Noakhali Science and Technology University

Institute of Information Technology

(Software Engineering)

Session: 2019-2020

Course: Software Metrics Lab

Code: SE 3205

Submission Date: 26/12/2023

Lab Report: Code Size & Code Structure

Prepared For:

Md Hasan Imam

Lecturer

Institute of Information Technology

Noakhali Science and Technology University

Prepared By:

Shoriful Habib MUH2025003M Year-3 Term-2 shoriful2515@student.nstu.edu.bd	Sanjida Akter Samanta BFH2025010F Year-3 Term-2 samanta2515@student.nstu.edu.bd
Sumaiya Begum BKH2025015F Year-3 Term-2 sumaiyabegum2515@student.nstu.edu.bd	Mehedi Hasan MUH2025032M Year-3 Term-2 mehedi2515@student.nstu.edu.bd

Table of Contents

List Of Figures:

Determining Code Size for SPL-II Projected:

Lines of Code (LOC)

Definition	Lines of code are the "source code" of the program, and one line may generate one machine instruction or several depending on the programming language
Measurement Procedure	Automated Program
Value	15244

Commented lines of code (CLOC)

Definition	A comment is a programmer-readable explanation or annotation in the source code of a computer program.
Measurement Procedure	Automated Program
Value	5600

Non commented lines of code (NCLOC)

Definition	The number of physical lines that contain at least one character which is neither a whitespace nor a tabulation nor part of a comment.
Measurement Procedure	Manually
Value	9644

Executable Lines of Code(ELOC)

Definition	Executable Lines of Code (ELOC) is the number of executable lines of code in a class (component) or a function.
Measurement Procedure	Automated Program
Value	8296

Blank Lines of Code

Definition	Blank Lines represent lines without any statement or symbol. They are present in code to increase readability and clarity.
Measurement Procedure	Automated Program
Value	1348

Density of comments

Definition	Comment density is the percentage of comment lines in a given source code base, that is, comment lines divided by total lines of code.
Measurement Procedure	Manually
Value	5600/ 8296=0.6750

Halstead's Approach

Definition	Halstead's theory is an analytical estimation technique to measure the size, development effort, and development cost of software products
Measurement Procedure	Size of vocabulary η =η1 + η2 where, η = number of vocabulary in a program η1 = number of unique operators η2 = number of unique operands Length of program N = N1 + N2 Where, N = Length of program N1= Total occurrences of operators N2 = Total occurrences of operands Program Volume (V), V= N x log2η
Value	n1: 332 n2: 12991 N1: 13881 N2: 44852 Program Length (N): 58733 Vocabulary Size (n): 13323 Program Volume (V): 539214.5666963514

Number of bytes of computer storage

Definition	Number of bytes used in the computer storage for the program text.
Measurement Procedure	Manually
Value	876,895,133

Number of characters

Definition	Character is alphabets of written code.
Type	Automated program

Value	1332596

Average number of characters per Class

Definition	Average number of characters per class. Average character = total characters/ total class
Type	Manually
Value	1332596/16=83287.25

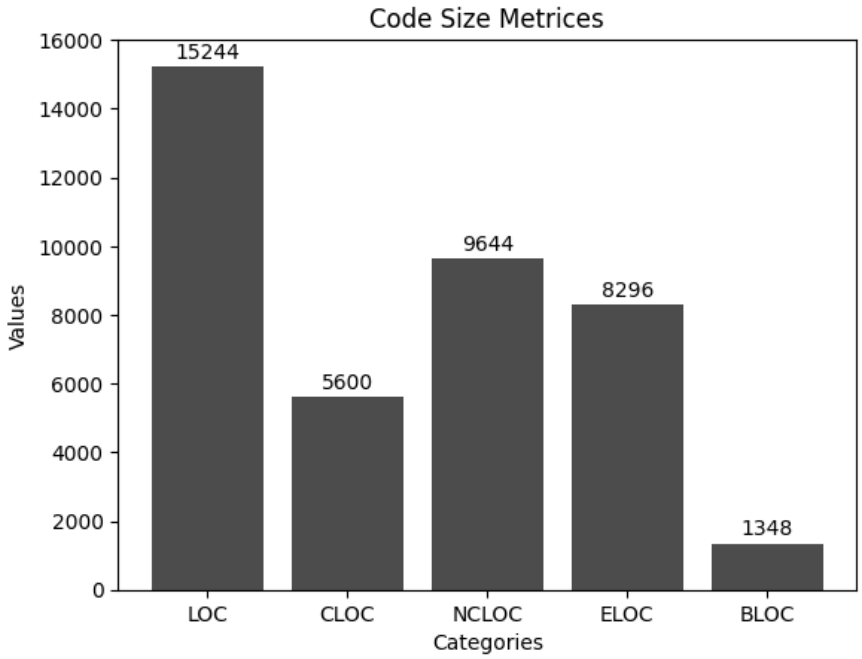

Figure 1: Code Size

Class	LOC	CLOC	NCLOC	BLOC	ELOC
Change.php	63	3	53	7	53
Code.php	140	47	72	21	72
CodeLogin.php	51	2	42	7	42
DbConnection.php	3	0	3	0	3
Dynamic.php	139	47	79	13	79
Education.php	72	3	62	7	62
Index.php	165	13	132	20	132
LogIn.php	97	1	85	11	85
Logout.php	5	0	5	0	5
Predict.php	12	4	7	1	7
Process.php	95	6	76	13	76
Register.php	83	1	64	18	64
Txttosign.php	96	2	83	11	83
Updateprofile.php	66	1	53	12	53
Final-pre.py	588	113	475	34	475
Sign.py	14	0	12	2	12

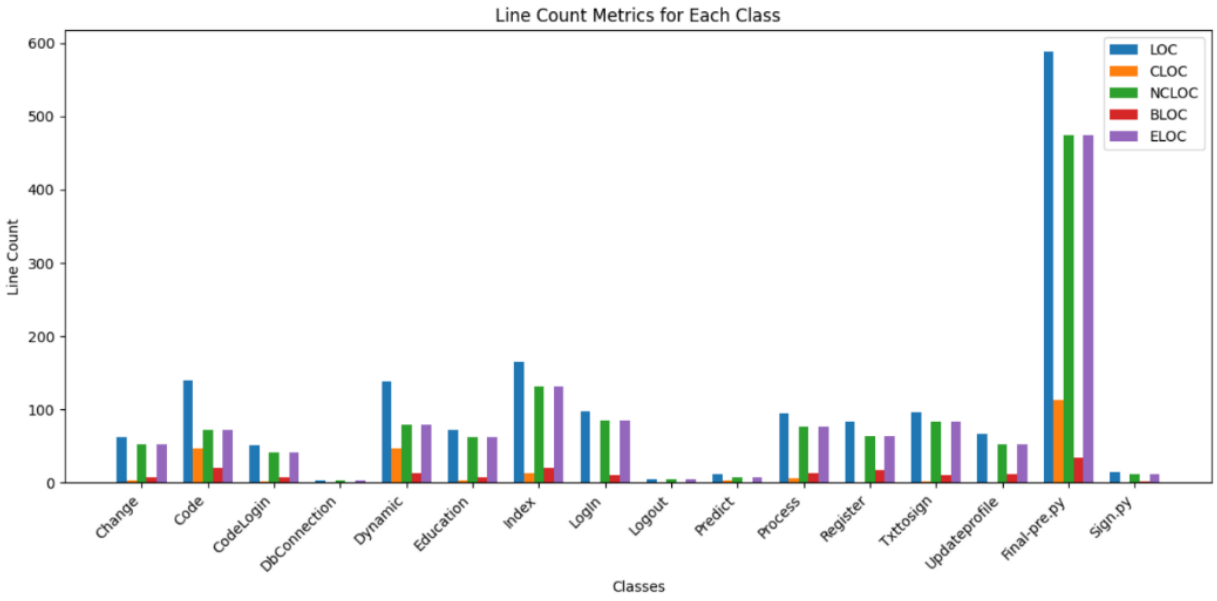

Figure 2: Code Size per Class

Code Structure Measurement:

The estimation of a software's structural qualities is crucial for both the product's maintenance and the development effort. Understanding the complexity involved in changing one product into another, testing a product, or forecasting the external software attributes from early internal product measures is made easier by looking at the structure of requirements, design, and code.

Types of Measurement:

The structure of software can be measured using

- **Control-flow structure** − It is the sequence in which instructions are executed in a program.
- **Data-flow structure** − It is the behavior of the data as it interacts with the program.

We have used Cyclomatic Complexity measurement to understand the complexity of our system.

Cyclomatic Complexity Measurement:

The Cyclomatic complexity defines the number of independent paths in the basis set of the program that provides the upper bound for the number of tests that must be conducted to ensure that all the statements have been executed at least once.

There are three methods of computing Cyclomatic complexities.

Method 1: The Cyclomatic complexity, V (G) for a flow graph G can be defined as V

(G) = E - N + 2

Where: E is the total number of edges in the flow graph. N is the total number of nodes in the flow graph.

Method 2: The Cyclomatic complexity V (G) for a flow graph G can be defined as V

$$(G) = P + 1$$

Where: P is the total number of predicate nodes contained in the flow G.

Cyclomatic Complexity of Each Class:

Txttosign.php

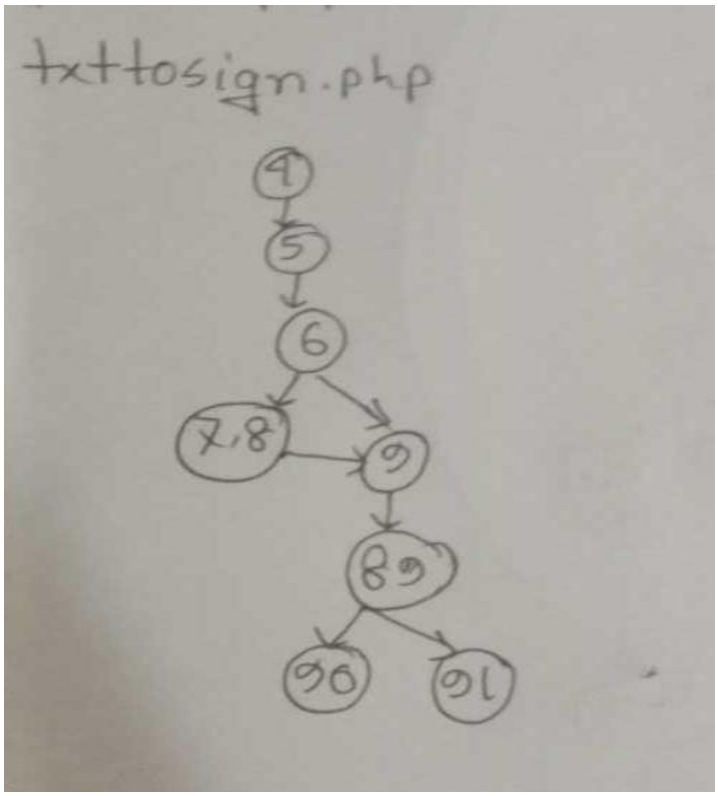

Figure 3: txttosign DD-Graph

Complexity:

Method 1: E - N+2

= 8 – 8+2

= 2

dynamic.php

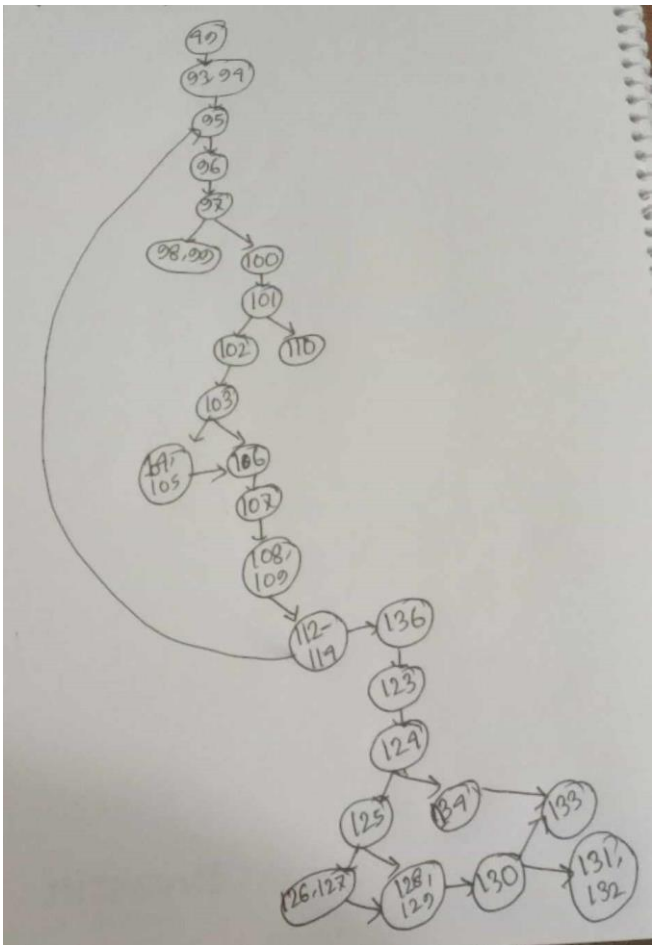

Figure 4: dynamic DD-Graph

Complexity:

Method 1: E - N+2

= 29 – 26+2

= 5

Login.php

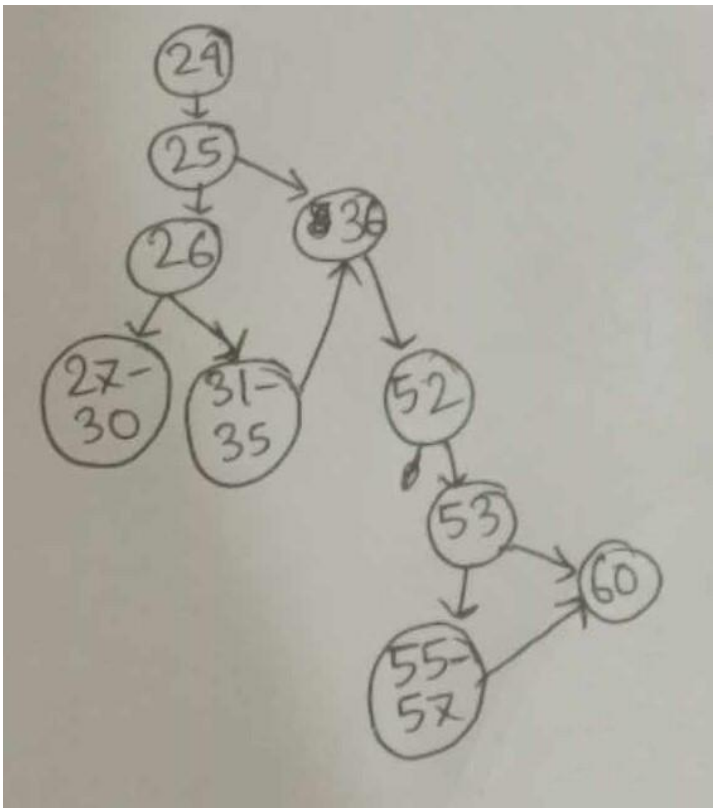

Figure 5: LogIn DD-Graph

Complexity:

Method 1: E - N+2

= 11 – 10+2

= 3

Education.php

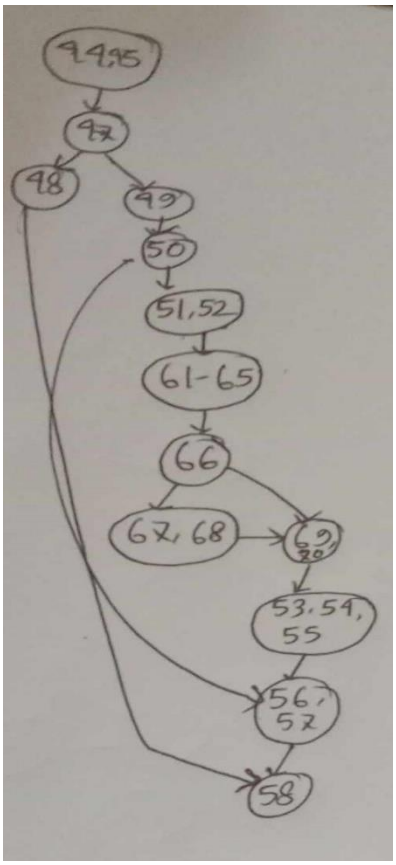

Figure 6: Education DD-Graph

Complexity:

Method 1: E - N+2

= 15 – 13+2

= 4

Sign.py

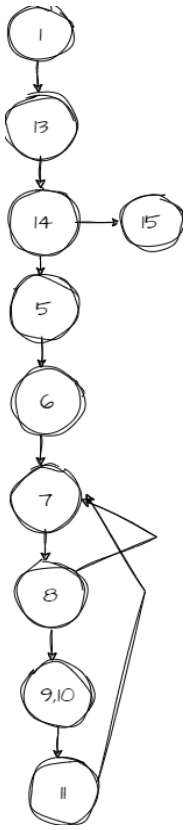

Figure 7: Sign DD-Graph

Complexity:

Method 1: E - N+2

= 11 – 10+2

= 3

Register.php

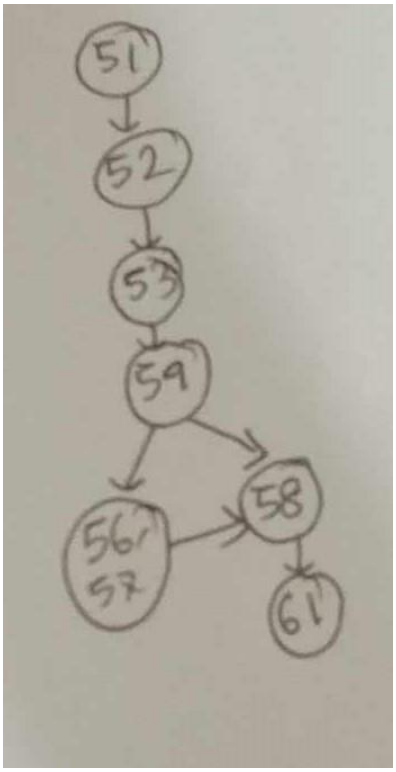

Figure 8: Register DD-Graph

Complexity:

Method 1: E - N+2

= 7 – 7+2

= 2

Index.php

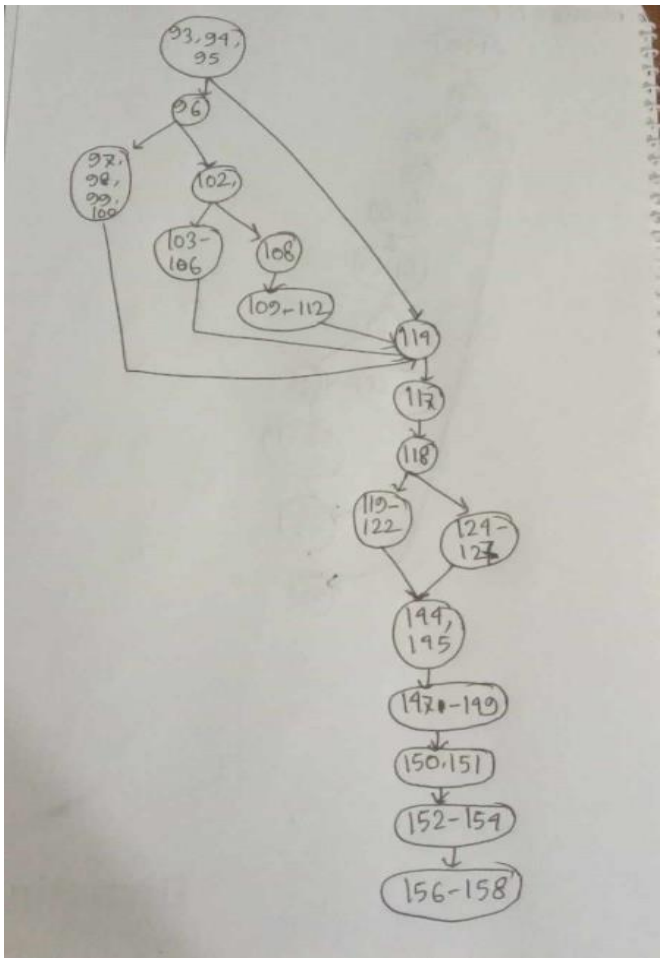

Figure 9: Index DD-Graph

Complexity:

Method 1: E - N+2

= 20 – 17+2

= 5

Process.php

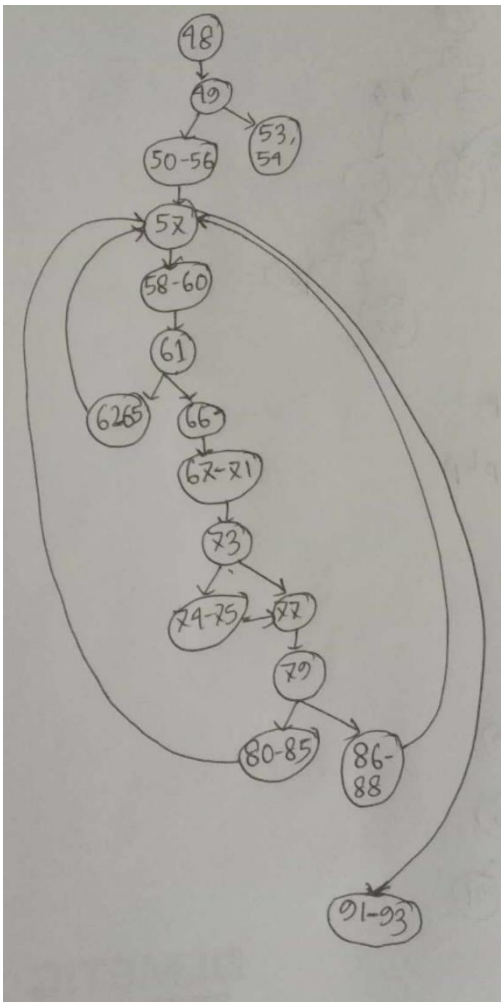

Figure 10: Process DD-Graph

Complexity:

Method 1: E - N+2

= 20 – 17+2

= 5