Table 1: CYCLO Design Metric - Definition and Computation

Cyclomatic Complexity (CYCLO)	
Definition	Cyclomatic Complexity is the maximum number of linearly independent paths in a method. A path is linear if there is no branch in the execution flow of the corresponding code.
Worse	For greater values.
Computation Details	We compute the strict Cyclomatic Complexity: the Cyclomatic Complexity with logical conjunction and logical and in conditional expressions also adding 1 to the complexity for each of their occurrences. i.e., The statement if (a && b || c) would have a Cyclomatic Complexity of one but a strict Cyclomatic Complexity of three. The minimum Cyclomatic Complexity is one.
Visitor Type	AST Visitor

Implementation details for each entity the visitor can visit

visit: Method	**Visit Type**: AST Visitor
	Applicability: not Abstract Method, not belonging to Annotation nor Interface

Table 2: WMC Design Metric - Definition and Computation

Weighted Methods Count (WMC)	
Definition	WMC is the sum of complexity of the methods that are defined in the class. We compute the complexity with the Cyclomatic Complexity metric (CYCLO).
Worse	-
Visitor Type	AST Visitor

Implementation details for each entity the visitor can visit

visit: Class	**Visit Type**: Model Visitor
	Applicability: not Annotation nor Interface ComplexType

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Methods Declared In Class	Type	Type
CYCLO	Method	Type

Table 3: WMCNAMM Design Metric - Definition and Computation

Weighted Methods Count of Not Accessor or Mutator Methods (WMCNAMM)	
Definition	WMCNAMM is the sum of complexity of the methods that are defined in the class, and are not accessor or mutator methods. We compute the complexity with the Cyclomatic Complexity metric (CYCLO)
Worse	-
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Class	**Visit Type**: Model Visitor
	Applicability: not Annotation nor Interface ComplexType

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Methods Declared In Class	Type	Type

CYCLO	Method	Type

Table 4: AMW Design Metric - Definition and Computation

Average Methods Weight (AMW)	
Definition	The average static complexity of all methods in a class. We compute the complexity with the Cyclomatic Complexity metric (CYCLO).
Worse	-
Computation Details	$f(x) = \begin{cases} AMW = \frac{WMC}{NOM}, & NOM \neq 0 \\ AMW = 0, & NOM \neq 0 \end{cases}$
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Class	**Visit Type**: Model Visitor
	Applicability: not Annotation nor Interface ComplexType

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
WMC	Type	Type
NOM	Type	Type

Table 5: AMWNAMM Design Metric - Definition and Computation

Average Methods Weight of Not Accessor or Mutator Methods (AMWNAMM)	
Definition	The average static complexity of all methods in a class, which are not accessor or mutator. We compute the complexity with the Cyclomatic Complexity metric (CYCLO).
Worse	-
Computation Details	$f(x) = \begin{cases} \text{AMWNAMM} = \frac{\text{WMCNAMM}}{NOMNAMM}, & NOMNAMM \neq 0 \\ \text{AMWNAMM} = 0, & NOMNAMM \neq 0 \end{cases}$
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Class	**Visit Type**: Model Visitor
	Applicability: not Annotation nor Interface ComplexType

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
WMCNAMM	Type	Type
NOMNAMM	Type	Type

Table 6: MAXNESTING Design Metric - Definition and Computation

Maximum Nesting Level (MAXNESTING)	
Definition	The maximum nesting level of control structures within an operation.
Worse	For greater values.
Visitor Type	AST Visitor

Implementation details for each entity the visitor can visit

visit: Method	**Visit Type**: AST Visitor
	Applicability: not Abstract Method, not belonging to Annotation nor Interface

Table 7: WOC Design Metric - Definition and Computation

Weight of Class (WOC)	
Definition	The number of "functional" public methods divided by the total number of public members.
Worse	For lower values.
Computation Details	We compute this metrics as: $\frac{\text{Number of Non Abstract Public Non Accessor or Mutator Methods}}{\text{Total Number of Public Methods and Attribute}}$ If $Total\ Number\ of\ Public\ Methods\ and\ Attribute$ is zero, WOC is zero.
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Class	**Visit Type**: Model Visitor
	Applicability: Not Abstract ComplexType that are not Interface nor Annotation

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Methods Declared In Class	Type	Type
Attributes of Class	Type	Type

Table 8: CLNAMM Design Metric - Definition and Computation

Called Local Not Accessor or Mutator Methods (CLNAMM)	
Definition	The number of lines of code of an operation or of a class, including blank lines and comments.
Worse	-
Computation Details	We sum up the number of not accessor or mutator $Called\ Intra\ Methods$.
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Method	**Visit Type**: Model Visitor
	Applicability: not Abstract Method, not from Interface and Annotation ComplexType.

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Called Intra Methods	Method	Method

Table 9: NOP Design Metric - Definition and Computation

Number of Parameters (NOP)	
Definition	Number of parameters of a method.
Worse	For greater values.
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Method	**Visit Type**: Model Visitor
	Applicability: Method

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Methods Declared In Class	Type	Type

Table 10: NOAV Design Metric - Definition and Computation

Called Local Not Accessor or Mutator Methods (CLNAMM)	
Definition	The total number of variables accessed directly or through accessor methods from the measured operation. Variables include parameters, local variables, but also instance variables and global variables declared in classes belonging to the system
Worse	For greater values.
Computation Details	We count the $Used\ Variables$ defined within the system and not in external libraries. The context we consider are: $MethodDeclarationParameter$, $CatchClause$, $EnumConstantDeclaration$, $VariableDeclarationExpression$, $VariableDeclarationStatement$, $SingleVariableDeclaration$, $VariableDeclarationFragment$. To count the variable accessed through accessor methods we get the list of $Called\ Methods$ and we count the $Used\ Intra\ Variables$ by each accessor method in the set of $Called\ Methods$. We count also the variable access through static methods.
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Method	**Visit Type**: Model Visitor
	Applicability: not Abstract Method, not from Interface and Annotation ComplexType.

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Used Variables	Method	Method
Used Intra Variables	Method	Project
Called Methods	Method	Method

Table 11: ATLD Design Metric - Definition and Computation

Access to Local Data (ATLD)	
Definition	The number of attributes declared by the current classes accessed by the measured method directly or by invoking accessor methods.
Worse	For greater values.
Computation Details	We count the $Used\ Intra\ Variables$ defined within the system and not in external libraries. To count the variable accessed through accessor methods we get the list of $Called\ Intra\ Methods$ and we count the $Used\ Intra\ Variables$ by each accessor method in the set of $Called\ Methods$.
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Method	**Visit Type**: Model Visitor
	Applicability: not Abstract Method, not from Interface and Annotation ComplexType.

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Used Intra Variables	Method	Project

Used Hierarchy Variables	Method	Project
Called Intra Methods	Method	Method

Table 12: NOLV Design Metric - Definition and Computation

Access to Local Data (ATLD)	
Definition	Number of local variable declared in a method. The method's parameter are considered local variable.
Worse	For greater values.
Computation Details	We count the number of **Used Intra Variable** in contexts that represent variable declaration (e.g., $MethodDeclarationParameter$, $CatchClause$, $EnumConstantDeclaration$, $VariableDeclarationExpression$, $VariableDeclarationStatement$, $SingleVariableDeclaration$, $VariableDeclarationFragment$)
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Method	**Visit Type**: Model Visitor
	Applicability: not Abstract Method, not from Interface and Annotation ComplexType.

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Used Intra Variables	Method	Method

Table 13: TCC Design Metric - Definition and Computation

Tight Class Cohesion (TCC)	
Definition	TCC is the normalized ratio between the number of methods directly connected with other methods through an instance variable and the total number of possible connections between methods. A direct connection between two methods exists if both access the same instance variable directly or indirectly through a method call. TCC takes its value in the range [0,1].
Worse	For lower values.
Computation Details	**Given:** Maximum number of possible connections: Where N is the number of visible methods. $NP = \frac{N * (N-1)}{2}$ Number of direct connections: NDC, computed using a connectivity matrix that records all direct connected methods, making attention to cyclic calls among methods. We compute: $\begin{cases} TCC = \frac{NDC}{NP} & NP \neq 0 \\ TCC = 1 & NP = 0 \end{cases}$ For TCC only visible methods are considered, i.e., they are not private, implement an interface, or handle an event. Constructors are ignored. Constructors are a problem, because of indirect connections with attributes. They create indirect connections between methods which use different attributes, and increase cohesion, which is not real.

Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Class	**Visit Type**: Model Visitor
	Applicability: Not Abstract ComplexType that are not Interface nor Annotation.

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Methods Declared In Class	Type	Type
Used Hierarchy Variables	Type	Type
Called Intra Methods	Type	Type
Used Intra Variables	Type	Type

Table 14: LCOM5 Design Metric - Definition and Computation

Tight Class Cohesion (TCC)	
Definition	$$LCOM5 = \frac{NOM - \frac{\sum_{m \in M} NOAcc(m)}{NOA}}{NOM - 1}$$ where M is the set of methods of the class, NOM the number of methods, NOA the number of attributes, and $NOAcc(m)$ is the number of attributes of the class accessed by method m.
Worse	For lower values.
Computation Details	For $\sum_{m \in M} NOAcc(m)$ we sum up $Used\ Intra\ Variables$ by not constructor methods of the measured class. Then we compute: $LCOM5 = \frac{NOM - \frac{\sum_{m \in M} NOAcc(m)}{NOA}}{NOM - 1}$ $\quad \text{NOM} > 1 \wedge \text{NOA} > 0$ $LCOM5 = 0$ $\quad \text{NOM} \leq 1 \vee \text{NOA} \leq 0$
Visitor Type	Model Visitor

Implementation details for each entity the visitor can visit

visit: Class	**Visit Type**: Model Visitor
	Applicability: not Interface nor Annotation ComplexType

Dependencies Information:

Dep-visitor	**Dep-entity**	**Dep-level**
Methods Declared In Class	Type	Type
Used Hierarchy Variables	Type	Type
NOM	Type	Type
NOA	Type	Type