D.I.K. Rajapakshe IT16178700, P.M.C.P. Paththinisekara IT17056212, M.P.P. Shamil IT17018760, and

S.K. Liyanage IT17152938

Code Complexity Measuring Tool

*Abstract*— **For our project, we decided to choose web Application for the code complexity tool requested that CEO of ABC Solution. This project is aimed at developing an Automated Code Complexity Measuring Tool that is optimize developer and quality engineer workload. The general nature of the software is uploading code to the software and measuring complexity. At present, they only concentrate on developing C++ and Java based projects. The primary objective of ABC solutions is to build quality software that could be easily maintained. In an attempt to reduce the maintenance cost of the software developed by the company. This project was selected as they were in need of a comprehend automated system to manage and replace human power which consumes less time and fewer resources having a higher accuracy.**

**Moreover, the procedure involved in the ABC Solution is difficult to handle where they currently developing C++ and Java base system for developing. They requested to measure the complexity introduced due to the following factors. Size, Type and the nesting level of control structures, Inheritance and Recursion. So, they cannot get decision immediately and wasting their workable hours for manually measuring code complexity of the code. The primary need of the client was to view data efficient in a well-ordered manner, restrict employee works and reporting capabilities. As a solution, we decided to develop web application to solve their problems including all the client requirements.**

***Index Terms*—Code, Complexity, Measuring, Tool, Web Application, Code Size, Type and the nesting level of control structures, Inheritance and Recursion, Java, C++.**

1. INTRODUCTION

A BC Solutions is a startup and is evolving rapidly. At present, they only concentrate on developing C++ and Java based projects. The primary objective of ABC solutions is to build quality software that could be easily maintained.

In an attempt to reduce the maintenance cost of the software developed by the company, we developed code complexity measuring tool for their primary needs C++ and Java language. In our tool users can upload their code and measure Code size, Type and the nesting level of control structures, Inheritance and Recursion. Code complexity correlates with the defect rate and robustness of the application program.

Code with good complexity contains less errors, is easier and faster to test, is easier to understand, is easier to maintain. In Many software, they measure code complexity separate way to java and C++. E.g.:- verifysoft. In our software, we developed to measure for both language without developed multiple tools.

So, in the future with the help our system they will be able to handle measure code complexity and reach their business goals successfully.

Since they have high competition in this business field, they request a system that can solve above problems at the conference we had. So, we decided to develop a system reduce complicated manual work by computerize code complexity and code details etc. And handle the process easily in a secure manner.

The main objective of Creating the Code Complexity Measuring Tool is to implement all processes and manage the working process in a more efficient way.

This system covers all key processes that are currently needs of mentioned. And this tool will automate the manual processes done by the users in the ABC Solution and according to their suggestions this system would be a great help to all the work.

In Code Complexity Measuring Tool, there are different measuring levels considering the overall process. Such as Code Size, Type and the nesting level of control structures, Inheritance, and Recursion.

This system will give benefits through increased efficiency and effectiveness. It also shows the commitment to increased performance, developer and quality engineer satisfaction, and continuous improvement. To build that we are using java as the programing language, and Apache Tomcat Server. The source code will be created in the software Intellij IDEA.

It provides details on the requirement specification, analysis, application modeling, design, testing and implementation future scope and limitation of the application development. The importance of this system to organization is Efficient data access.

The software helps them maintain day to day transaction in computer and it could lead business to its success.

1. RELATED WORK/ LITERATURE REVIEW

The primary object of ABC solutions is to build quality software that could be easily maintained and build quality project. So, user wants to upload the code and check software complexity. They can improve their coding using this tool all the numbers want to implement UI parts to view their calculation values in generating complexity tool app.

Measuring code complexity tool user wants to upload the code and check software complexity. They can improve their coding. Measuring total complexity of a program statement is addition of measure of number of types of control structures, nesting level of control structures, and inheritances. It wants calculate complexity of a program statement. Measuring recursion is total complexity of a program statement power to the two. It supports to calculate complexity of a program.

Measuring complexity of a program statement is addition of complexity of a program statement and value of recursions measure. This tool is viewing complexity of a program value so developer and QA engineers can identify the quality of project code and can improve the quality of code. Measuring recursion is complexity of a program statement power to the two. It supports to calculate complexity of a program. That helps to get overall benefit of the software.

The complexity of all the program statements which belongs to a class is assigned the same weight that the class has due to its inheritance. Measuring the complexity of a program statement due to size indicate operators, Key words and Numeric Values. The weight allocated for the program statements is increased by one for each level of nesting. It wants to calculate total weight of the code.

Calculating complexity of inheritance helps to measure complexity of the code. Users who using this tool can check complexity user want upload code file to the software. So, user wants to read uploaded the code and check software complexity.

Also want to implement read file which upload project code to measure to complexity of the code.

Implement read UI uploaded file to measure complexities can improve their coding using this tool all the numbers want to implement UI parts to view their calculation values in generating complexity tool app. Also want to implement read file which upload project code to measure to complexity of the code.

1. PROPOSED SYSTEM

Once the code of the program uploaded to the tool it will automatically calculate the code complexity using various factors and also, it will show the results in a table and a chart according to the main factors to make it easier for the user and also the system consist of features that allows the users to download the computed complexity results in various formats and print them accordingly.

1. *Upload a Code file*.

The complexity measuring tool was developed so that you can upload a code file to the tool from the computer through the browsing window. The file reader included in this tool is not only able to read just common text files but also the tool can read word(docx), text(txt), Java(java), C++(cpp) file formats without any issues.

Not only the file reader in this complexity measuring tool just read the code file but also, it’s able to format any kind code according to the standard to be used inside the tool while it measuring complexity. This reformatting includes erasing all the unnecessary comments (multi line and single line) because the comments are irrelevant when measuring complexity, reformatting all the curly brackets which are not according to the coding standards, removing all the unnecessary tabs and unnecessary empty spaces within the code specially the tabs and spaces on beginning of code lines which is unnecessary during the process of measuring complexity. and etc. So, there is no need to manually adjust the code which you want to measure complexity of according to the requirements of this tool. So, this unique upload functionality improves the uniqueness of the code. And the uploading and analyzing time is not considerably increased by this unique reformatting feature.

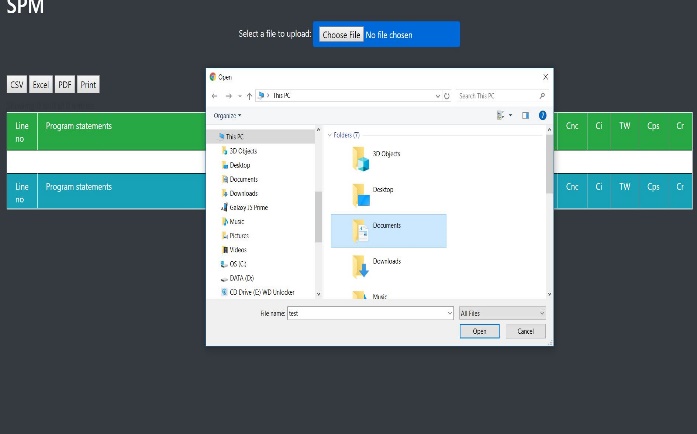
****

Fig. 1. File browse and uplaod interface.

1. *Compute complexity*.

After reading the code file to be analyzed in order to measure complexity the tool will read line by line of the code in order to compute the code’s complexity under four major factors, they are,

1. *Size factor-*

Complexity of a program statement due to size (Cs).

1. *Type and the nesting level of control structures-*

Complexity of a program statement due to type of control structures (Ctc).

1. *Inheritance-*

Complexity of a program statement due to nesting of control structures (Cnc).

1. *Recursion-*

Complexity of a program statement due to inheritance (Ci).

The tool will compute complexity according to those actors and it will display the complexity of the uploaded program line by line in a table. The table displays the line no, complexity according to the above factors, total complexity of a program statement (Tw), complexity introduced due to recursion (Cr) and complexity of a program (Cp).

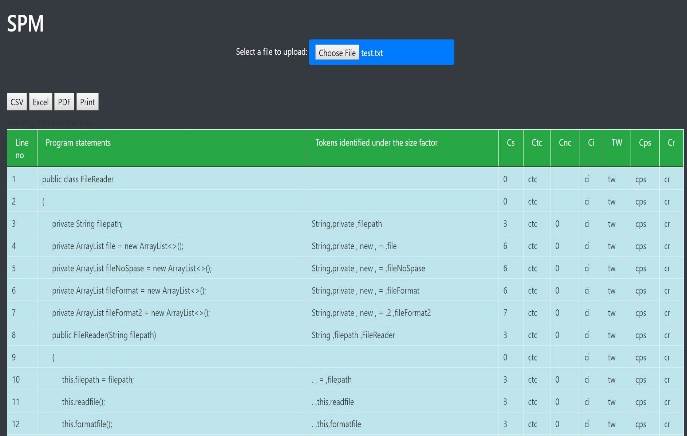
****

Fig. 2. Complexity table.

1. *Complexity Chart.*

The complexity chart is a feature that was implemented for the easy use of the user. This chart can be used to get a better understanding of the complexity variations of the uploaded code which was analyzed. This chart displays analyzed data according to the complexity table that was created before. It displays same data on the table line by line according the four-complexity factor that was mentioned before.

So, the user can easily compare the complexity data by factors side by side. And also, the charts display various factors in different colors which can be easily comparable. And it also has the option to display factors one by one or two or three without the whole set. That function was added so to make the chart less complex. So, this feature is also making this complexity measuring too unique.

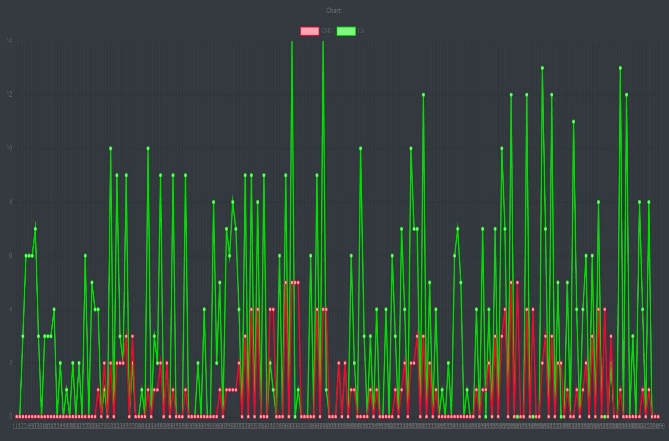
****

Fig. 3. Comlplexity chart interface.

1. *Export complexity table.*

Download feature is also an extra feature that implemented make the users work easier by giving them the ability to export the latest analyzed code’s code complexity table. The code can be exported using three formats which is CSV, PDF and Excel which enable the users to directly use exported results directly in other applications that supports the file formats which are mentioned above. And also, since exports can be done in PDF format the export can be directly considered as a complexity report of that particular code. This feature also can be considered as a unique feature of the system considering other similar tools Since most of the tools and systems out there doesn’t have ability to export several different formats. So, it improves the uniqueness of this tool.

1. *Print feature.*

The print feature is a function that can directly print the complexity table after analyze code complexity of a particular code. This enables the user to print their latest analyzed code directly. And the other important thing is this process can be done without exporting the result. And also, it seems most of the other system and tools doesn’t exactly focus their attention on printing the table including complexity data as it is directly it also increases the uniqueness of the system

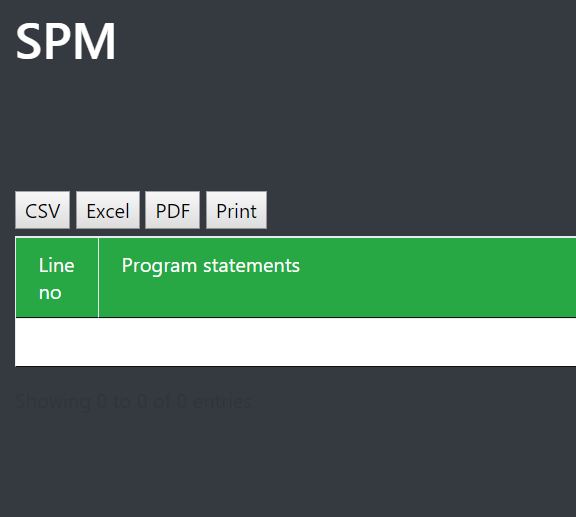


Fig. 4. Exporting and printing.

1. METHODOLOGY

In the development of this code complexity measuring tool there were few technologies and tools had to be used in order to complete successfully. And also, there were different methods had to be followed in the process of derive the complexity of a program due to each factor.

1. *Technologies and tools used.*

In order to develop this tool, there were sustain technologies and tools that had to be used to make this tool well designed complexity measuring tool.

In terms of technologies in the process of designing backend Java language was used. More specifically Java JDK 8 was used when creating the backend. So, it’ll require Java runtime environment 8 or above version to execute the tool without any issues. Since the backend is java and the frontend are not a complicated part in this tool it was logical to use JSP as the frontend. So, this tool is a web-based application since the front end designed using JSP. So, a server environment is required to run the application. So, as the server the Tomcat server was used. Tomcat server 8.5 or above version will be sufficient to run the code. And as an IDEA the Intellij IDEA was used to develop this complexity measuring tool. Since this project was a maven project few maven dependencies were used.





Fig.5. Used dependencies

And also, few other tools were used in order to make the development process easier. They are as follows,

1. *SonarQube-*

SonarQube tool was used for continuously inspect the code quality to receive automatic reviews with static analysis of code to detect bugs and code smells. SonarQube was used mainly to make sure that the final product is a reliable one. Since the SonarQube can identify duplications of the code accurately with percentage value it also affects the coding standards in a good way too.

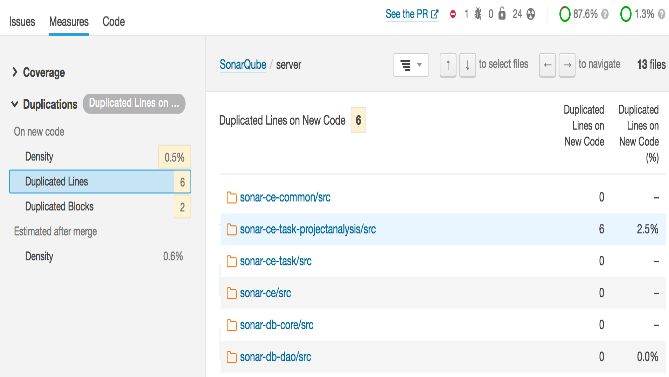


Fig.6. SonarQube

1. *TargetProcess-*

Targetprocess was used to manage project according to Agile method. Because it’s very important to manage this project collaboratively. So, it’s easy to manage a project visually when all the members update the completion status implementation of relevant functions accordingly.

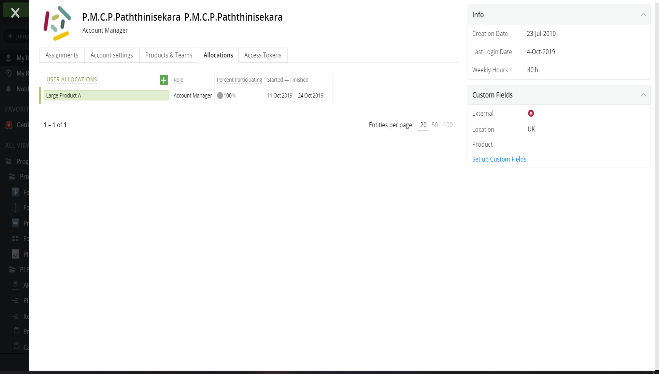


Fig.7. *TargetProcess*

1. *GIT-*

GIT tool used for version control of the entire project. So, GitHub repository was created to upload each member’s functions according to their completion time to the relevant Git branches and finally merge all the branches for the final product.

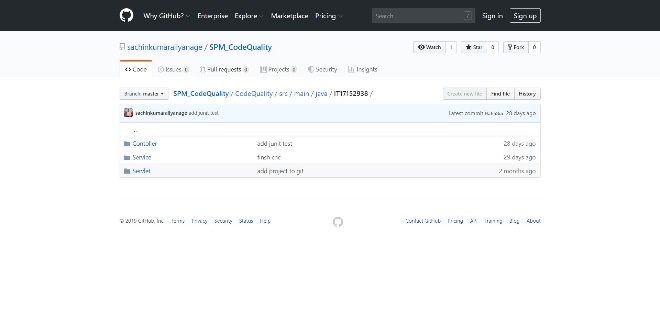


Fig.8.GitHub

1. *Selenium-*

Selenium was used as a testing tool for this complexity measuring tool. Since this is a web application it was logical to use selenium since it’s an automated testing tool.

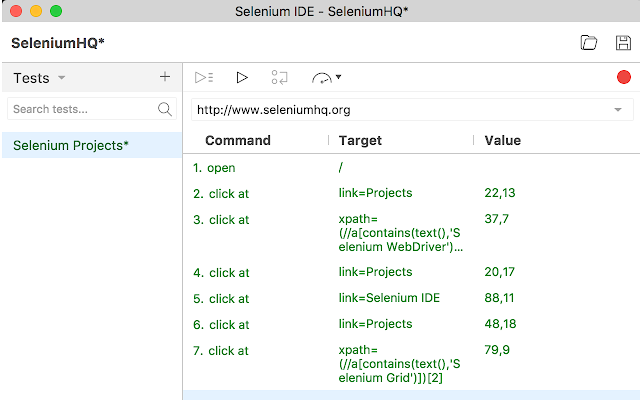


Fig.9. *Selenium*

1. *Approaches used to derive the complexity of a program*.

In order to measure the complexity in a given program four factors had to be considered and implemented functions to measure them using various logical methods.

1. *Size (Cs)-*

Various java functions were implemented to measure arithmetic operators, relational operators, logical operators, Bitwise operators, miscellaneous operators, Assignment operators, key words, manipulators, text inside a pair of double quotes, Classes, method, object, variable, and array names and Numeric values (numbers) as separate items.

|  |  |
| --- | --- |
| Arithmetic operators | + - \* / % ++ -- |
| Relation operators | == != > < >= <= |
| Logical operators | && || ! |
| Bitwise operators | | ^ ~ << >> >>> <<< |
| Assignment operators | += -= \*= /= = >>>= |= &= %= <<= >>= ^= |
| Key words | void, double, int, float, string, printf, println, cout, cin, ‘if’, ‘for’, ‘while’, ‘do-while’, ‘switch’, ‘case’ etc. |
| Manipulators | ‘endl’, ‘ \n’, etc. |
| Text inside a pair of double quotes | Eg: “The greatest is” |
| Class, method, object, variable, and array names | |
| Numeric values (numbers) | |

Table.1.Size values

Those were counted on line by line basis and constant value of one was added for each one of those that was detected most of those were detected using regex patterns. These methods had to be programmed to measure complexity of both c++ and java code. So, some operators like (\*) sign had to be disregard in c++ pointers because it’s not a deference operator there. And constant value of two had to be programmed to counting Reference (&) and dereference (\*) operators and ‘new’, ‘delete’, ‘throw’, and ‘throws’ key words. Finally, all of those results of individual counting methods get as a total size value of a particular line. And the total of those size factors of all the lines considered the total size factor of the code.

1. *Type of control structures (Ctc)-*

To measure the complexity of given code due to type of control structures. Java measuring functions were implemented as several different logical methods. First checked whether an ‘If’ condition is in a given line of code and weight of one was assigned. And next for each logical (‘&&’ and ‘||’) or bitwise (‘&’ and ‘|’) operator found in that particular if condition weight of one was added. And if a program statement consist of iterative control structure such as a ‘for’, ‘while’, or ‘do-while’ loop the weight of two was assigned and as the same as in the ‘if’ condition each logical (‘&&’ and ‘||’) and bitwise (‘&’ and ‘|’) operators were took in to account but in this time the weight was two. If a program statement consists of a ‘catch’ keyword weight of one was added to that line. In ‘switch’ statements number of cases inside was considered as the weight. And most importantly if a statement doesn’t meet any of the above conditions weight of zero was added. Finally, the combine count of all the statements in the code considered the total type of control structures factor of the code.

1. *Nesting level of control structures (Cnc)-*

This factor computes the complexity using the nesting level of the code. So, if a program statement doesn’t have any level of nesting the tool will consider that statement have a weight of zero. When the nesting level is at the outer most level weight of one was considered. And also, weight of two was added for next inner level of nesting. Then the complexity of each statement will be increased according to each level of nesting by one. And the total of all the statement complexity count was considered as the Nesting level of control structures factor.

1. *Inheritance (Ci)-*

This factor computes the level complexity using the inheritance in the code. The tool was designed to consider complexity of all the program statements which belongs to a class is assigned the same weight that the class has due to its inheritance (CCi). So, every statement in a given class has the same level of inheritance. So,

|  |
| --- |
| *Complexity of a class due to its inheritance (CCi) = Number of ancestor classes of the class + 1* |

Table.2. Complexity of a class due to inheritance.

1. *Recursion (Cr)-*

Measuring recursion require the functionalities of all the other factors because the recursion is simply the repetition. So, it required total complexity of a program statement (Cps) and multiply it by two and to get Cps it requires to multiply total weight of the statement (Tw) and the size (Cs). And the total weight is a combination of Ctc and Cnc and Ci. But if the code doesn’t consist of any recursive methods the program will disregard the calculation of recursion.

1. *Final complexity value (Cp).*

Final complexity value of a given code is mainly depending on two factors. Total complexity of a program statement and Recursion in the code. So, depend whether the code has recursive methods Total measurement of a program complexity varies. if program consist of recursive methods Complexity of the program will be Addition of all total Cps values which doesn’t belong in a recursive method to total of Cr values derived for the program statements that belongs to a recursive method. If program does not consist of recursive methods Cp is considered as the addition of the ‘Cps’ values of all the program statements in a program.

1. *Other special or unique features.*
2. *Complexity Chart-*

The complexity chart is a unique feature that was implemented for the easy use of the user. This chart can be used to get a better understanding of the complexity variations of the uploaded code which was analyzed. This chart shows all complexity factors in the program and their variations according to the program statements in different colors. This chart was developed using charts.js JavaScript library.

1. RESULTS AND DISCUSSION

When implementing most of the features specially feature that require to locate specific Strings it was convenient at the moment to use regex patterns but later realized that it wasn’t always usable because in a situation such as a when keyword name becomes a part of a variable name or function name regex doesn’t always give perfect results.

Some operators like (\*) sign had to be disregard in c++ pointers because it’s not a deference operator there when measuring size factor.

Inheritance complexity measurement in this tool has some level of similarity to [1] ‘Average Depth of Class in Hierarchy Tree.’ But there are some differences because of the calculations they used. And also, they consider the inheritance measurements are part of Size factor for the sustain moments.

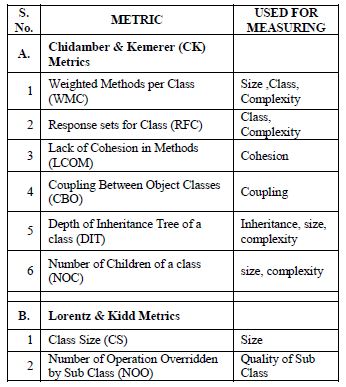


Table.3 Metrics with respect to their use according to [1]

And when calculating the method complexity this tool takes complexity of each statement in order to get the final result but ‘The proposed metric for object-oriented programming.’ [2] get the a class complexity using another equation.



Fig.10. Class complexity equation

1. Conclusion

Speaking of the article, we should say that the complexity measuring tool was developed that can upload any code file such as common text files, word(docx), text(txt), Java(java), C++(cpp) file formats without any issues to the tool from the computer through the browsing windows. Actually, this complexity measuring tool is not a tool that just measure complexity. It formatting the code to the coding standards before measuring complexity. So, it removes comments, reformatting all the curly brackets which are not according to the coding standards from the beginning stage after uploading the code to the tool. After uploading the code file in order to measure complexity the tool will read line by line of the code in order to compute the code’s complexity under size factor (Cs), Type and the nesting level of control structures (Ctc), Inheritance (Ci), Recursion (Cnc). The tool will compute complexity of the uploaded program line by line in a table. In this tool there is a special feature it is the complexity chart. That implemented for easy to use for the user. This chart can be used to get a better understanding of the complexity variations of the uploaded code which was analyzed. This chart displays the calculated complexity of earlier stated categories and display the values by line by line as a graph. It is very use full thing to user because he/she can see the most valuable complexity line of the code. User can download the CSV, PDF and Excel and other different formatted calculated complexity files from here. It was the extra feature. The other thing is the print feature, that is a function that can directly print the complexity table after analyze code complexity of a particular code. Finally, the main language use to develop this tool is JAVA with Java JDK 8 was used to creating the backend and JSP used for a front end. This tool is runs in Tom Cat 8.5 server and the maven dependencies are used for it.

When you are use this tool, you feel this kind of tools are very easy and user friendly to measure the complexity of code. Once you run the tool, we know definitely you feel better because of the features in the tools.

# References

|  |  |
| --- | --- |
| [1] | Meenakshi Kandpal,Anmol Kandpal, "Critical Analysis of Traditional Size Estimation Metrics for Object Oriented Programming," vol. 58– No.13, November 2012. |
| [2] | SANJAY MISRA, IBRAHIM AKMAN ,MURAT KOYUNCU, "An inheritance complexity metric for object-oriented code," Vols. 36,Part 3, June 2011. |