**Abstract**

The technological advancements have influenced the society to take a leap towards success. Every technological reform is a small step towards advancement and progress of mankind. Development in information have also been impacting upon educational institutes. The utilizes computers and internet. But the available technology is not used in a way that can achieve more or in a way which will increase the productivity. This system will support the environment strategic goals and directions. But this discussion is planned to furnish driving force in exploration and help excite further interest. In to this research is about a tool designed and developed to detect the complexity of both java and C++ programs by linking weights to different characters of classes in the different or same file, due to complexity of calculating final value according to the Size, Methods and Variables, Inheritance, Coupling, and Control Structures. Eventually, the total complexity of the program is calculated.

The important of of measure the coupling and cohesion of the modules in the codebase as well in order to get a true system-level, software complexity grater predictability knowing the level of complexity of the code being maintained makes it easier to know how much maintain a program will need.

Most traditional measures used to quantify software complexity. They are simple, essay to count, and very easy to understand. They do not, however, take into account the intelligence content and the layout of the code. There are many benefits of software complexity. There are improve code quality, Reduce maintenance cost, Heighten productivity, Increase robustness, Meet architecture standards.

There are many various complexity tools. There were many attempts to measures software complexity but they all failed in some way. For example

* Line of Code
* Cyclomatic complexity
* Assignment, Branches and Condition Metric.

Code Development tools consists of mainly programming languages, Debugging tools, Version Management tools compiling tools and Integrated development <https://www.tldp.org/HOWTO/Scientific-Computing-with-GNU-Linux/devtools.html><https://www.researchgate.net/figure/Main-features-of-the-developed-tool_fig4_322506435>

**Introduction**

First we thank for lecturer to give us this topic of project. It is a new side of the projects. We do projects in last semesters. But this project is very important for us. Also in this project about deeply get things in this. Actually different projects in YouTube, internet etc. But in this type projects directly cannot see in the internet. Also, How to build this project? , How to run this project? , What advantages of project? etc information get by internet indirectly. It is an accustomed measurement tool to test their ASCII document. Then they will reduce complexity and maintenance costs within the long run. It is a controversy within the software development lifecycle.

In this topic, we build a Desktop Application. It will support many of the activities in the Application. We use java language and create interfaces to use netbeans software. We includes size, variables, methods, inheritance, coupling and control structured in this project. Our system will give many solutions to the problems that they are currently facing.

**Complexity of a program statement due to size.**

The user is able to measure his submitted code according to the size component through the web base code measuring tool which the complexity is focused on Java.

**Complexity of a program statement due to variable.**

Only the lines which consist of declared or defined variables are considered under this factor. Scope, primitive data type and composite data type variables are considered when computing.

**Complexity of a program statement due to method.**

Only the lines which include method signatures are considered under the methods factor.

**Tasks of the developer under the size, variable, and method**

Firstly, the developer wants to acquisition the software. After that start to analyze the complexity of a program statement due to its keywords, identifiers, operators, numerical values, and string literals.

Actually people should care about our project. There are five reasons we has measure software complexity. In IT industry, you can’t manage what you can’t measure. Also knowing how an organization’s application portfolio is provides insight into how to manage it best. The problem is the issue that comprise software complexity. The multiple system interfaces and complex requirements, the complexity of the systems sometimes grows beyond control, rendering application and portfolios too costly to maintain ad risky to enhance.

Complexity Metrics

Fortunately, there have been many methods developed for measuring software complexity. Most break down software complexity according to the following metrics.

Cyclomatic Complexity

Programs with more conditional logic are more difficult to understand, therefore measuring the level of cyclomatic complexity unveils how much need to be managed. For example If,do,select etc. In the first case complexity metrics will look bad. In second, the complexity metrics will look good, but the result will be deceptive. It is important, therefore, to measure the coupling and cohesion of the modules in the codebase as well in order to get a true system-levels, software complexity measure.

Halstead Volume

We are measuring how much “information” is in the source code and needs to be learned.

Maintainability index

Maintainability index is more of an empiric measurement, having been developed over a period of years by consultants working with Hewlett-Packard and its software teams.

Organizations that have this information can capitalize in a number of ways.

1. Greater predictability
2. Software Risk Mitigation
3. Reduced Cost
4. Extended Value
5. Decision Support

<https://www.castsoftware.com/blog/five-reasons-to-measure-software-complexity>

Complexity aries when the dependencies among the elements become important. The behavior of many complex systems emerges from the activities of lower-level componets. Typically, this emergence is the result of a very powerful oraganizing force that can overcome a variety of changes to the lower-level components.

There are four proposed tools to use. There ara sonqube, selinyam, targetprocess and git. In this desktop application we can use sonqube, targetprocess and git.

**Result and Discuss**

**2.1** **Complexity of program due to size.**

Cs = (Wkw \* Nkw) + (Wid \* Nid) + (Wop \* Nop) + (Wnv \* Nnv) + (Wsl \* Nsl)

**2.2 Complexity of program due to variables**.

Cv = Wvs **[**(Wpdtv \* Npdtv) + (Wcdtv \* Ncdtv)**]**

**2.3 Complexity of program due to methods.**

Cm = Wmrt + (Wpdtp \* Npdtp) + (Wcdtp \* Ncdtp)

**2.4** **Complexity of program due to coupling.**

Ccp = (Wr \* Nr) + (Wmcms \* Nmcms) + (Wmcmd \* Nmcmd) + (Wmcrms \* Nmcrms) + (Wmcrmd \* Nmcrmd) + (Wrmcrms \* Nrmcrms) + (Wrmcrmd \*Nrmcrmd) + (Wrmcms \* Nrmcms) + (Wrmcmd \* Nrmcmd) + (Wmrgvs \*Nmrgvs) + (Wmrgvd \* Nmrgvd) + (Wrmrgvs \* Nrmrgvs) + (Wrmrgvd \* Nrmrgvd)

**2.5 Complexity program due to control structure.**

Ccs = (Wtcs \* NC) + Ccspps

The result will be out come of the funtions.