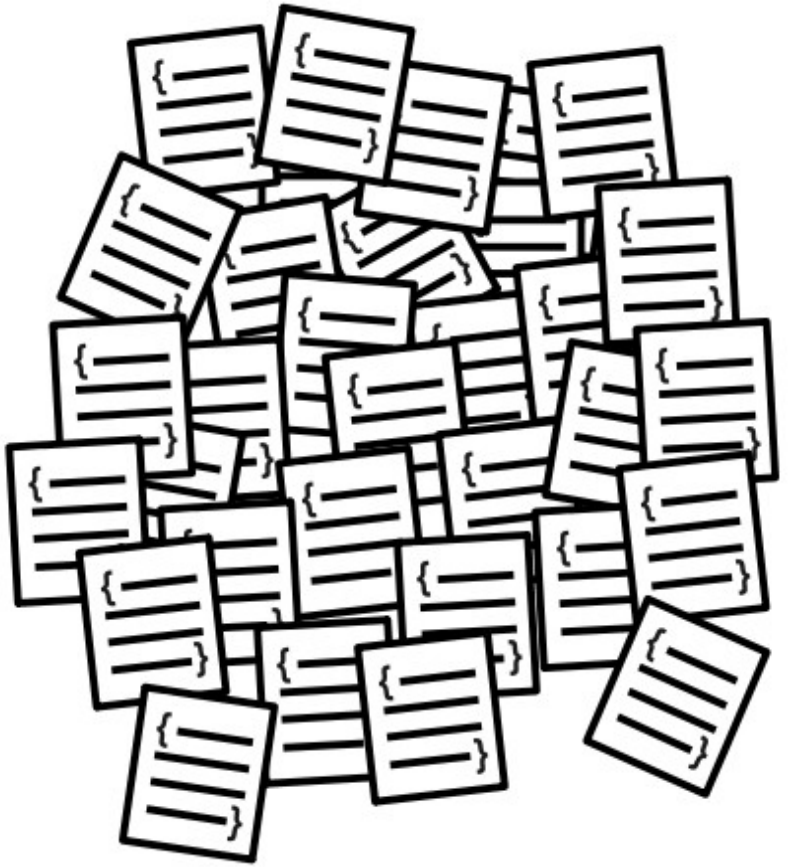
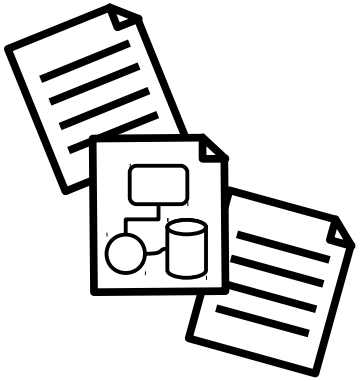
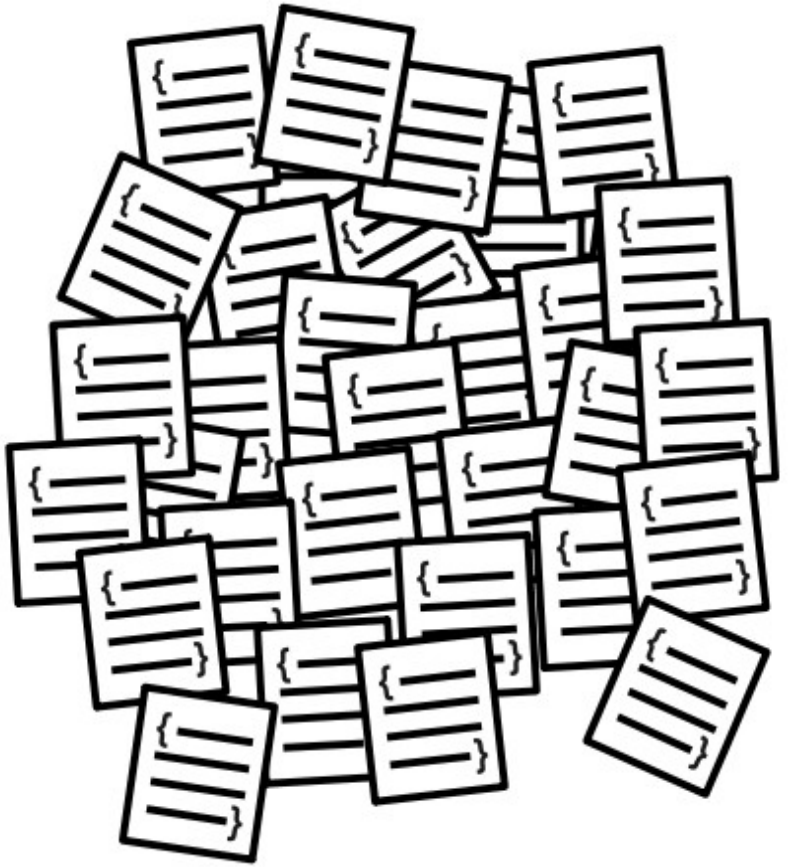
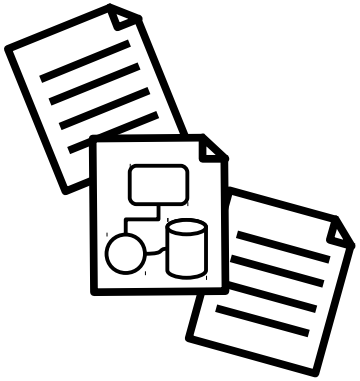
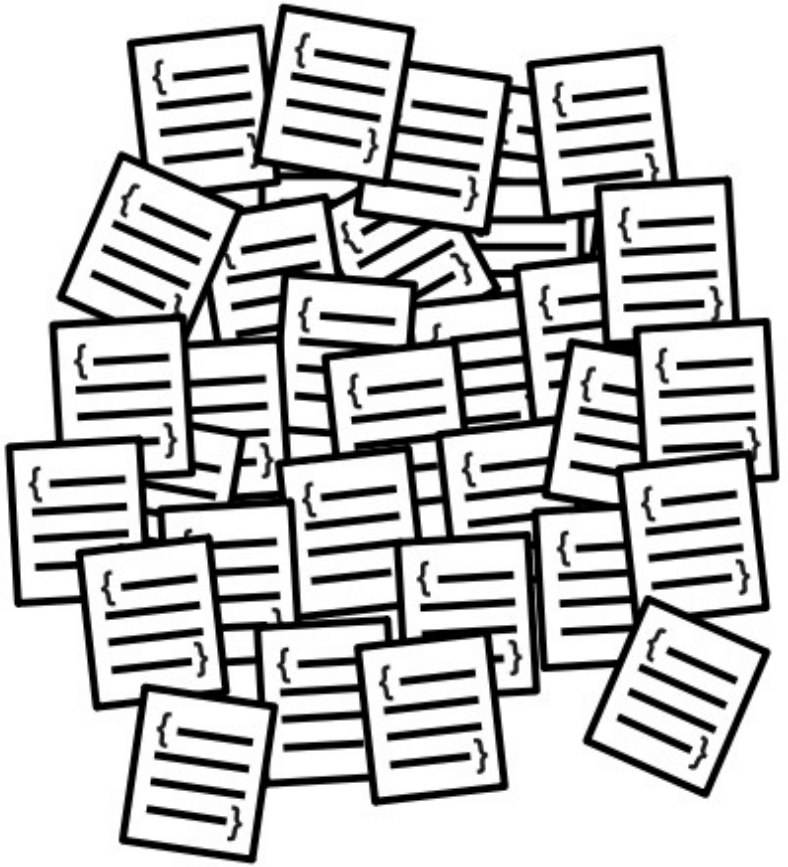
Métricas



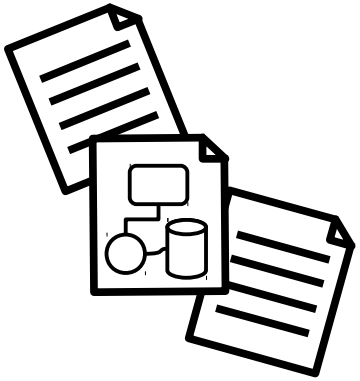


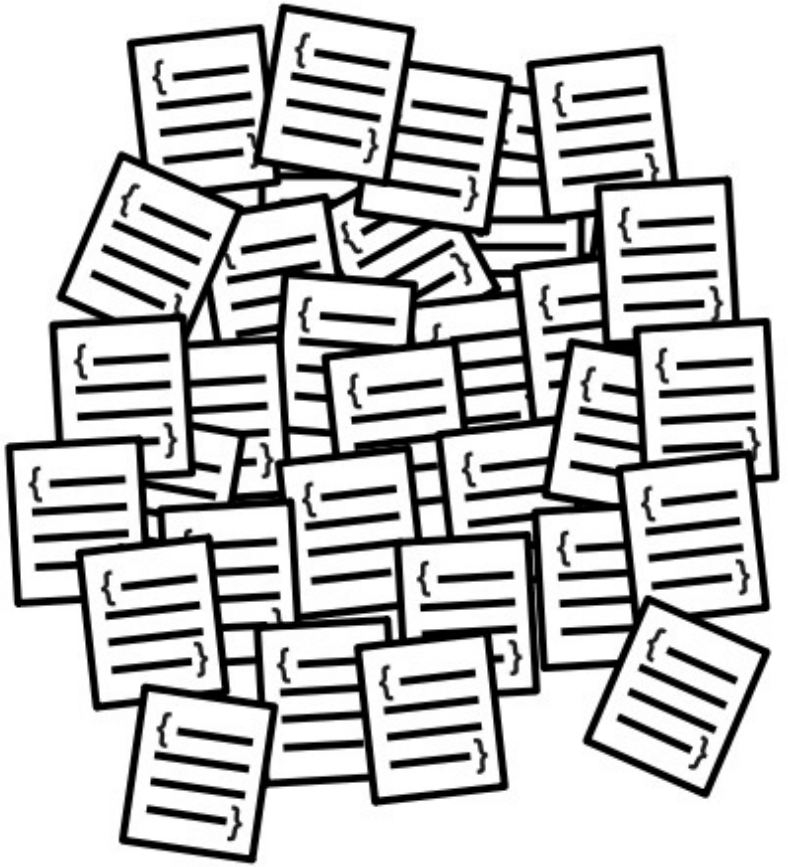


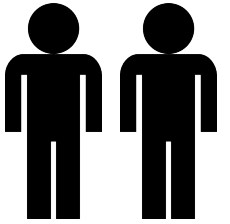




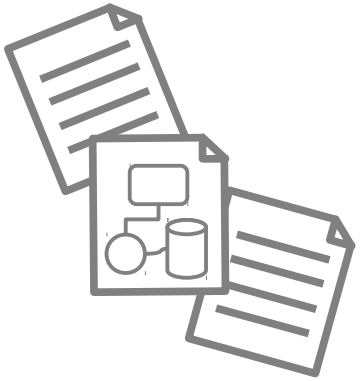

tests

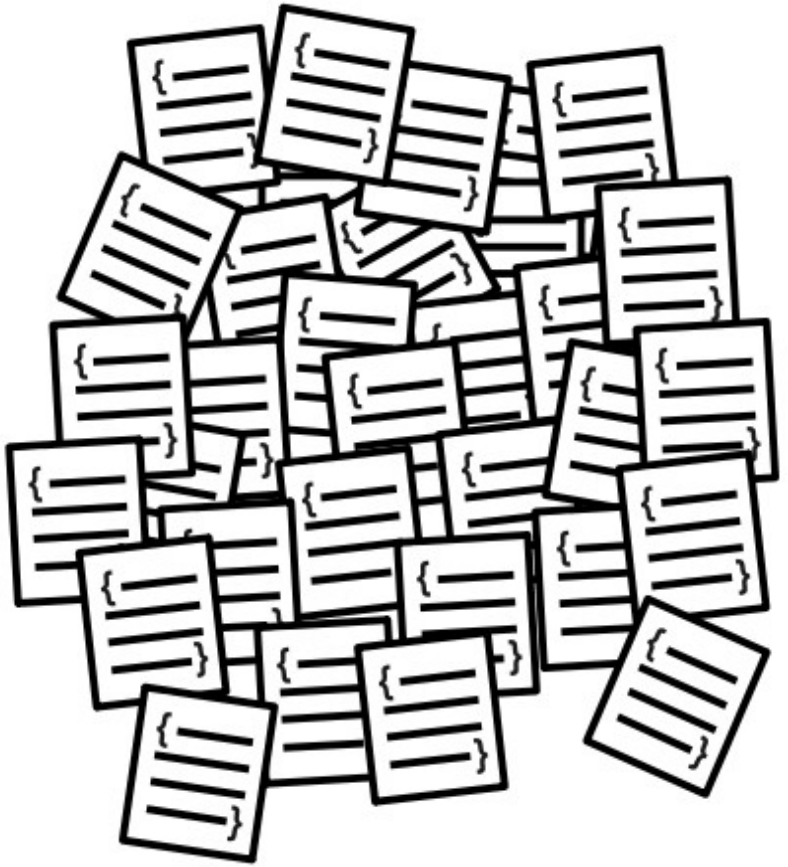


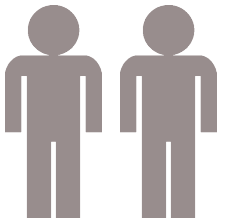




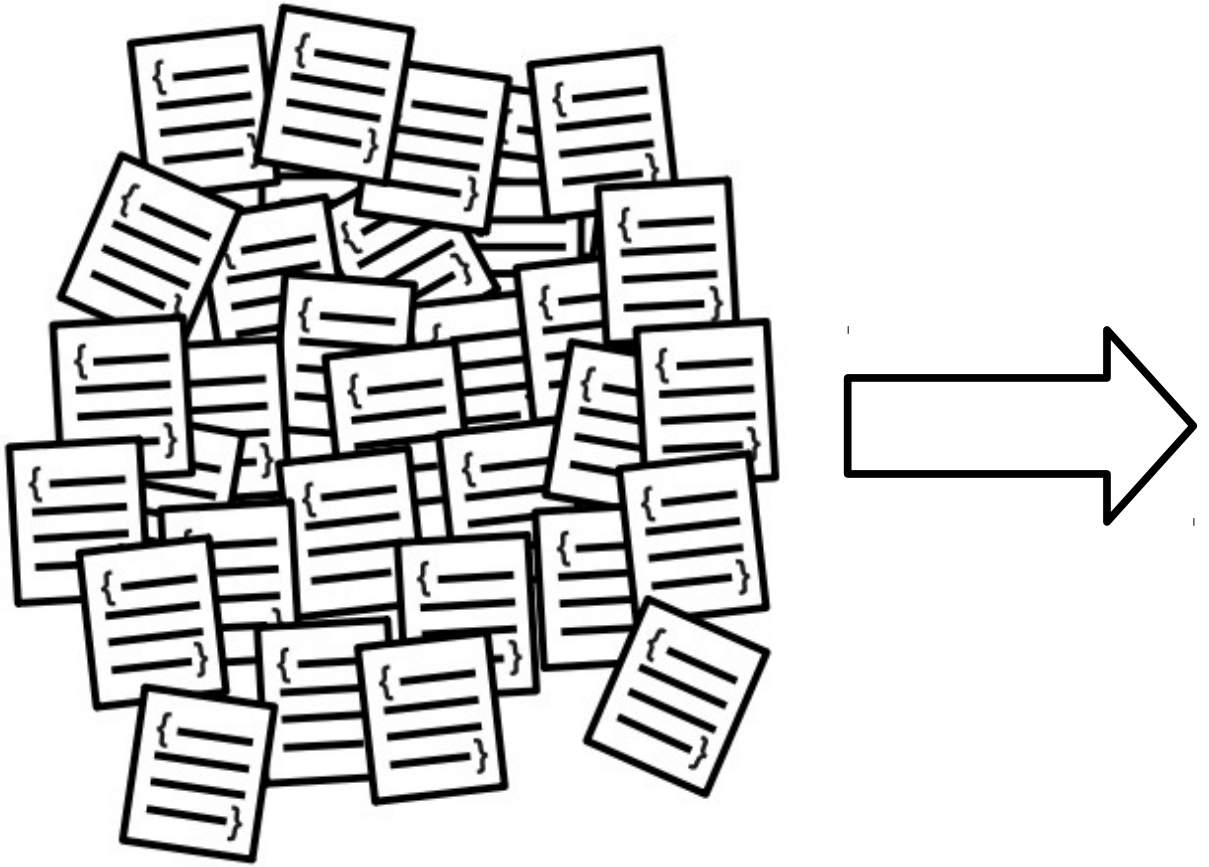

tests

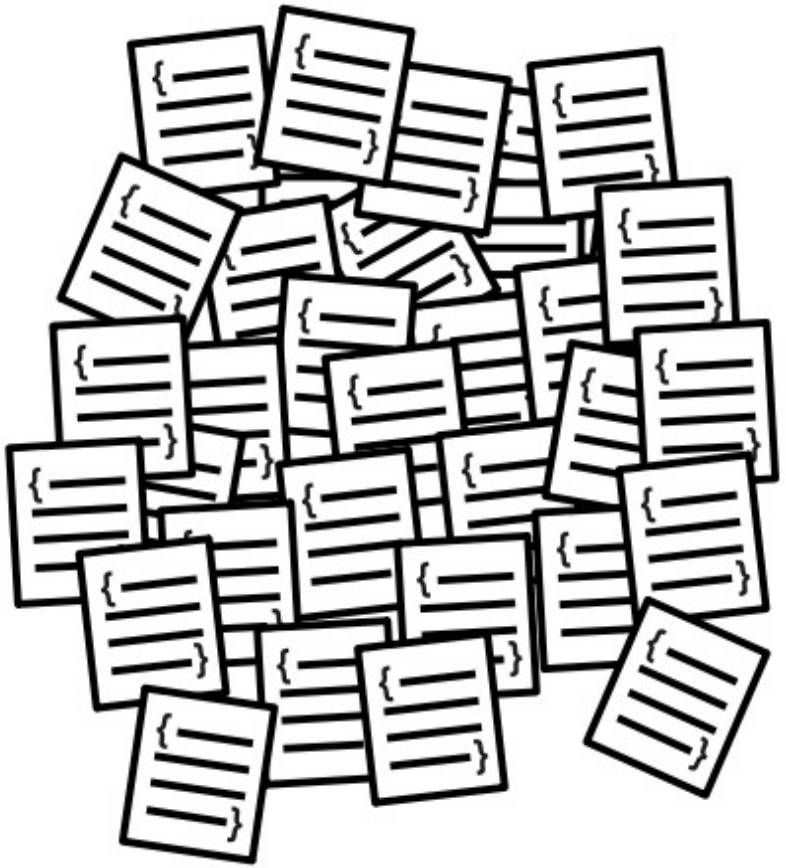


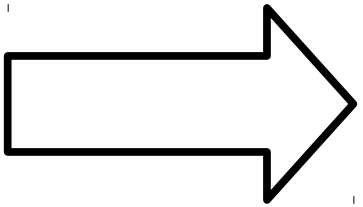





tests







100233 LOC

3563 NOC

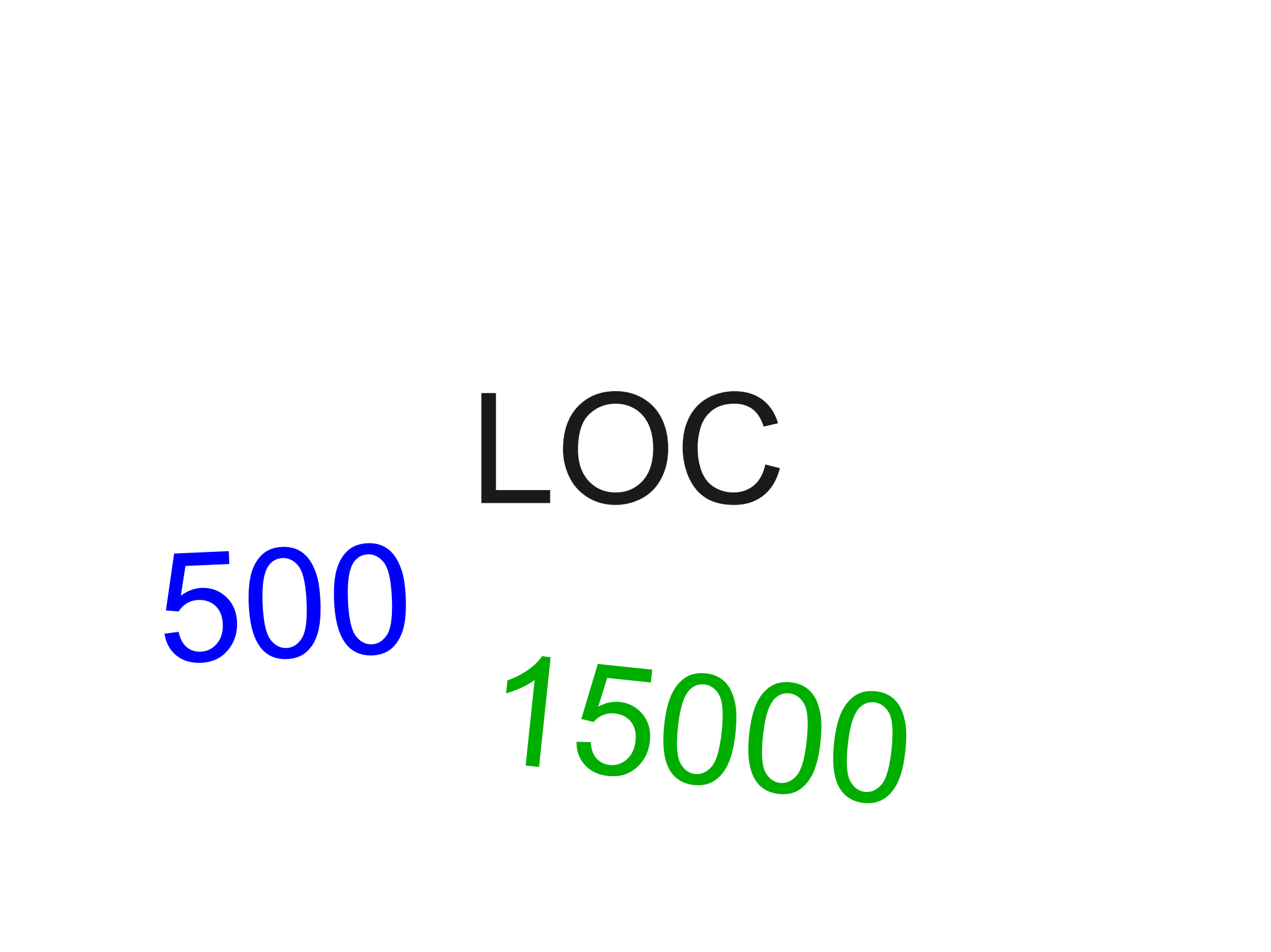
1072 NOP

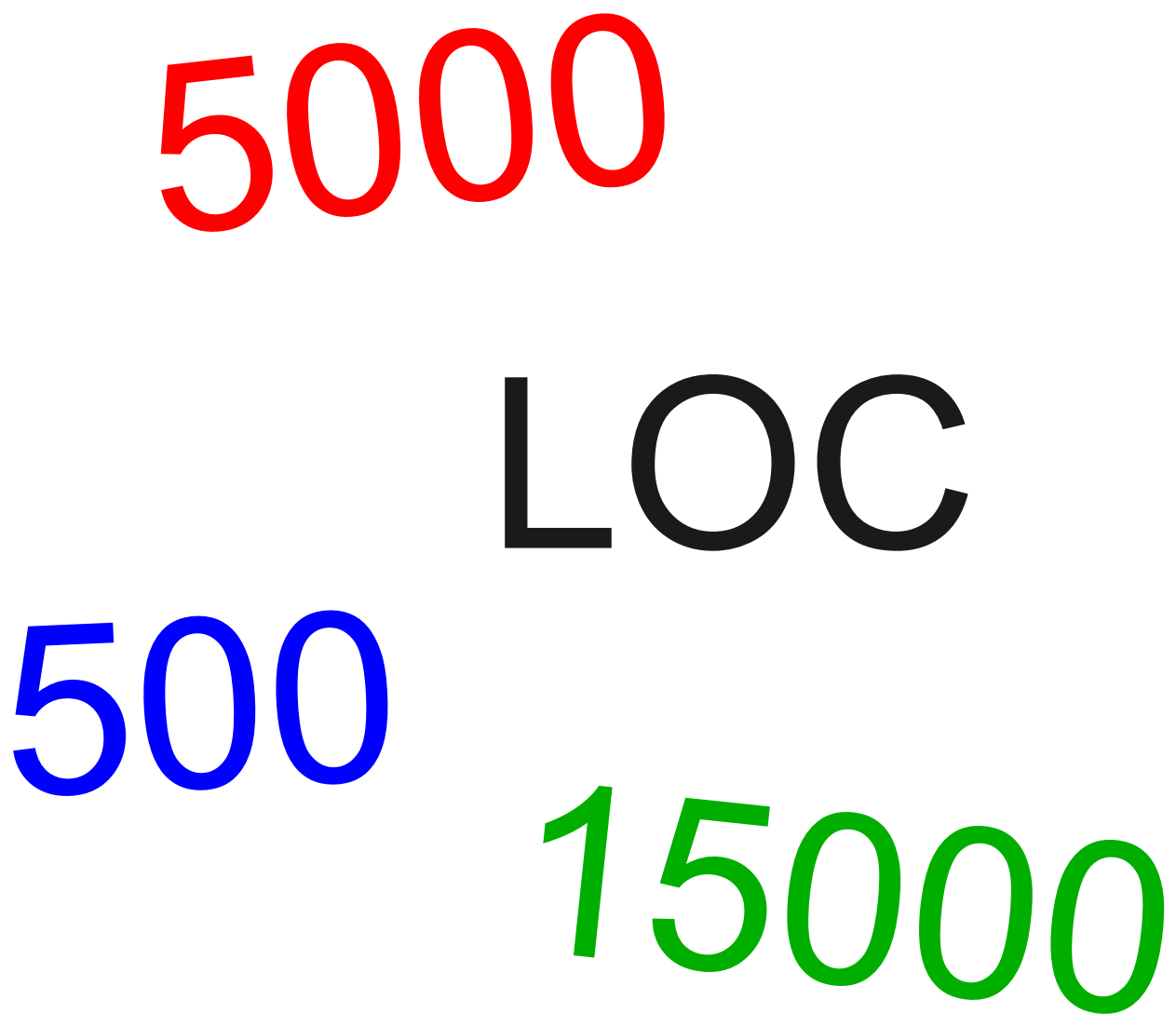
...

LOC

LOC

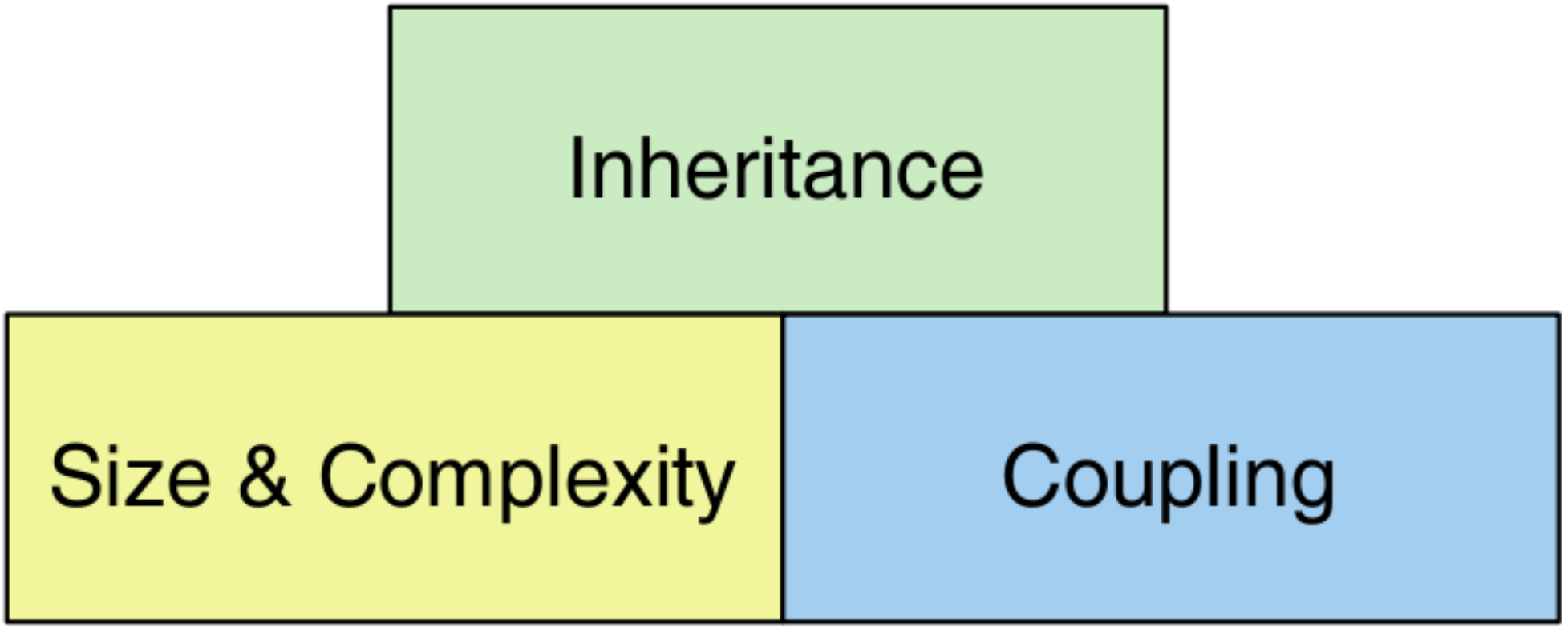
500

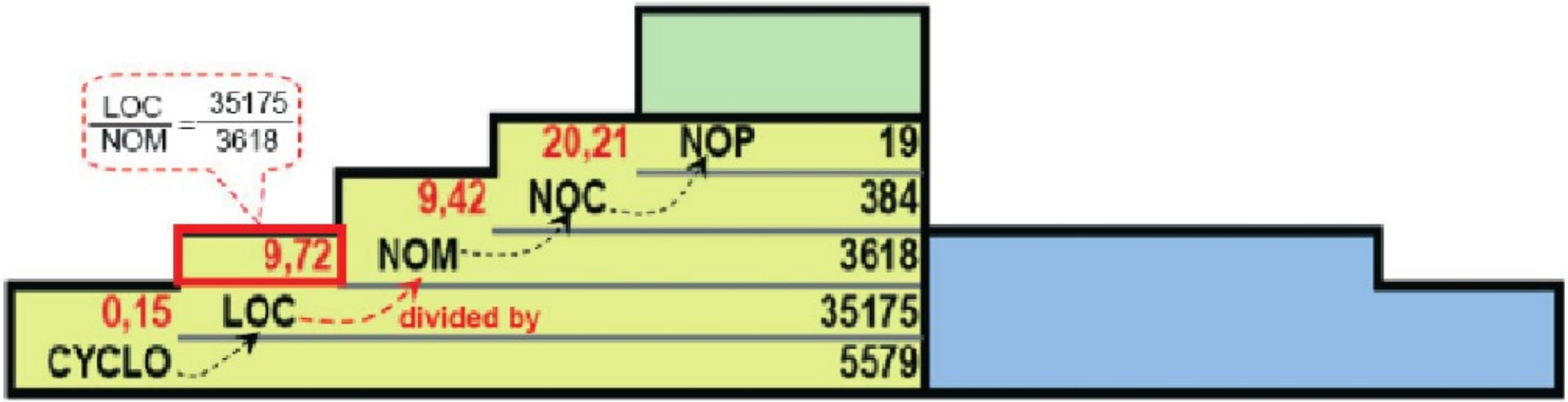

LOC
500
15000

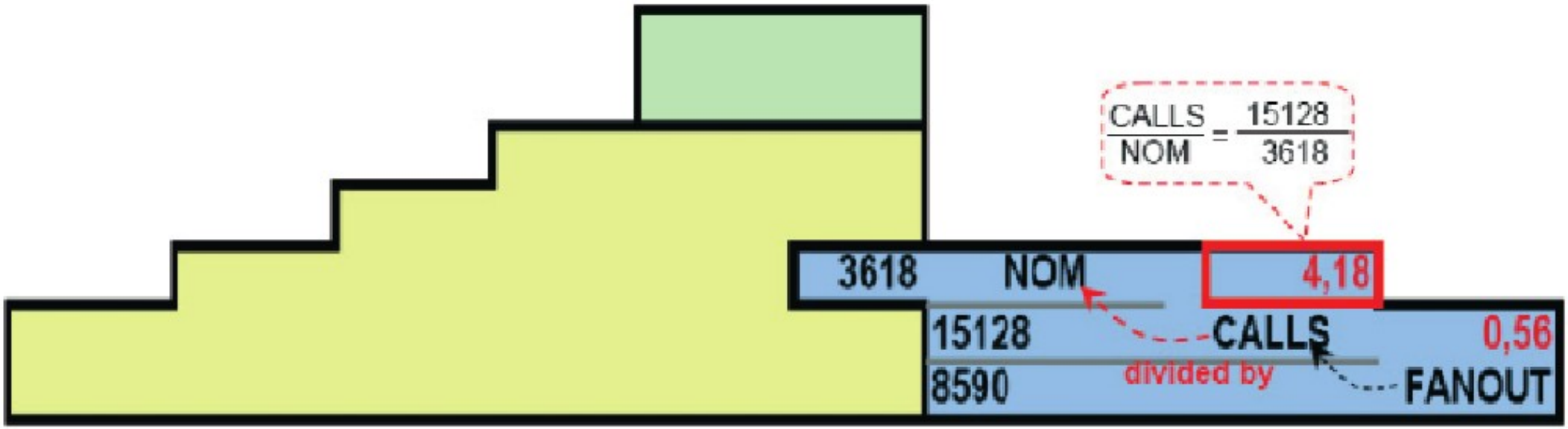

5000
LOC
500
15000

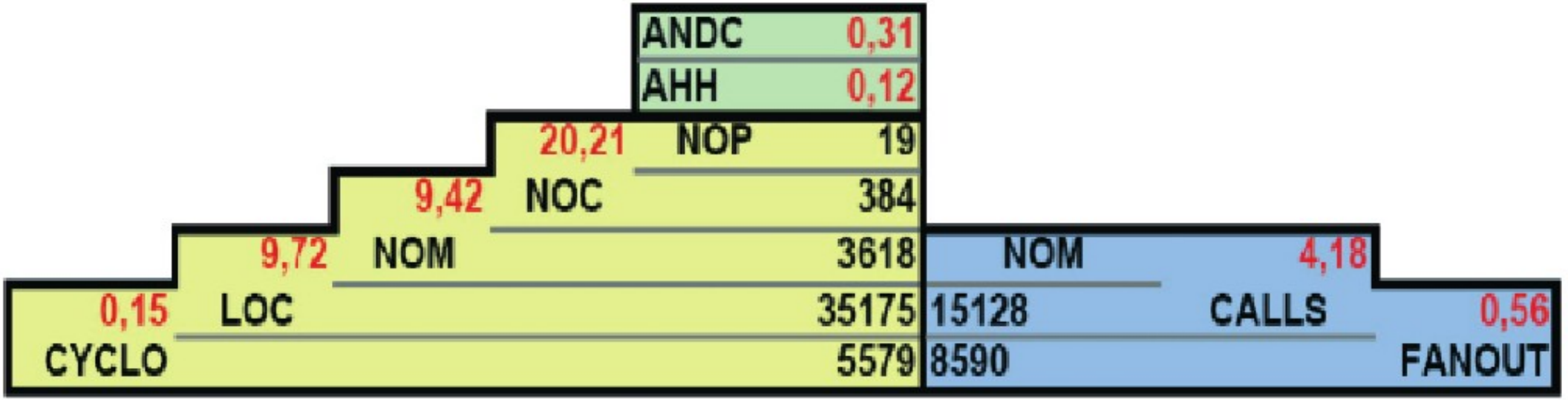
Caracterización de Sistemas

Overview Pyramid

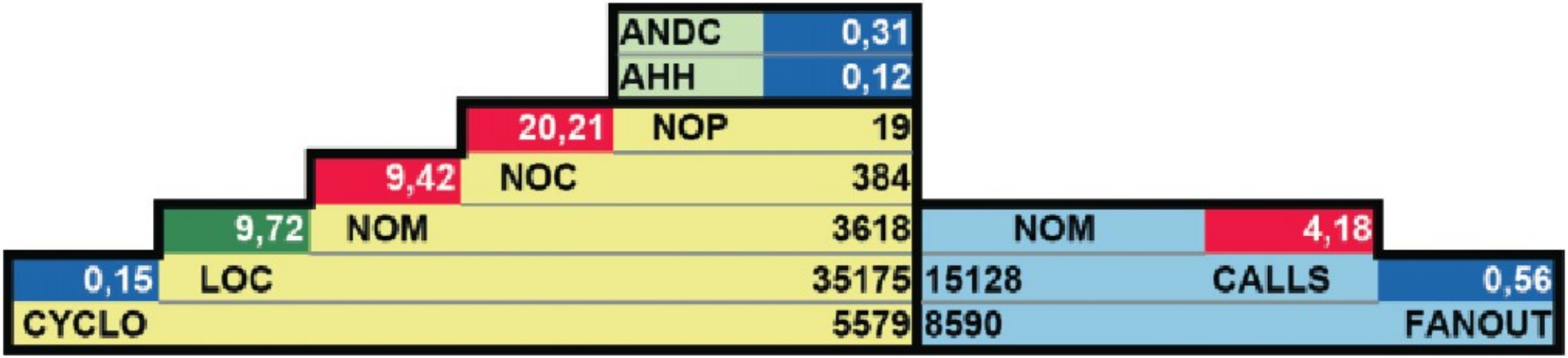

Inheritance
Size & Complexity
Coupling


LOC / NOM = 35175 / 3618
20,21
NOP
19
9,42
NOC
384
9,72
NOM
3618
0,15
LOC
divided by
35175
CYCLO
5579

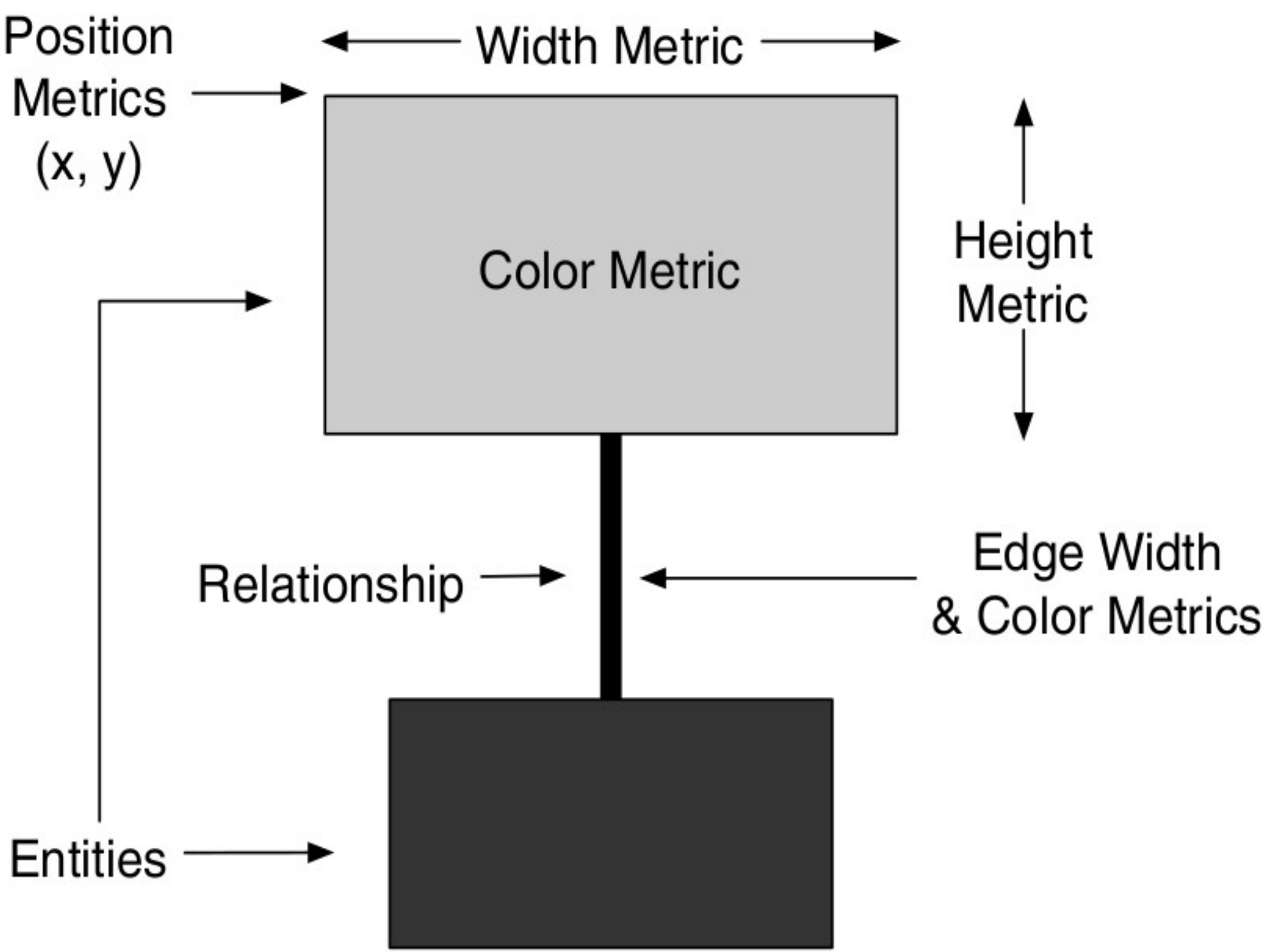

CALLS / NOM = 15128 / 3618
3618
NOM
4,18
15128
CALLS
0,56
divided by
8590
FANOUT

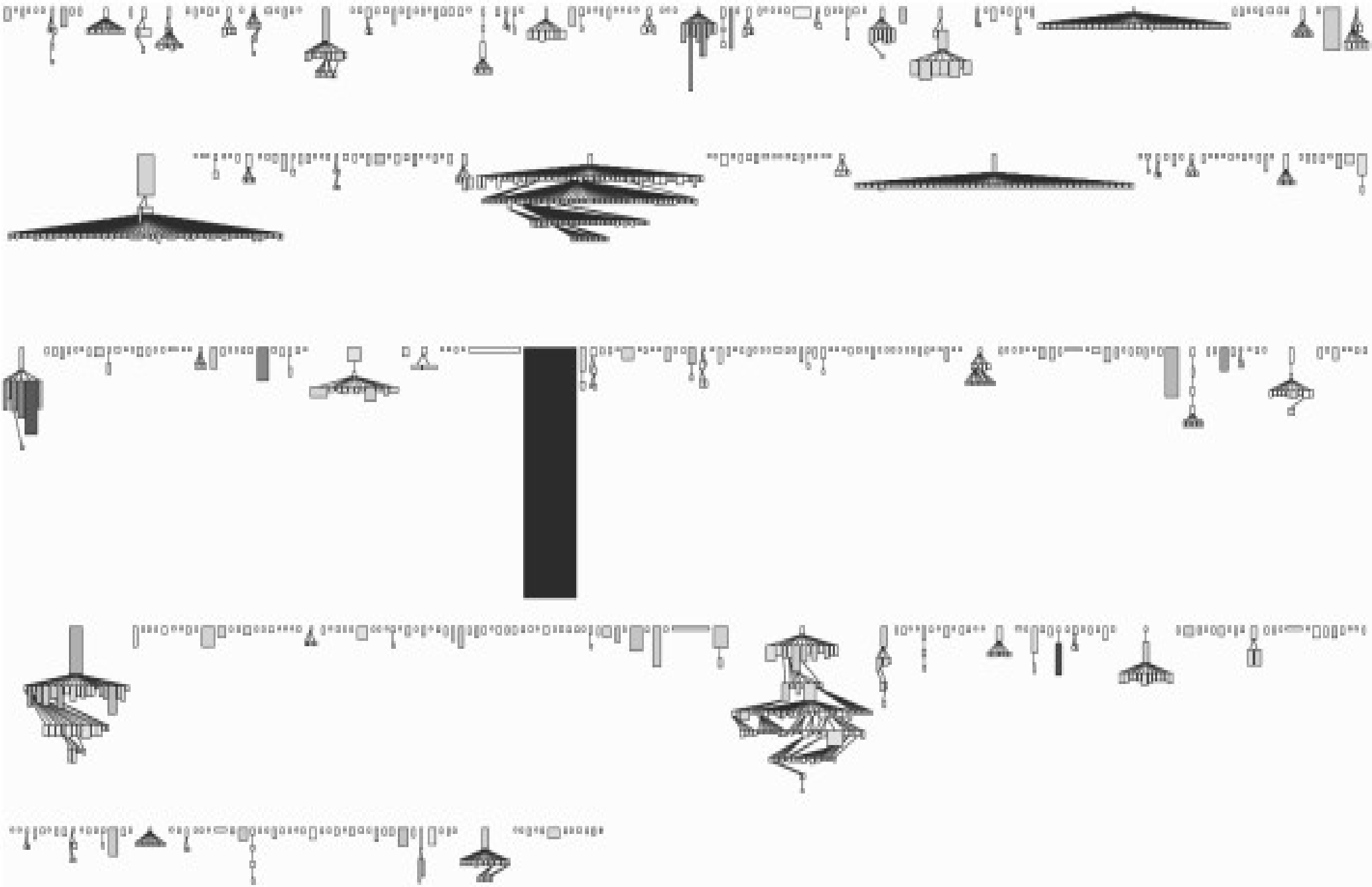

ANDC 0,31
AHH 0,12
20,21 NOP 19
9,42 NOC 384
9,72 NOM 3618
NOM 4,18
0,15 LOC 35175
15128 CALLS 0,56
CYCLO 5579
8590 FANOUT

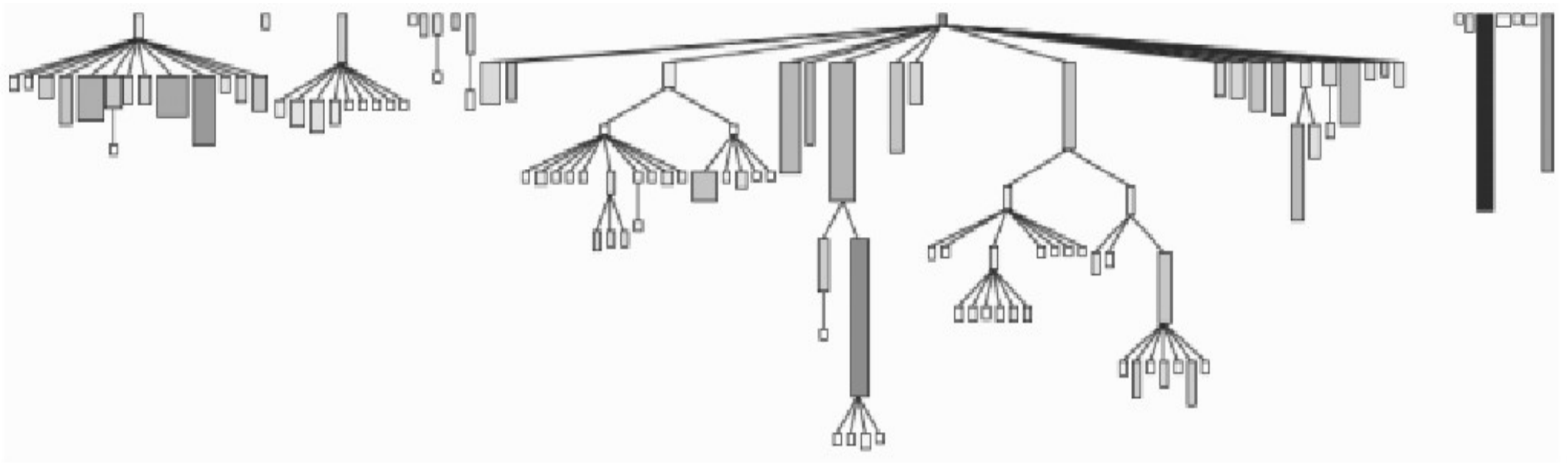
	Java			C++		
Metric	Low	Average	High	Low	Average	High
CYCLO/Line of code	0.16	0.20	0.24	0.20	0.25	0.30
LOC/Operation	7	10	13	5	10	16
NOM/Class	4	7	10	4	9	15
NOC /Package	6	17	26	3	19	35
CALLS/Operation	2.01	2.62	3.2	1.17	1.58	2
FANOUT /Call	0.56	0.62	0.68	0.20	0.34	0.48
ANDC	0.25	0.41	0.57	0.19	0.28	0.37
AHH	0.09	0.21	0.32	0.05	0.13	0.21

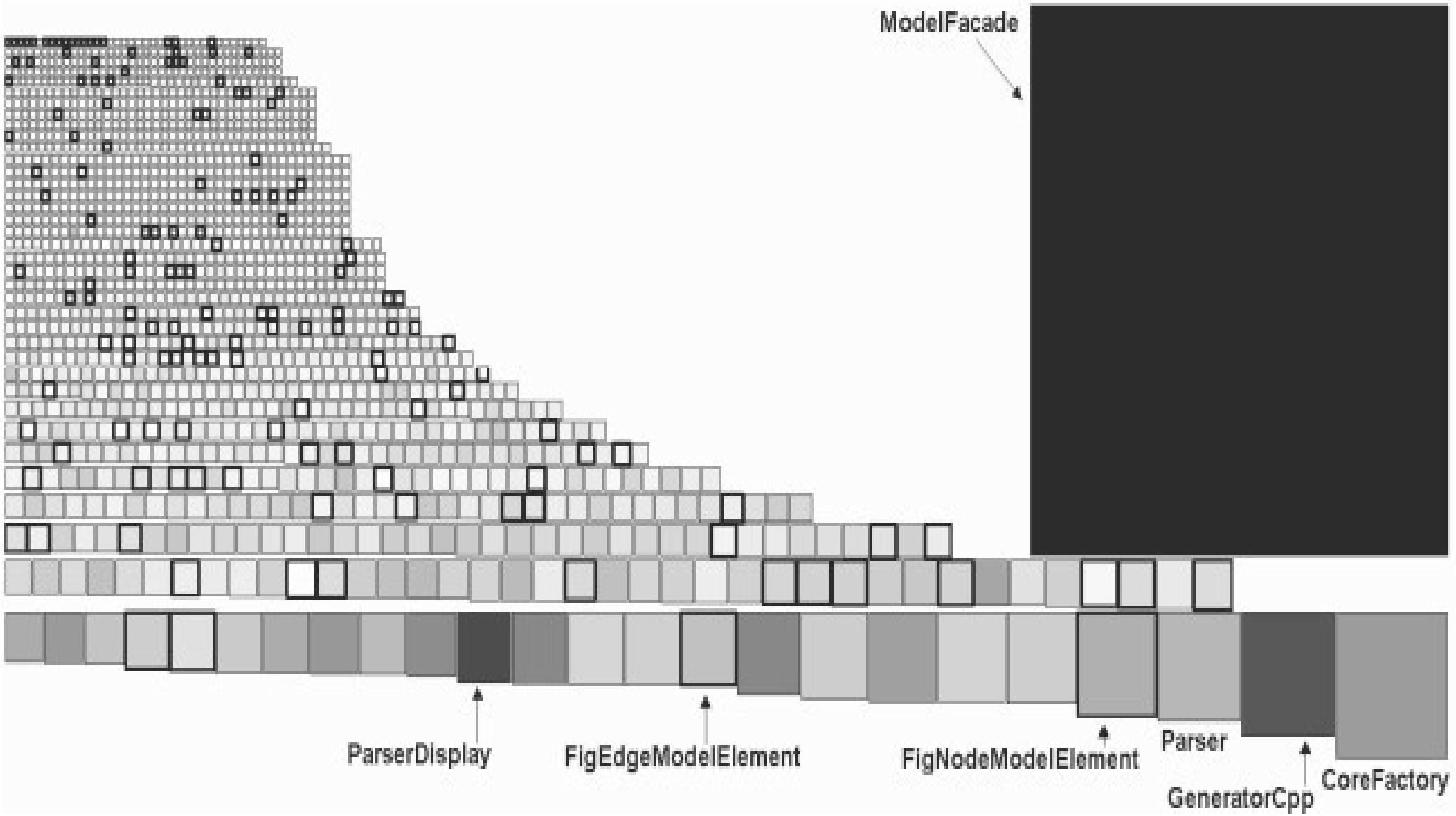

ANDC 0,31
AHH 0,12
20,21 NOP 19
9,42 NOC 384
9,72 NOM 3618
0,15 LOC 35175
CYCLO 5579
NOM 4,18
15128 CALLS 0,56
8590 FANOUT

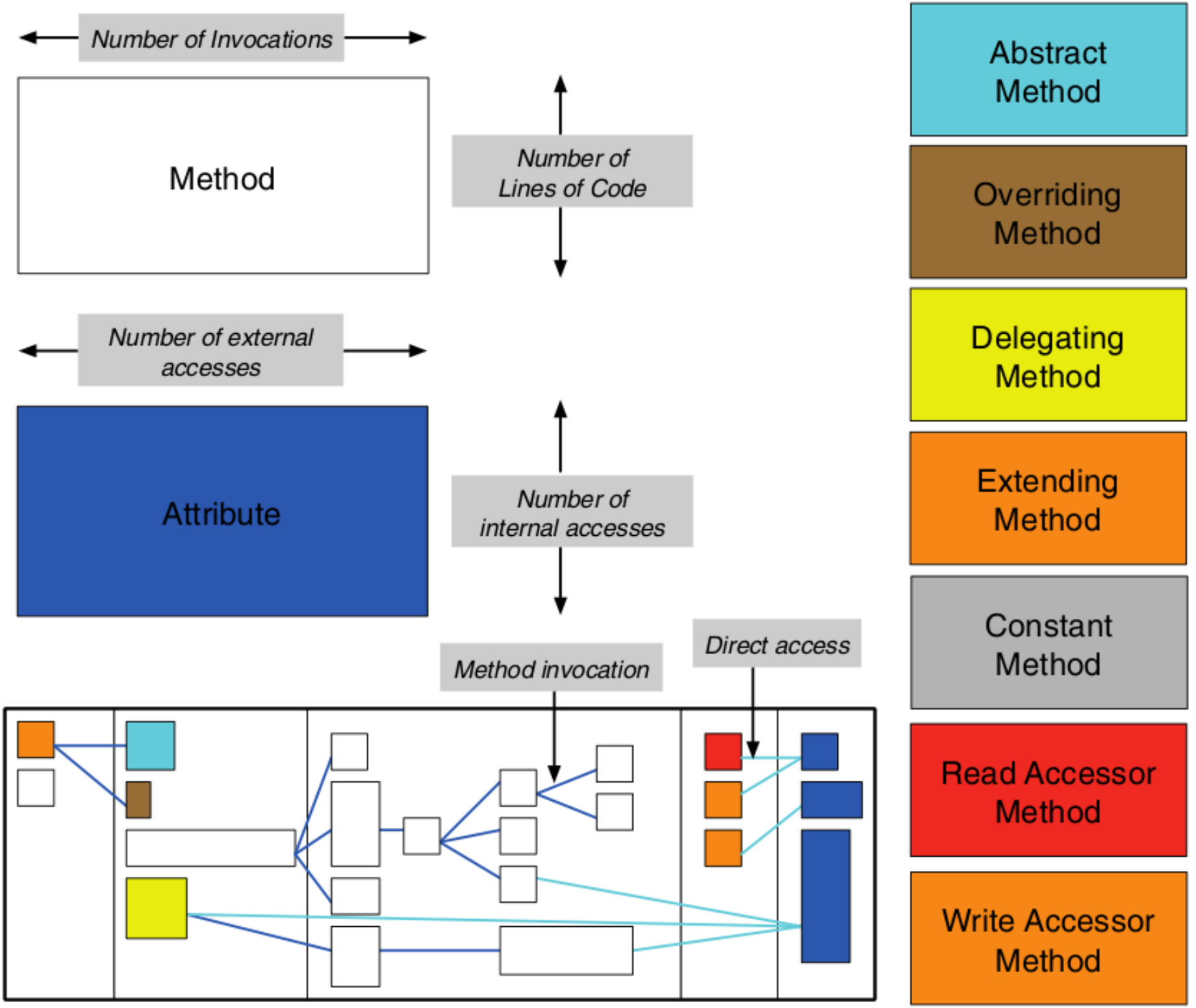
Polimétricas

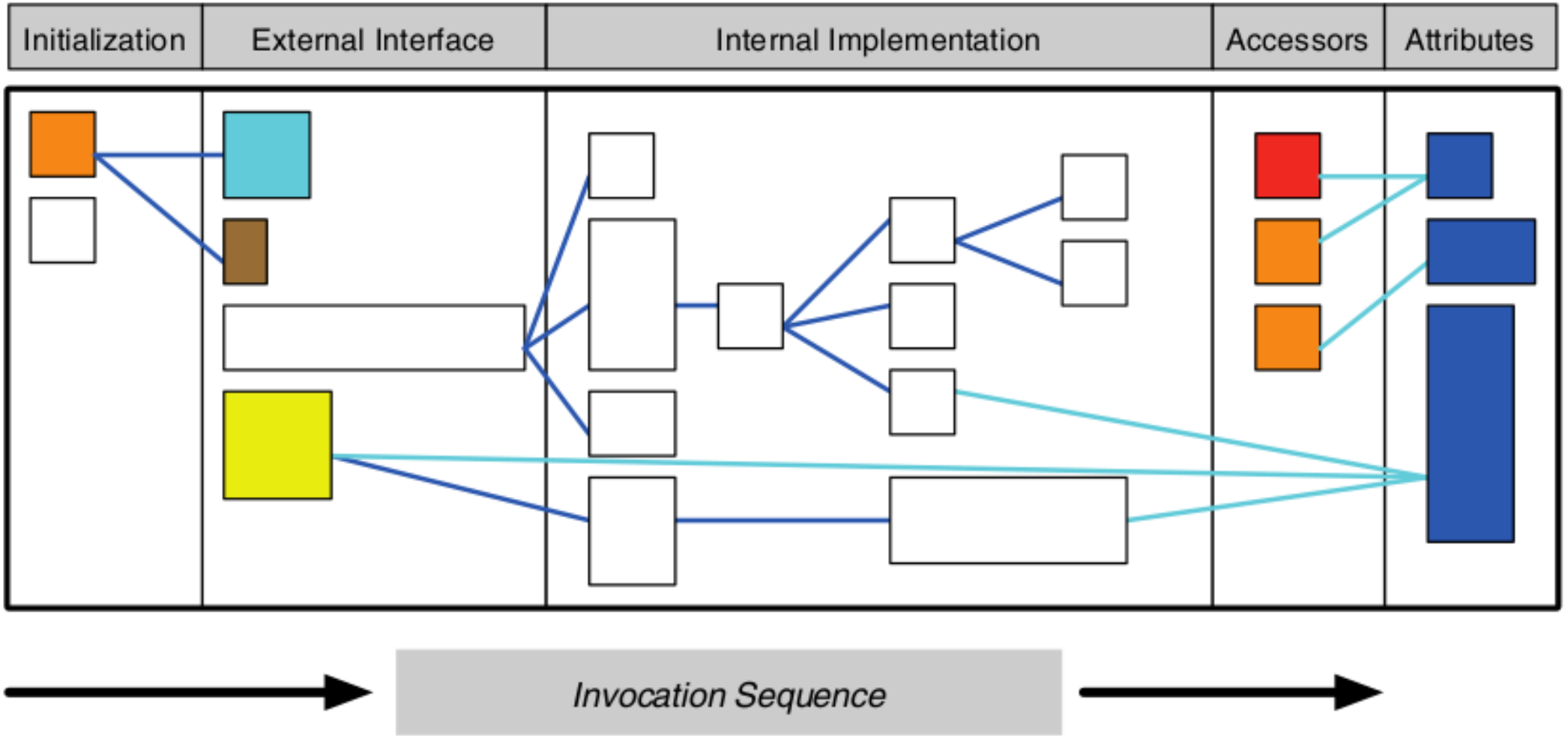

Position
Metrics
(x, y)
Width Metric
Color Metric
Height
Metric
Relationship
Edge Width
& Color Metrics
Entities

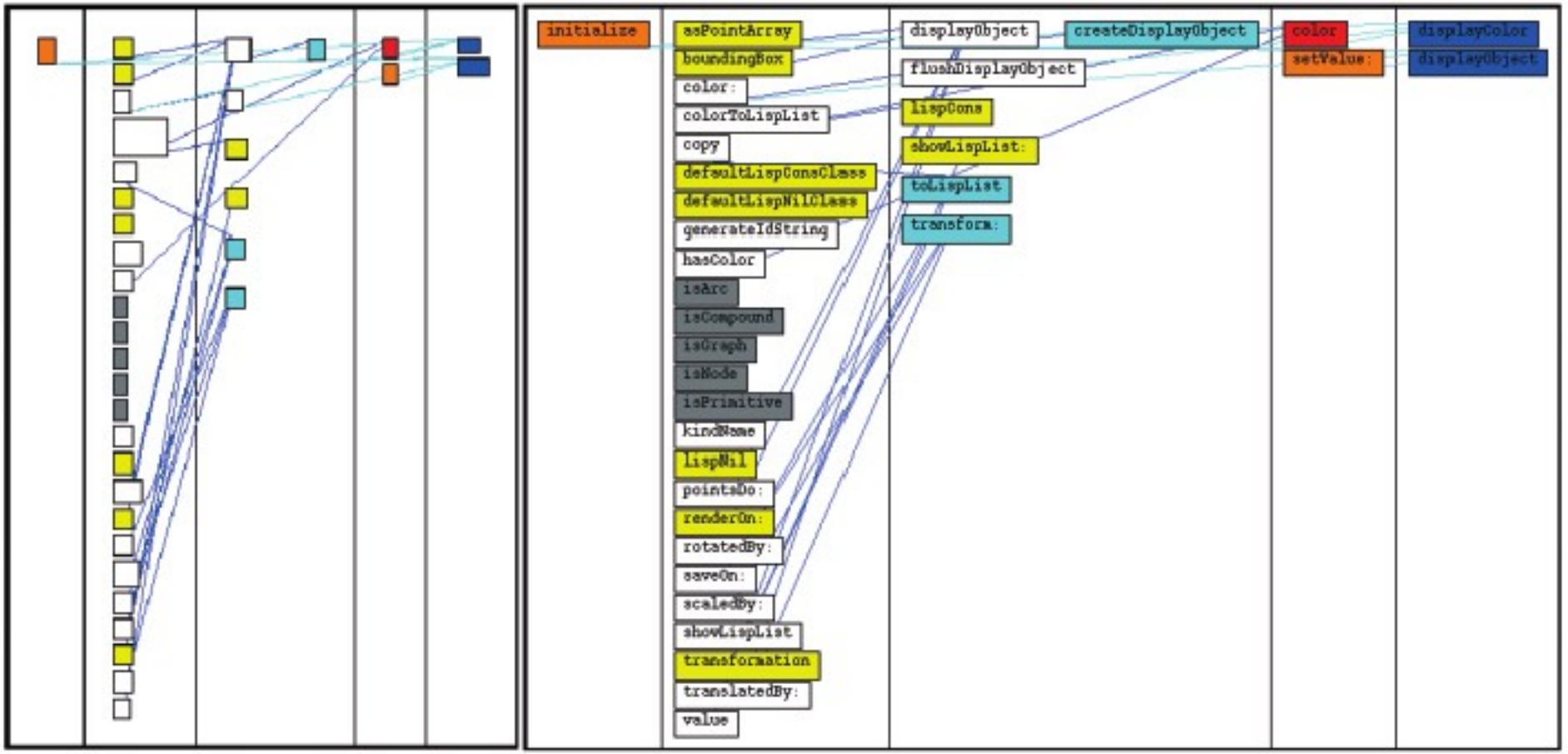


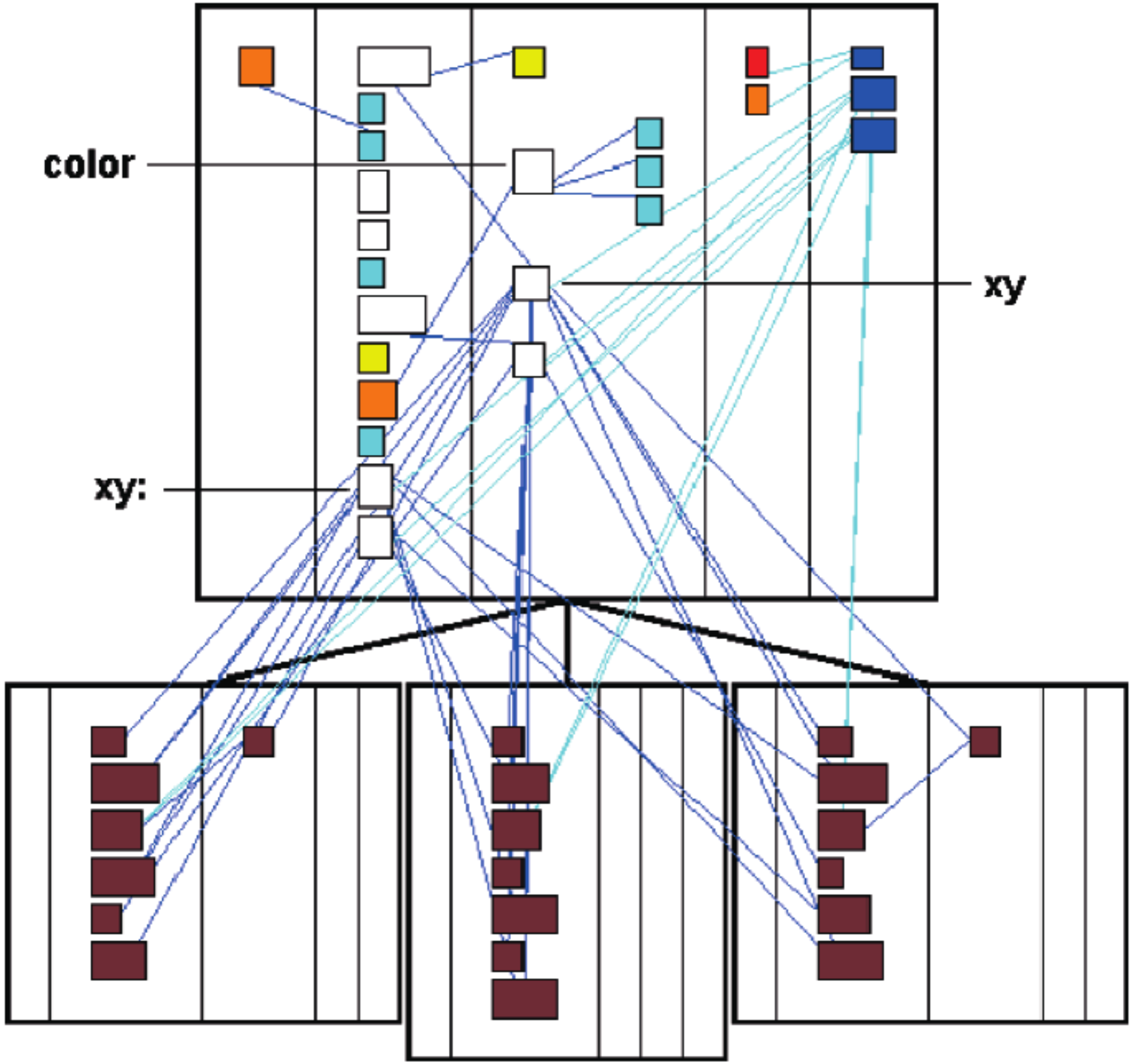



ModelFacade
ParserDisplay
FigEdgeModelElement
FigNodeModelElement
Parser
GeneratorCpp
CoreFactory

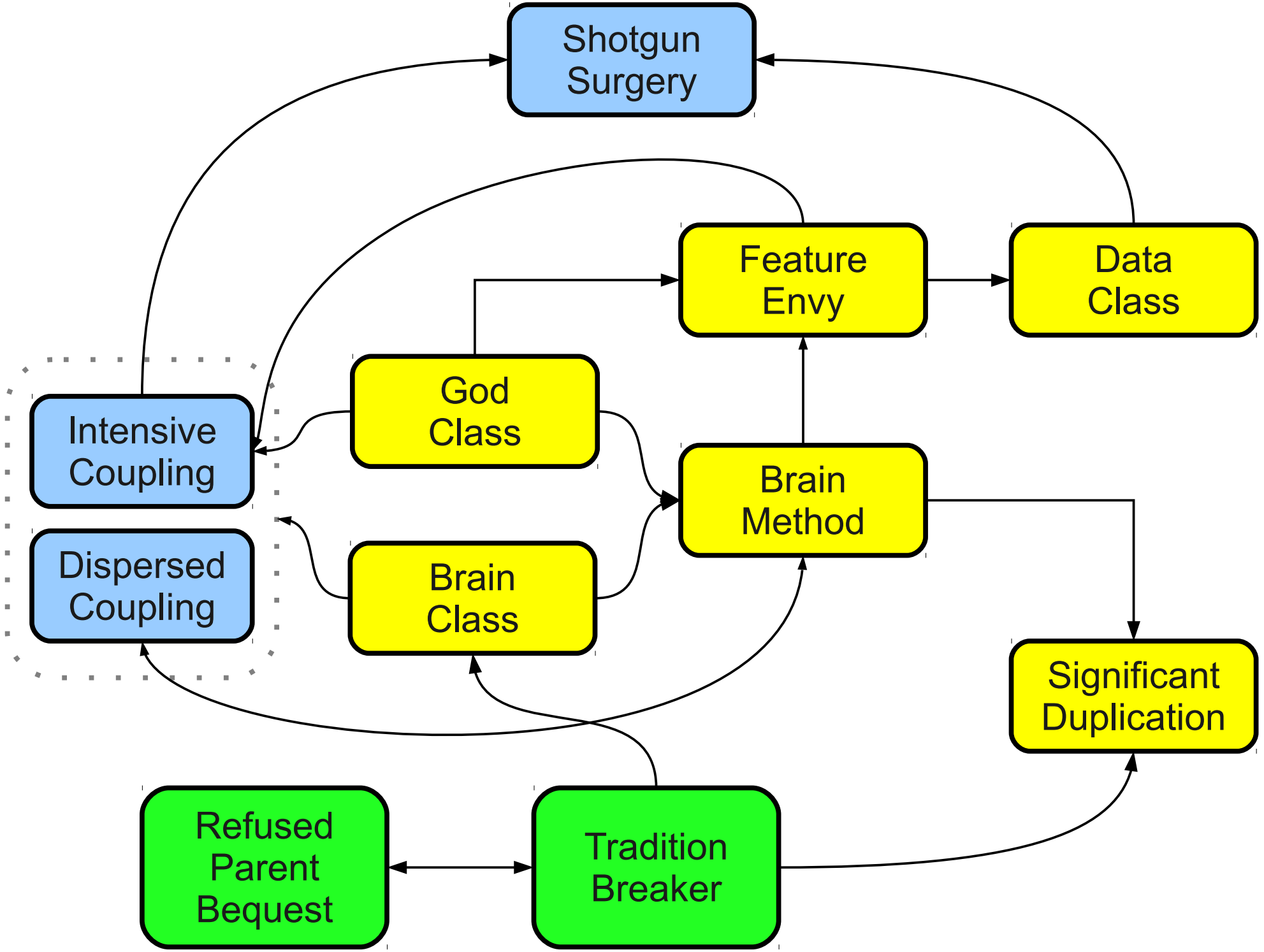

Number of Invocations
Method
Number of Lines of Code
Number of external accesses
Attribute
Number of internal accesses
Method invocation
Direct access
Abstract Method
Overriding Method
Delegating Method
Extending Method
Constant Method
Read Accessor Method
Write Accessor Method


Initialization
External Interface
Internal Implementation
Accessors
Attributes
Invocation Sequence


initialize
asPointArray
boundingBox
color:
colorToLispList
copy
defaultLispConsClass
defaultLispNilClass
generateIdString
hasColor
isArc
isCompound
isGraph
isNode
isPrimitive
kindName
lispNil
pointsDo:
renderOn:
rotatedBy:
saveOn:
scaledBy:
showLispList
transformation
translatedBy:
value
displayObject
createDisplayObject
flushDisplayObject
lispCons
showLispList:
toLispList
transform:
color
setValue:
displayColor
displayObject


color
xy
xy:

Disharmonias de Diseño


Shotgun Surgery
Feature Envy
Data Class
God Class
Intensive Coupling
Brain Method
Dispersed Coupling
Brain Class
Significant Duplication
Refused Parent Bequest
Tradition Breaker

Estrategias de Detección

1

Reglas de Diseño (Informales)

1

Reglas de Diseño
(Informales)

"Tener cuidado con clases que acceden directamente a datos de otras"

“Ojo con los métodos grandes“

1

Reglas de Diseño
(Informales)

“Tener cuidado con clases que acceden directamente a datos de otras“

1

Reglas de Diseño
(Informales)

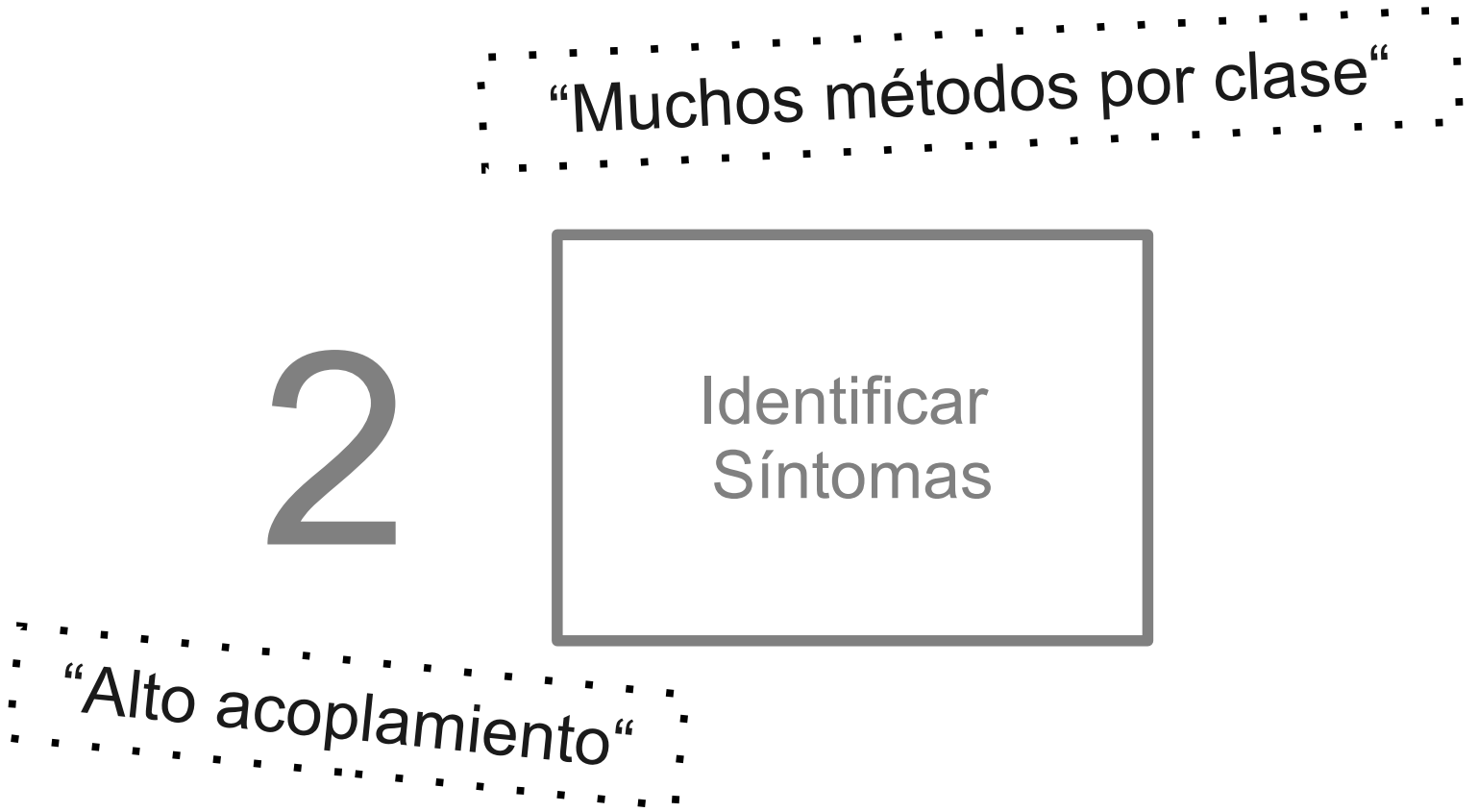
“Ojo con los métodos grandes“

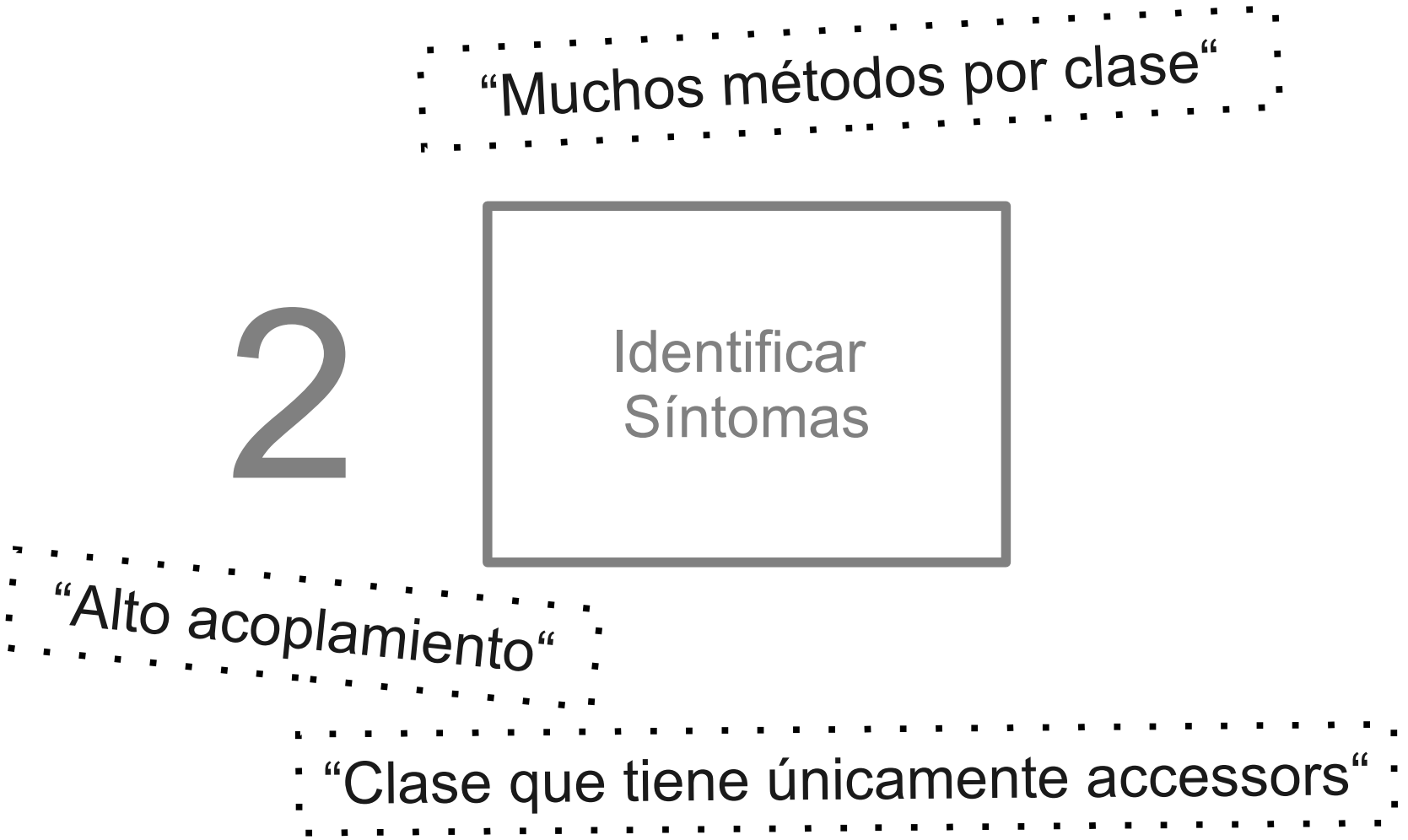
“Tener cuidado con clases que acceden directamente a datos de otras“

“Cuidado con las clases que tienen muchas responsabilidades“

2


2
Identificar
Síntomas
“Alto acoplamiento“


“Muchos métodos por clase“
2
Identificar
Síntomas
“Alto acoplamiento“


“Muchos métodos por clase“
2
Identificar
Síntomas
“Alto acoplamiento“
“Clase que tiene únicamente accessors“

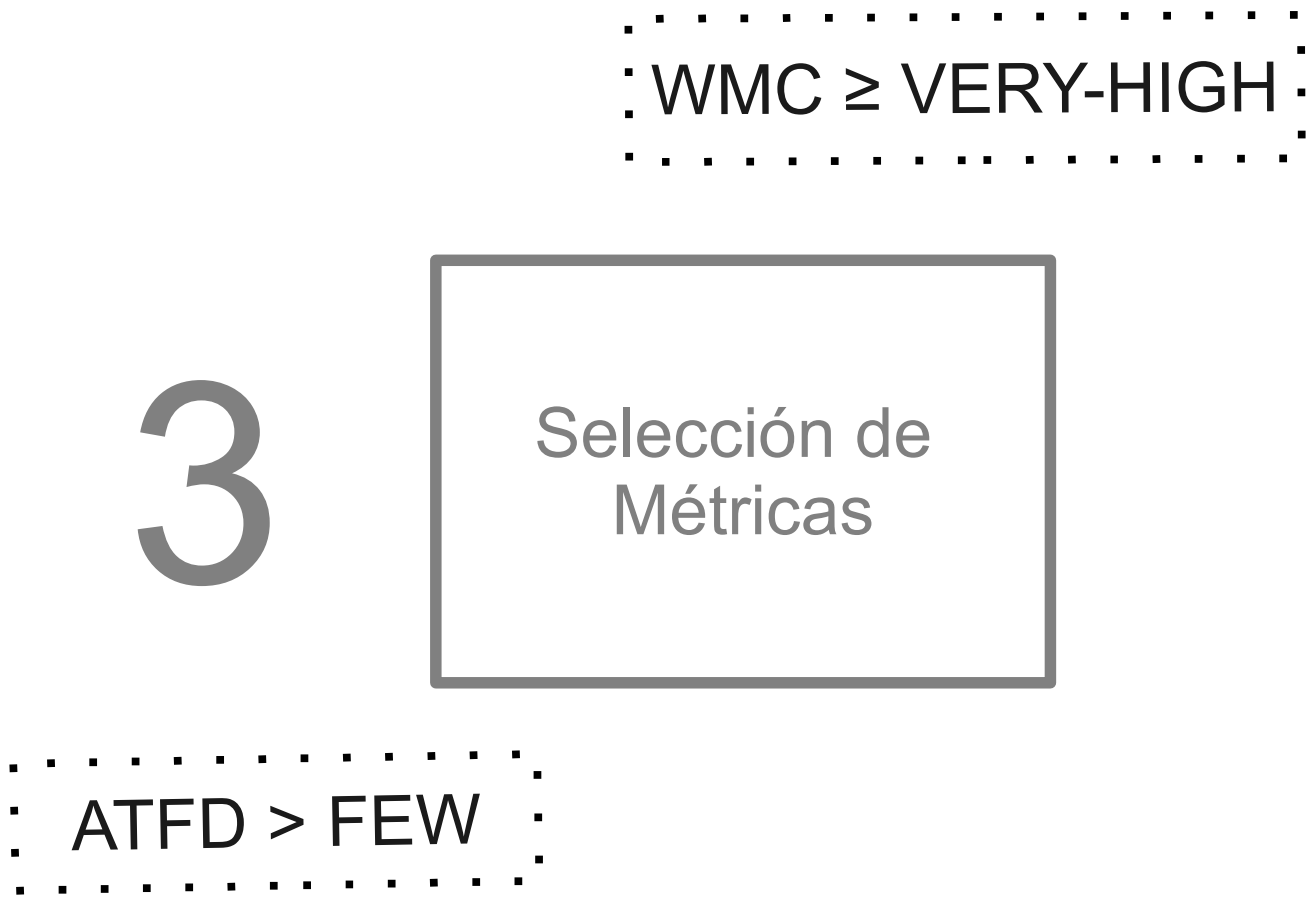
3

Selección de Métricas

3

Selección de Métricas

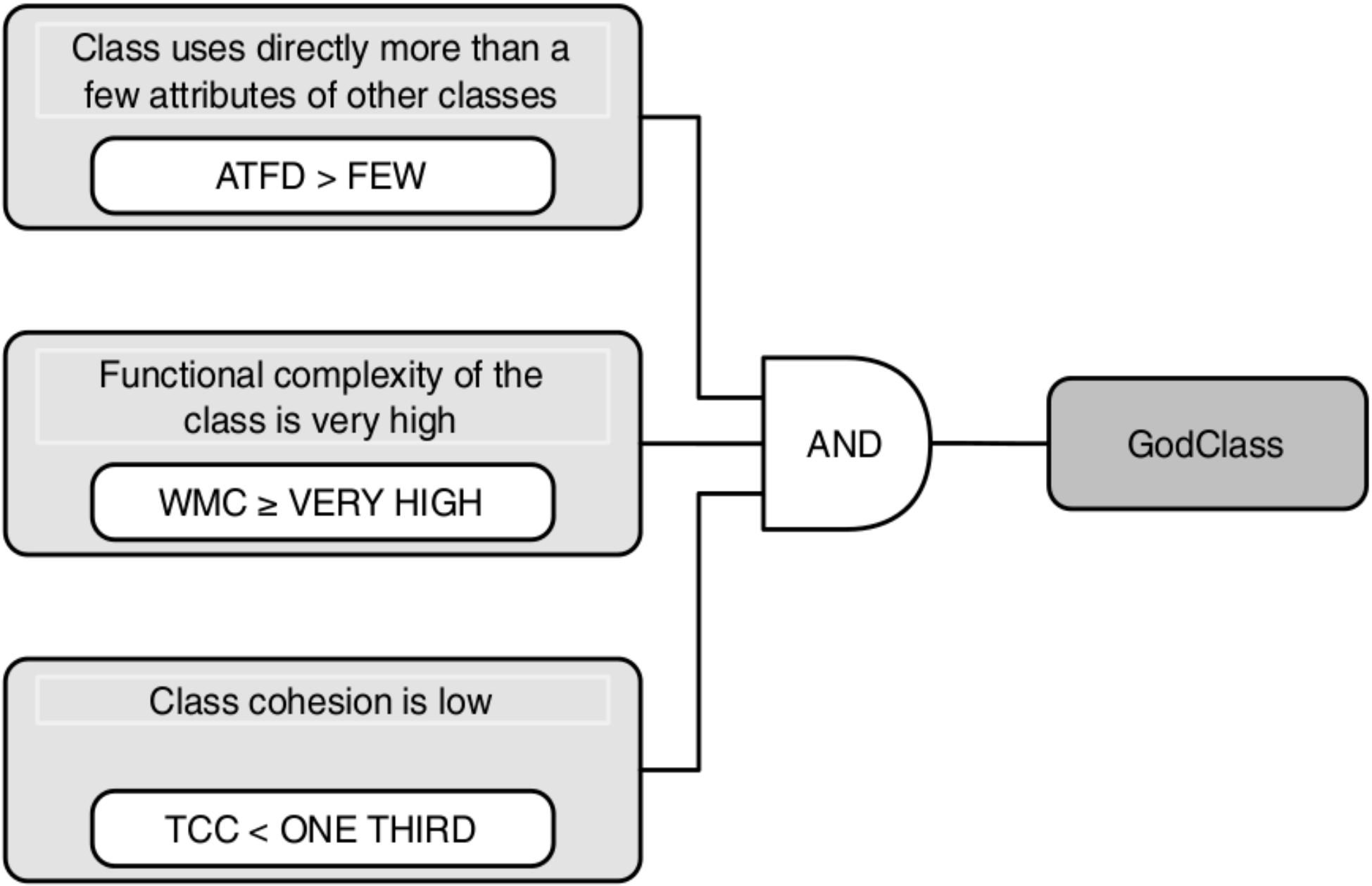
ATFD > FEW

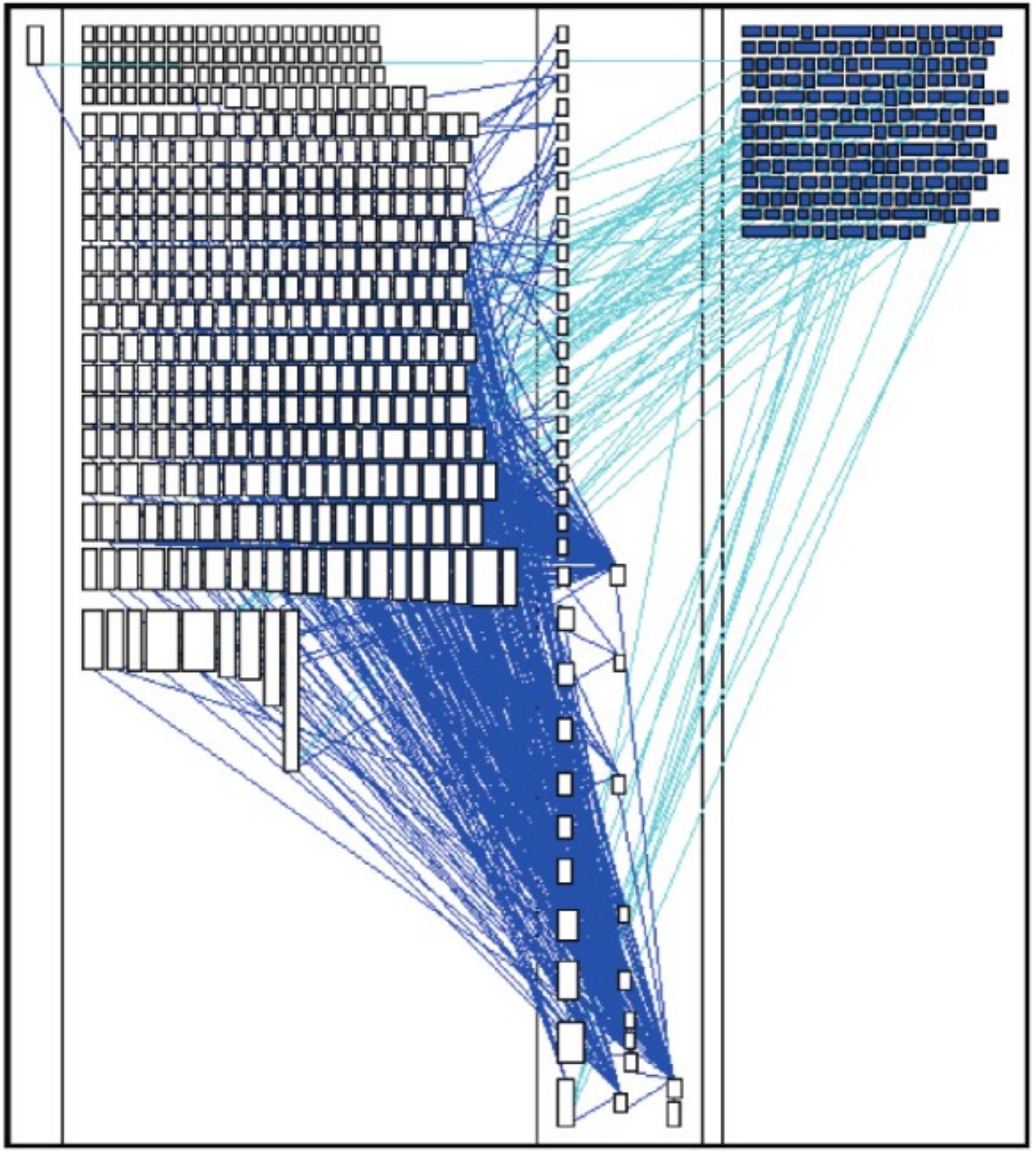

WMC ≥ VERY-HIGH
3
Selección de Métricas
ATFD > FEW

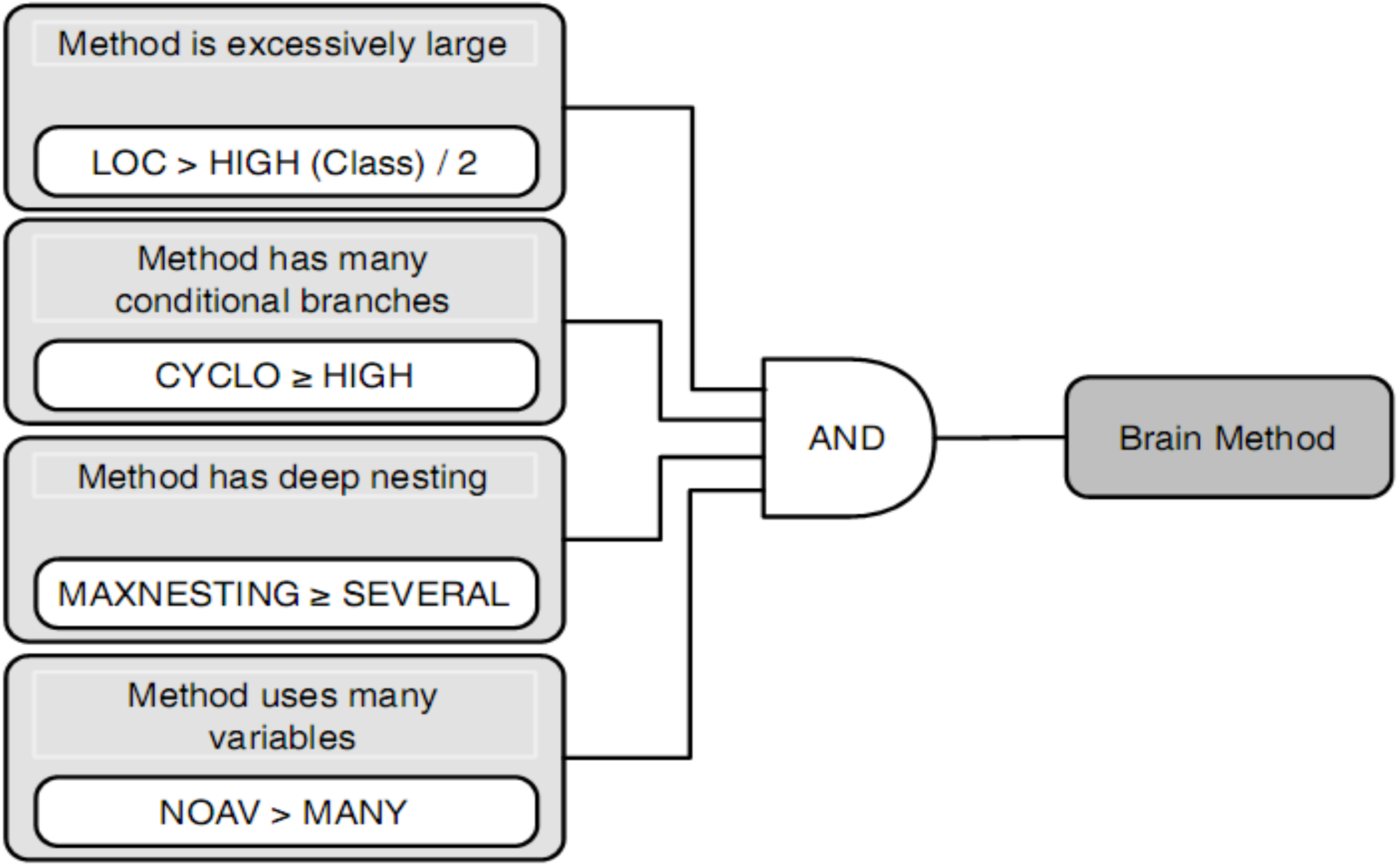

WMC ≥ VERY-HIGH
3
Selección de Métricas
ATFD > FEW
TCC < ONE-THIRD

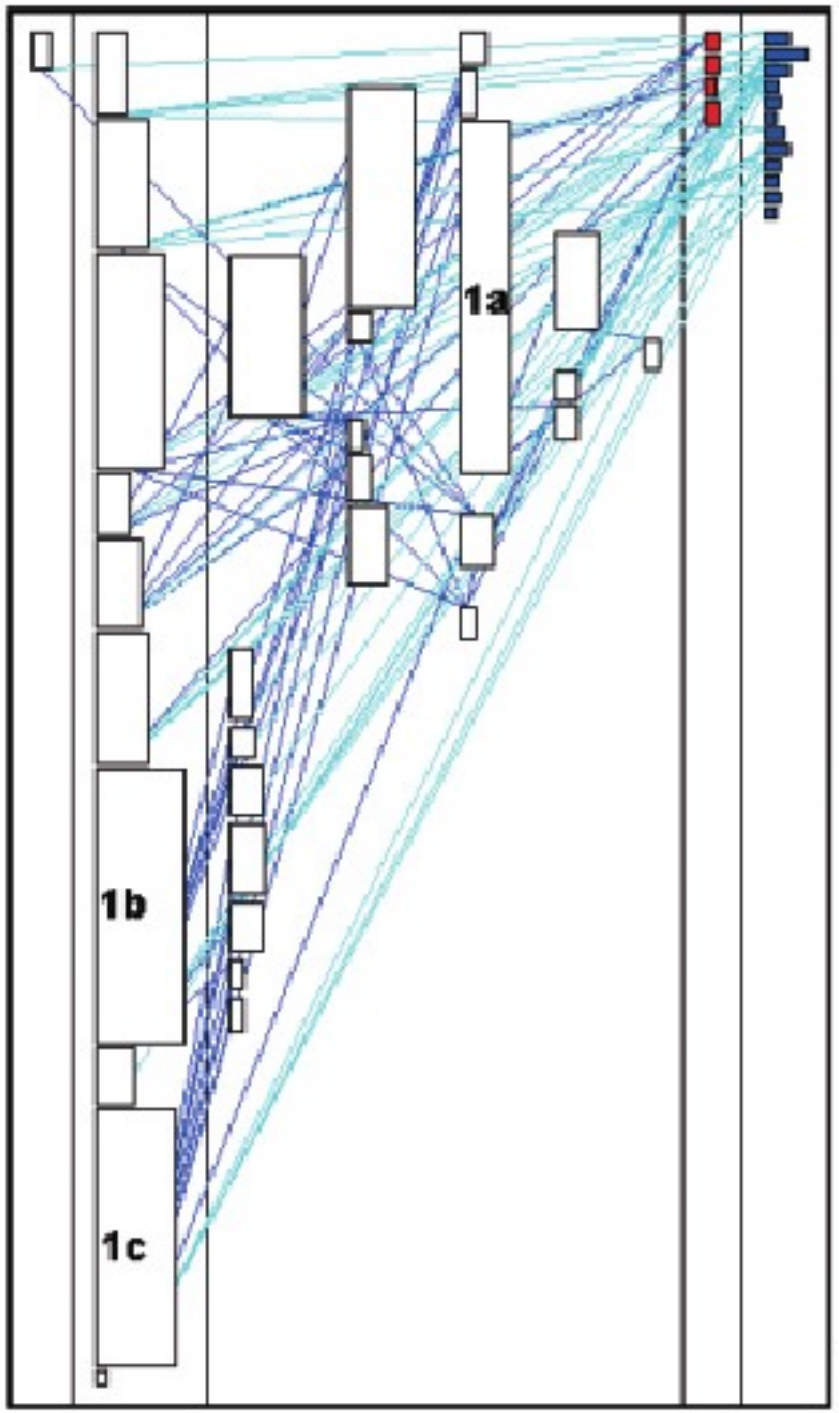
4

Estrategia de Detección

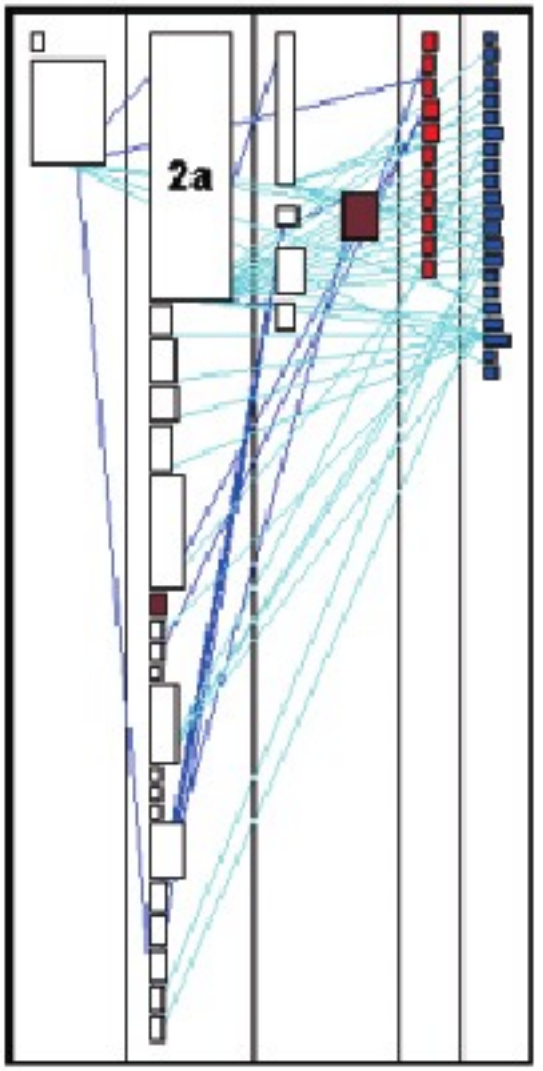

Class uses directly more than a few attributes of other classes
ATFD > FEW
Functional complexity of the class is very high
WMC ≥ VERY HIGH
Class cohesion is low
TCC < ONE THIRD
AND
GodClass




Method is excessively large
LOC > HIGH (Class) / 2
Method has many conditional branches
CYCLO ≥ HIGH
Method has deep nesting
MAXNESTING ≥ SEVERAL
Method uses many variables
NOAV > MANY
AND
Brain Method



Modeller



ProjectBrowser

Conformity Strategies

God classes
Class1
Class2
Class3
Class4
Class5

CYCLO > 10

$$\text{HigherThan} = \frac{\text{actualValue}}{\text{metricThreshold}}$$

$$\text{LowerThan} = \frac{\text{metricThreshold}}{\text{actualValue}}$$

$$or = \max(rating(A), rating(B))$$

$$and = \min(rating(A), rating(B))$$

CYCLO > 10

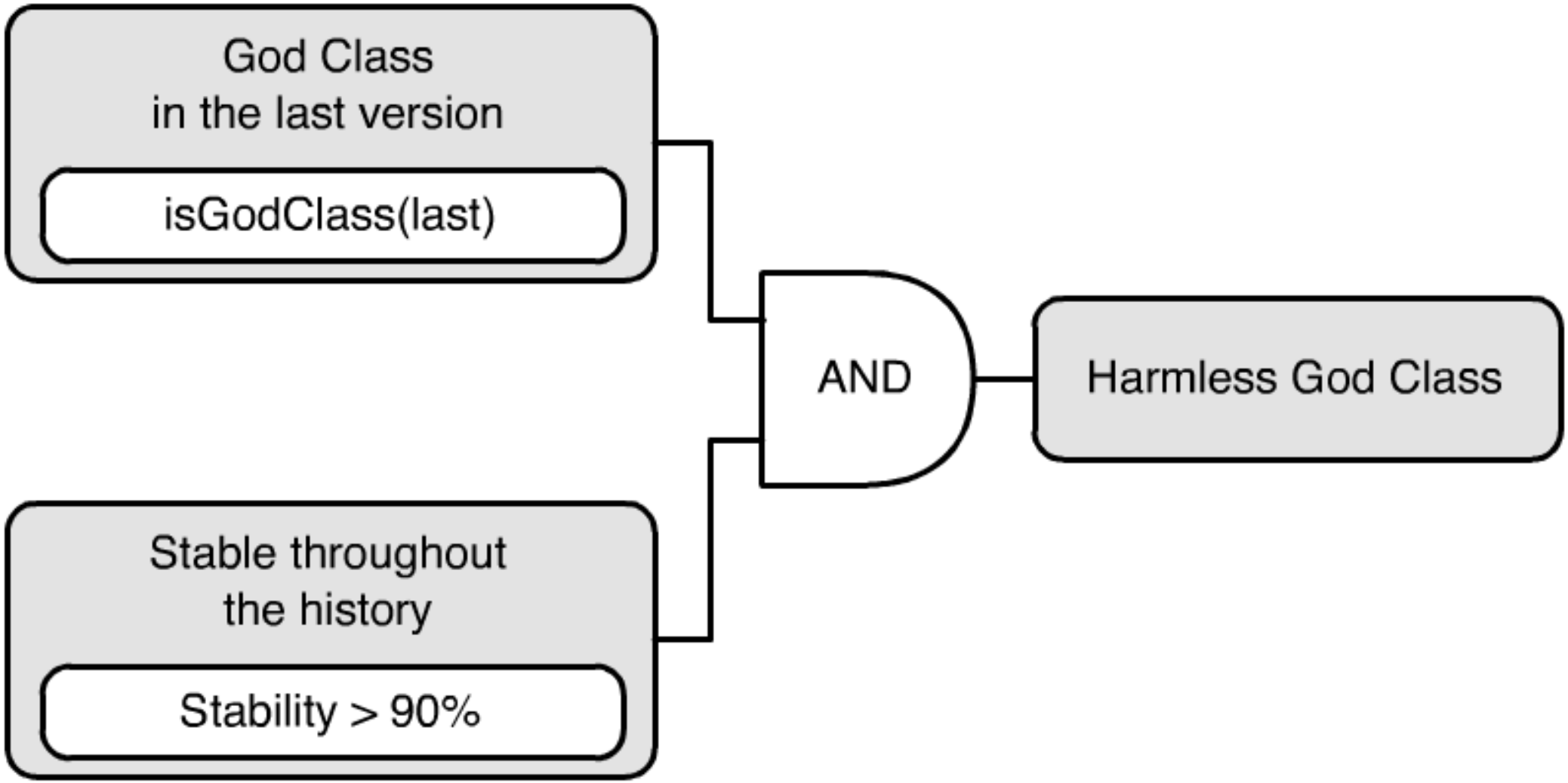
CYCLO = 5 → 50%

CYCLO = 10 → 100%

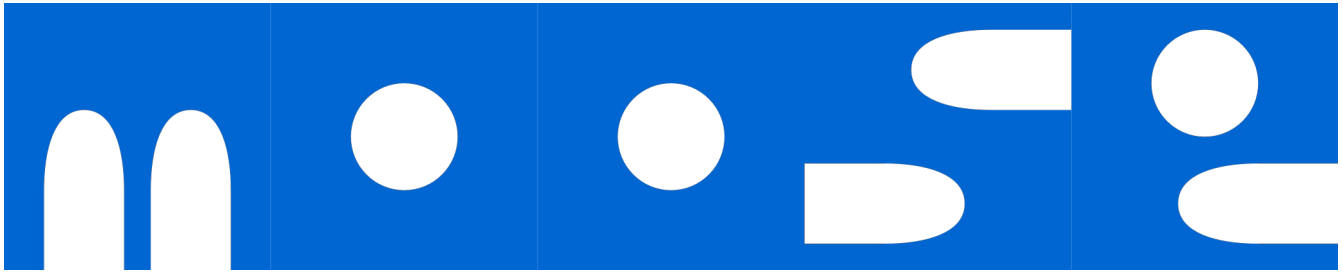
CYCLO = 20 → 200%

God classes	Conformity
Class5	233%
Class3	90%
Class1	83%
Class2	76%
Class4	70%

Usando la Historia


God Class
in the last version
isGodClass(last)
AND
Harmless God Class
Stable throughout
the history
Stability > 90%

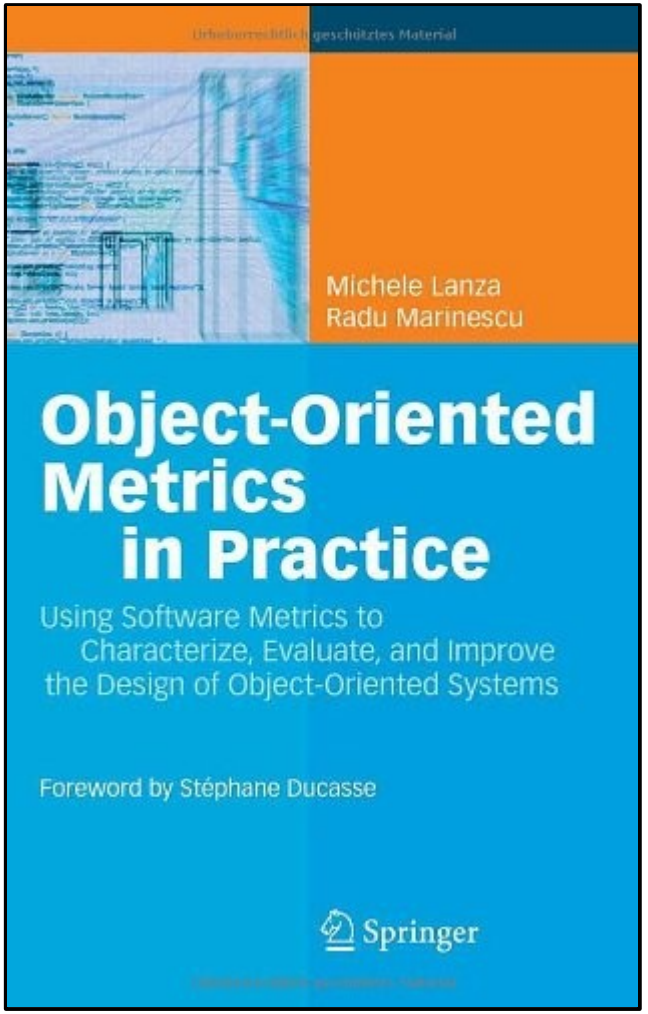
Herramientas



http://www.moosetechnology.org

http://www.intooitus.com/products/infusion

Bibliografía


Urheberrechtlich geschütztes Material
Michele Lanza
Radu Marinescu
Object-Oriented Metrics in Practice
Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems
Foreword by Stéphane Ducasse
Springer

http://www.moosetechnology.org

http://www.loose.upt.ro/reengineering

http://scg.unibe.ch/research

http://www.tudorgirba.com/projects/hismo

http://scg.unibe.ch/archive/external/Lung04a.pdf