

SOEN 6611: SOFTWARE MEASUREMENT

Milestone 4



|  |  |
| --- | --- |
| **Student ID** | **Name** |
| 40024381 | Amar Kumar |
| 40018973 | Charanpreet Kaur |
| 40013983 | Ajeeta Kaveti |
| 40021928 | Tarun Jeet Singh |
| 40019612 | Navnoor Multani |
| 40010983 | Lovish Tayal |
| 40031066 | Bitta Rani Rana |
| 40012480 | Mandeep Kaur |

Table of Contents

[Study 1](#_Toc478901127)

[Calculation of Maintainability from LOGISCOPE: 1](#_Toc478901128)

[Linear Regression Analysis: 3](#_Toc478901129)

[Pearson correlation for Maintainability 3](#_Toc478901130)

[References 7](#_Toc478901131)

[Appendices: 8](#_Toc478901132)

[Appendix 1: 8](#_Toc478901133)

[Appendix 2: 8](#_Toc478901134)

[Appendix 3: 9](#_Toc478901135)

List of Table

[Table 1 Value of Linear Regression 4](#_Toc478861014)

[Table 2 Value of Pearson coefficient 4](#_Toc478861015)

[Table 3 Validation of Hypothesis 7](#_Toc478861016)

List of Figures

[Figure 1 Maintainability jEdit5.1.0 2](#_Toc478861022)

[Figure 2 Analyzability jEdit5.1.0 3](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861023)

[Figure 3 Changeability jEdit5.1.0 3](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861024)

[Figure 4 Stability jEdit5.1.0 3](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861025)

[Figure 5 Testability jEdit5.1.0 3](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861026)

[Figure 6 CK Metrics vs Maintainability 4](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861027)

[Figure 7 WMC vs Maintainability 5](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861028)

[Figure 8 CBO vs Maintainability 5](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861029)

[Figure 9 DIT vs Maintainability 5](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861030)

[Figure 10 NOC vs Maintainability 5](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861031)

[Figure 11 LCOM vs Maintainability 6](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861032)

[Figure 12 RFC vs Maintainability 6](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861033)

[Figure 13 CC vs Maintainability 7](file:///C:\Users\Amar\Desktop\Study\SOEN%206611%20Measurement%20Piccirili\Submitted%20Milestone\M4_draft1.docx#_Toc478861034)

We analyzed the collected data and attempted to find the relationship between the CK metrics suite and the maintainability of different versions of jEdit software. In addition, we also tried to correlate Cyclomatic complexity of different versions of jEdit software. We investigated how the maintainability changes over different releases of jEdit. We implemented univariate linear regression. Apart from this, we worked on the calculation of Pearson correlation coefficient to determine the relationship of all metrics with maintainability.

# Study

We have analyzed different versions of jEdit software and thereafter calculated the maintainability of all the versions of jEdit with the help of logiscope tool. The Logiscope tool decomposes the maintainability factor into four criteria: Analyzability, Changeability, Stability and Testability.

The formula to evaluate the maintainability factor is:

Maintainability = Analyzability + Changeability + Stability + Testability

In our study we have calculated the maintainability for 26 versions of jEdit considering all classes namely, EXCELLENT, GOOD, FAIR and POOR in order to find the correlation between the metrics (C&K metric suite and McCabe cyclomatic complexity) and their maintainability. The maintainability has been considered as average maintainability of all classes in each version which is calculated using class metric level approach of Logiscope. . Higher the maintainability of a class, more difficult it is to maintain.

With this approach, we have achieved effective variations of maintainability for 26 versions of jEdit.

# Calculation of Maintainability from LOGISCOPE:

The below pie chart depicts the results of Analyzability, Changeability, Stability and Testability for version 5.1.0 of jEdit at class metric level

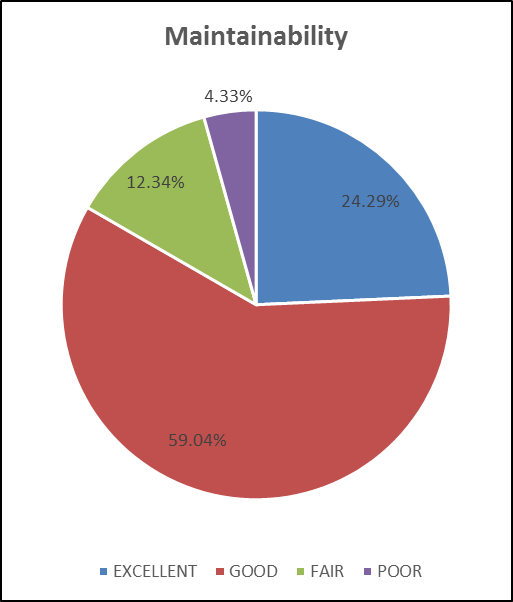


Figure 1 Maintainability jEdit5.1.0

Figure 2 Analyzability jEdit5.1.0

Figure 3 Changeability jEdit5.1.0

Figure 4 Stability jEdit5.1.0

Figure 5 Testability jEdit5.1.0

Using Logiscope, we analyzed all the classes of the jEdit for its different versions and measure the Analyzability, Changeability, Stability and Testability at the class metric level. The formula to compute the criteria is mentioned in *Appendix 1.*

LOGISCOPE defines the maximum and minimum bounds for each operands that are used in the calculation of Analyzability, Changeability, Stability and Testability. In a class, if the percentage of operands that are out of bounds is higher, then the maintainability will be higher which means that the class is difficult to maintain. *Appendix* *2* shows the minimum and maximum values and out of bounds of bounds percentage for metrics used to calculate average maintainability.

# Linear Regression Analysis:

On the basis of the measured values, we will correlate metrics in CK metric suite and McCabe’s Cyclomatic Complexity and their Maintainability*. Appendix 3* illustrates the measured values of CK metric suite and McCabe’s Cyclomatic Complexity and their Maintainability for jEdit software. *Figure 6* illustrates the variation of CK metric suite and McCabe’s Cyclomatic Complexity and average maintainability values for different versions of jEdit. It can be deduced that the maintainability of jEdit gradually decreases over the versions of jEdit.

Figure 6 CK Metrics vs Maintainability

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metrics | WMC | CBO | DIT | NOC | RFC | LCOM | McCabeCC |
| R square | 0.0153 | 0.1403 | 0.179 | 0.1615 | 0.0818 | 0.1904 | 0.2159 |

Table 1 Value of Linear Regression

# Pearson correlation for Maintainability

Pearson correlation is defined as a measure of linear association between two variables. We have assumed the metrics as independent variable and maintainability as dependent variable. The below table shows the Pearson correlation between C&K metrics and McCabe’s Cyclomatic Complexity and their maintainability.

|  |  |  |
| --- | --- | --- |
| **Correlation** | **Negative** | **Positive** |
| None | -0.09 to 0.0 | 0.0 to 0.09 |
| Weak | -0.3 to - 0.1 | 0.1 to 0.3 |
| Average | -0.5 to -0.3 | 0.3 to 0.5 |
| Strong | -1.0 to -0.5 | 0.5 to 1.0 |

Table 2 Value of Pearson coefficient [3]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metrics | WMC | CBO | DIT | NOC | RFC | LCOM | McCabe CC |
| R(Maintainability) | 0.123787 | 0.37457 | -0.42303 | -0.4019 | -0.28605 | 0.436307 | 0.4647 |

It has been illustrated in the above table that WMC, CBO, LCOM and McCabe cyclomatic complexity has positive correlation with maintainability which means that these metrics have direct impact on maintainability whereas DIT, NOC, RFC have negative correlation which means that there exists an inverse relation with maintainability.

Figure 7 WMC vs Maintainability

Figure 8 CBO vs Maintainability

In *figure 7*, it is shown that WMC has weak positive correlation (0.123787) with the maintainability. If the number of methods defined in a class increases, complexity of the class will increase which will require more testing and hence, maintenance becomes difficult. Hence, it will be more difficult to maintain. *Figure 8*, shows that CBO has average positive correlation (0.37457) with the maintainability. Therefore, if number of classes to which a class is coupled increases, there will be more dependency. Change in one class will affect the other classes which implies that high value of CBO means high change-proneness which further leads to more fault proneness. All the coupled classes need to be tested. Hence, high maintainability. This proves our hypothesis for WMC and CBO.

Figure 9 DIT vs Maintainability

Figure 10 NOC vs Maintainability

DIT and NOC are very closely related. From our studies, it is analyzed that DIT has average negative correlation with maintainability. This is illustrated in *Figure 9*. Hence, if the number of ancestors of a class increases, then it is easy to maintain. But this is true till a certain value. The desired value of DIT is 5. If DIT increases more than this then maintainability also increases. For our software, the average value of DIT is 3 for the latest versions which is good. Hence, the latest versions of jEdit are easier to maintain in terms of DIT. Furthermore, *Figure 10* represents that NOC has average negative correlation with the maintainability. The increase in number of direct descendants of a class will result in greater reuse of code which decreases the fault proneness and testability. Change in the parent class will automatically result in changes in the subclasses and hence easier maintenance. This validates our hypothesis.

Figure 11 LCOM vs Maintainability

Figure 12 RFC vs Maintainability

There is an average negative correlation of RFC with maintainability which is shown in *Figure 11*. Hence, if there is increase in number of response sets of a class, then the maintainability decreases gradually. High values of RFC indicate more faults and hence, the classes with high RFC value are complex and difficult to understand. So, with the increase in the value of RFC, maintainability should also increase. But this is not the case for our system. Hence, we reject the hypothesis.

*Figure 12* indicates that LCOM has positive correlation with the maintainability. High cohesion is always desirable. High value of LCOM indicates high complexity which increases the understandability. The value of LCOM increases for the latest versions of jEdit which means that the classes are more cohesive and require less testing. This also shows that they are less change-prone and fault-prone. Thus, easier to maintain. This proves our hypothesis.

Figure 13 CC vs Maintainability

*Figure 13* shows that McCabe’s cyclomatic complexity has average positive correlation with the maintainability. With the increase in complexity, testability increases. Complex classes are more fault and error prone. Changes to a class lead to greater complexity. Hence, classes with high cyclomatic complexity are difficult to maintain.

|  |  |
| --- | --- |
| Hypothesis | Results |
| Higher the value of WMC, higher the Maintainability(fault proneness, change proneness, testability) | Proved |
| Higher the value of CBO, higher the Maintainability(fault proneness, change proneness, testability) | Proved |
| Higher the value of DIT, lower the Maintainability(fault proneness, change proneness, testability) | Proved |
| Higher the value of NOC, lower the Maintainability(fault proneness, change proneness, testability) | Proved |
| Higher the value of RFC, higher the Maintainability(fault proneness, change proneness, testability) | Rejected |
| Higher the value of LCOM, higher the Maintainability(fault proneness, change proneness, testability) | Proved |
| Higher the value of McCabe CC, higher the Maintainability(fault proneness, change proneness, testability) | Proved |

Table 3 Validation of Hypothesis

# References

[1] Madhu Rohilla and P. K. Bhatia,” Prediction of Fault- Proneness Using CK Metrics,” in International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 8, August 2013).

[2] Aman Kumar Sharma, Arvind Kalia, and Hardeep Singh, “Empirical Analysis of Object Oriented Quality Suites,” in International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-4, April 2012. [3] Chidamber & Kemerer object-oriented metrics suite [Online]. Available: http://www.aivosto.com/ project/help/pm-oo-ck.html

[3] Tsantalis, Nikolaos. "Empirical Validation of Software Metrics" 2013. Lecture. Available: <https://moodle.concordia.ca/moodle/mod/resource/view.php?id=1460593>”

[4] Masoud Bozorgi et al.,"A Case Study on Quality Attribute Measurement using MARF and GIPSY," 2013, Concordia University Montreal, QC, Canada.

# Appendices:

## Appendix 1:

|  |  |
| --- | --- |
| **Analyzability** | cl\_wmc + cl\_comf + in\_bases + cu\_cdused |
| **Changeability** | cl\_stat + cl\_func + cl\_data |
| **Stability** | cl\_data\_publ + cu\_cdusers + in\_noc + cl\_func\_publ |
| **Testability** | cl\_wmc + cl\_func + cu\_cdused |
| **Maintainability** | Analyzability + changeability + stability + testability |

## Appendix 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mnemonic** | **Metric Name** | **Min** | **Max** | **Out** |
| **cl\_comf** | Class comment rate | 0.2 | +oo | 79.26% |
| **cl\_comm** | Number of lines of comment | -oo | +oo | 0.00% |
| **cl\_data** | Total number of attributes | 0 | 7 | 13.64% |
| **cl\_data\_publ** | Number of public attributes | 0 | 0 | 9.09% |
| **cl\_func** | Total number of methods | 0 | 25 | 2.56% |
| **cl\_func\_publ** | Number of public methods | 0 | 15 | 3.98% |
| **cl\_line** | Number of lines | -oo | +oo | 0.00% |
| **cl\_stat** | Number of statements | 0 | 100 | 14.49% |
| **cl\_wmc** | Weighted Methods per Class | 0 | 60 | 4.26% |
| **cu\_cdused** | Number of direct used classes | 0 | 10 | 31.82% |
| **cu\_cdusers** | Number of direct users classes | 0 | 5 | 11.36% |
| **in\_bases** | Number of base classes | 0 | 3 | 0.57% |
| **in\_noc** | Number of children | 0 | 3 | 1.99% |

## Appendix 3:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Version** | **CBO** | **WMC** | **DIT** | **NOC** | **RFC** | **LCOM** | **McCabe CC** | **Average Maintainability** |
| jEdit 3.0.0 | 5.098 | 50.947 | 3.203 | 0.218 | 31.722 | 273.075 | 13.36 | 47.41% |
| jEdit 3.0.1 | 5.128 | 51.444 | 3.203 | 0.218 | 31.85 | 282.767 | 13.33 | 47.17% |
| jEdit 3.0.2 | 5.128 | 51.459 | 3.203 | 0.218 | 31.857 | 282.767 | 13.33 | 46.96 |
| jEdit 3.1.0 | 5.116 | 53.217 | 3.449 | 0.203 | 31.855 | 302.428 | 13.15 | 46.9 |
| jEdit 3.2.0 | 5.51 | 54.824 | 3.765 | 0.235 | 33.039 | 320.889 | 13.72 | 47.21 |
| jEdit 3.2.1 | 5.516 | 54.915 | 3.765 | 0.235 | 33.072 | 321.405 | 13.76 | 47.68 |
| jEdit 3.2.2 | 5.536 | 55.078 | 3.765 | 0.235 | 33.157 | 321.542 | 13.83 | 47.66 |
| jEdit 4.0.0 | 5.597 | 58.276 | 3.602 | 0.221 | 33.696 | 356.144 | 15.11 | 47.71 |
| jEdit 4.0.2 | 5.597 | 58.354 | 3.602 | 0.221 | 33.724 | 355.945 | 15.16 | 47.67 |
| jEdit 4.0.3 | 5.597 | 58.365 | 3.602 | 0.221 | 33.724 | 355.945 | 15.16 | 47.67 |
| jEdit 4.1.0 | 5.898 | 62.273 | 3.631 | 0.262 | 35.588 | 360.219 | 15.74 | 47.67 |
| jEdit 4.2.0 | 6.153 | 67.07 | 3.721 | 0.233 | 37.772 | 399.414 | 16.35 | 47.67 |
| jEdit 4.3.0 | 5.957 | 57.955 | 2.402 | 0.163 | 29.682 | 381.314 | 25.68 | 47.7 |
| jEdit 4.3.1 | 5.957 | 57.985 | 2.402 | 0.163 | 29.69 | 381.449 | 27.7 | 47.7 |
| jEdit 4.3.2 | 5.959 | 58.009 | 2.402 | 0.163 | 29.684 | 382.583 | 27.71 | 46.95 |
| jEdit 4.3.3 | 5.959 | 58.032 | 2.402 | 0.163 | 29.684 | 382.563 | 27.72 | 45.85 |
| jEdit 4.4.1 | 5.85 | 56.948 | 2.402 | 0.163 | 29.463 | 376.879 | 25.46 | 46.4 |
| jEdit 4.4.2 | 5.856 | 57.046 | 2.402 | 0.163 | 29.517 | 377.469 | 25.53 | 46.4 |
| jEdit 4.5.0 | 5.802 | 56.297 | 2.345 | 0.166 | 29.307 | 363.147 | 25.49 | 46.4 |
| jEdit 4.5.1 | 5.806 | 56.347 | 2.345 | 0.166 | 29.337 | 374.374 | 25.56 | 52.92 |
| jEdit 4.5.2 | 5.81 | 56.384 | 2.358 | 0.166 | 29.352 | 374.497 | 25.58 | 52.92 |
| jEdit 5.0.0 | 5.907 | 57.327 | 2.397 | 0.165 | 29.954 | 385.224 | 26.28 | 52.84 |
| jEdit 5.1.0 | 5.948 | 57.665 | 2.397 | 0.165 | 30.05 | 387.919 | 26.5 | 52.24 |
| jEdit 5.2.0 | 5.988 | 58.192 | 2.425 | 0.165 | 30.224 | 394.157 | 27.31 | 52.82 |
| jedit 5.3.0 | 5.992 | 58.275 | 2.438 | 0.167 | 30.526 | 395.612 | 27.35 | 52.82 |
| jedit 5.4.0 | 5.977 | 57.564 | 2.506 | 0.167 | 30.882 | 386.837 | 27.44 | 52.48 |