 | C:\Users\win 7 |

| Total | Number | Lines | Code | Comment | Mixed | Blank |
|---|---|---|---|---|---|---|
| #: | 13 | 1278 | 951 | 29 | 2 | 300 |
| %: | 100% | 100% | 74.41% | 2.27% | 0.16% | 23.47% |

**Figure 6:** Screen shot of Any Code Counter

**Table 2:** Summarized Result of LOC Tools

| Sr.No | LOC Metric Tool/Features | LOC Count |
|---|---|---|
| 1. | LocMetrics | 1454 |
| 2. | LineTally | 1278 |
| 3. | SourceMonitor | 1291 |
| 4. | Universal Code Lines Counter | 1278 |
| 5. | Any Code Counter | 1278 |

## IV. CONCLUSION

This paper gives experimental studies on various free LOC tools. From the results obtained it can be concluded that the software programmers should not rely on single tool to arrive at final inferences and should consider other aspects also before making make decisions.

## V. REFERENCES

[1] B. Boehm, C. Abts, S. Chulani, “Software development cost estimation approaches: A survey”,Annals of Software Engineering, 2000.

[2] Park, Robert E., et al.. "Software Size Measurement: A Framework for Counting Source Statements"

[3] Bhatt, K., Tarey, V., and Patel, P. 2012. Analysis Of Source Lines Of Code(SLOC) Metric, International Journal of Emerging Technology and Advanced Engineering, Vol. 2, No. 5, pp. 150-154

[4] http://www.locmetrics.com/

[5] http://winsimple.hyperphp.com/

[6] http://www.campwoodsw.com/sourcemonitor.html

[7] http://www.ab-tools.com/en/service/forum/17-redirect/index.php?Page=0

[8] http://www.any-code-counter.com