SOFTWARE METRICS

Prepared By:
Ikra Chowdhury Nowkshi (BFH1925019F)

Contents

TABLE 1

Software Metrics		Measuring Technique	Applicable Language
Properties Of Software Size	Code size	Manual Human Inspection	Java
	Halstead's Approach	Automated Program	Java
	Design Size	Manual Human Inspection	Java

Measurement

Code Size

The number of lines of code is the most popular metric for determining the size of a source code application.

TABLE 2

Measurement	Value
NLOC (Non commented line)	2674
CLOC (Comment lines of program text)	263
Blank Lines	693
Total size (LOC)	2937
density of comments (CLOC/LOC)	0.089

Halstead's Approach

Halstead's software science attempted to capture attributes of a program that paralleled physical and psychological measurements in other disciplines. He began by defining a program P as a collection of tokens, classified as either operators or operands.

$\mu 1$ = Number of unique operators

$\mu 2$ = Number of unique operands

N1 = Total occurrences of operators

N2 = Total occurrences of operands

The basic metrics for these tokens are the following:

TABLE 3

Measurement	Statistics
Unique Operator, $\mu 1$	15
Unique Operand, $\mu 2$	273
Total Operator, N1	1771
Total Operand, N2	1879

The length of P is defined to be N = N1 + N2, while the vocabulary of P is $\mu 1 = \mu 2 + \mu 3$. The volume of a program, akin to the number of mental comparisons needed to write a program of length N or the minimum number of bits to represent a program, is

$$N = N1 + N2$$
$$= 1771 + 1879$$
$$= 3650$$

While the vocabulary of P is:

$$\mu = \mu 1 + \mu 2$$
$$= 15 + 273$$
$$= 288$$

The volume of a program, akin to the number of mental comparisons needed to write a program of length N or the minimum number of bits to represent a program, that is

$$V = N \times \log_2 \mu$$
$$= 3650 \times 8.169$$
$$= 29{,}816$$

Design Size

We can gauge a design's size in a manner akin to how code size is evaluated. Instead of LOCs, we shall count design elements. The elements that we count rely on the design abstractions utilized and the design features that we are interested in. The level of abstraction, the developed objects, and the design technique all influence the right size measurement.

We will measure Design Size in term of

- Packages**:** Number of packages and subpackages, number of classes, interfaces (Java), or abstract classes.
    - Measure procedure : Manually.
    - Language : Java

- Design patterns:

1. Number of different design patterns used in a design.
2. Number of classes, interfaces, or abstract classes that play roles in each pattern realization.
    - Measure procedure : Manually.
    - Language : OOP

- Classes, interfaces, or abstract classes: Number of public methods or operations, number of attributes.
    - Measure procedure : Manually.
    - Language : Java, C++

- Methods or operations: Number of parameters, number of over loaded versions of a method or operation.
    - Measure procedure : Manually.
    - Language : All

- Weighted methods per class (WMC): Summing the weights of the methods in a class.
    - Measure procedure: Manually.
    - Language: OOP.

All Design measurement result for our Project "ChitChat(SPL1)" are shown in the table of term:

Packages

Type	Size
Packages	7
Classes	26
Abstract Classes	0
Interfaces	1

Design Pattern

Number of Design Pattern	**1**
Number of classes an interfaces used in the pattern	5

Methods

Public methods	147
Private methods	0
Overloaded methods	5
Parameters	132

Weighted methods per class (WMC):

WMC=5