Code Quality Metrics Derived from Software Design

Omar Masmali & Omar Badreddin

Department of Computer Science

The University of Texas, El Paso, Texas USA
email: oamasmali@miners.utep.edu & obbadreddin@utep.edu

Resume

- PhD Candidate, Department of Computer Science
- The University of Texas, El Paso, Texas USA
- He received his Bachelor of Computer and Information Systems from King Faisal University, Saudi Arabia in 2003.
- He received his Master’s degree in computer science with specialization in Software Engineering from University of Wollongong, Australia in 2008.
- He is working as an instructor at Dammam College of Technology in Saudi Arabia since 2004.

Research interest

- His research interests lie in the field of software engineering. More specifically, he is interested in software quality and sustainability, software sustainability metrics that can be applied proactively to measure software design and code quality. His research goal is establishing a comprehensive software sustainability metrics that can be applied proactively to measure software design complexity and code quality.

PROBLEM

Current code quality quantification methodologies adopt metrics with rigid thresholds.

These methodologies do not adequately consider variations in

Development technologies	The architectural roles of various code and design elements.

PROPOSED SOLUTION

Our proposed approach derives code quality metrics and their dynamic threshold values from software design.

The proposed approach includes:

- *Complexity Metrics*
- *Fuzzy Metrics*

Complexity Metrics

Attribute Complexity →
$$Att_{comp} = (V_{Att.} * Co_{Rate}) + (T_{Att.} * Co_{Rate})$$

Method Complexity →
$$Method_{comp} = (V_{Meth.} * Co_{Rate}) + (R_{Meth.} * Co_{Rate}) + \left(\sum_{i=1}^{n} (P_{Meth.} * Co_{Rate})\right)$$

Association Complexity →
$$Asso_{comp} = \left(\sum_{i=1}^{n} (IN_{As.} * Co_{Rate})\right) + \left(\sum_{i=1}^{n} (OUT_{As.} * Co_{Rate})\right)$$

Class Complexity →
$$Class_{comp} = \left(\sum_{i=1}^{n} Att_{comp}\right) + \left(\sum_{i=1}^{n} Method_{comp}\right) + Asso_{comp}$$

Complexity Metrics

- (Complexity Rate)

Element	Scope	Classification	Examples	Complexity Rate
Attributes & Methods	Visibility	Primitive	Private	1
		Simple	Protected, Package	2
		Complex	Public	3
	Type	Primitive	int, char, boolean	1
		Simple	float, long, double, str	2
		Complex	array, struct, tuple, date, time, list, map	3
		Derived	object, array of complex types	4
Association	Incoming & Outgoing	Primitive	1 to many	1
		Simple	many to many, 1 to 1	2
		Complex	all others (such as n .. m to many, etc..)	3

Fuzzy Metrics (Class)

Fuzzy Large Class → $FuzzyMetric(Class) = LOC(class) - ELOC(class)$

Expected Lines of Code → $ELOC(Class) = Class_{comp} * Class(LOC_{factor})$

Lines of Code Factor → $Class(LOC_{factor}) = \frac{LOC(Total) * LOC(Average)}{Class_{comp}(Total) * Class_{comp}(Average)}$

Fuzzy Metrics (Class)

Fuzzy Long Method → $FuzzyMetric(Method) = LOC(Method) - ELOC(Method)$

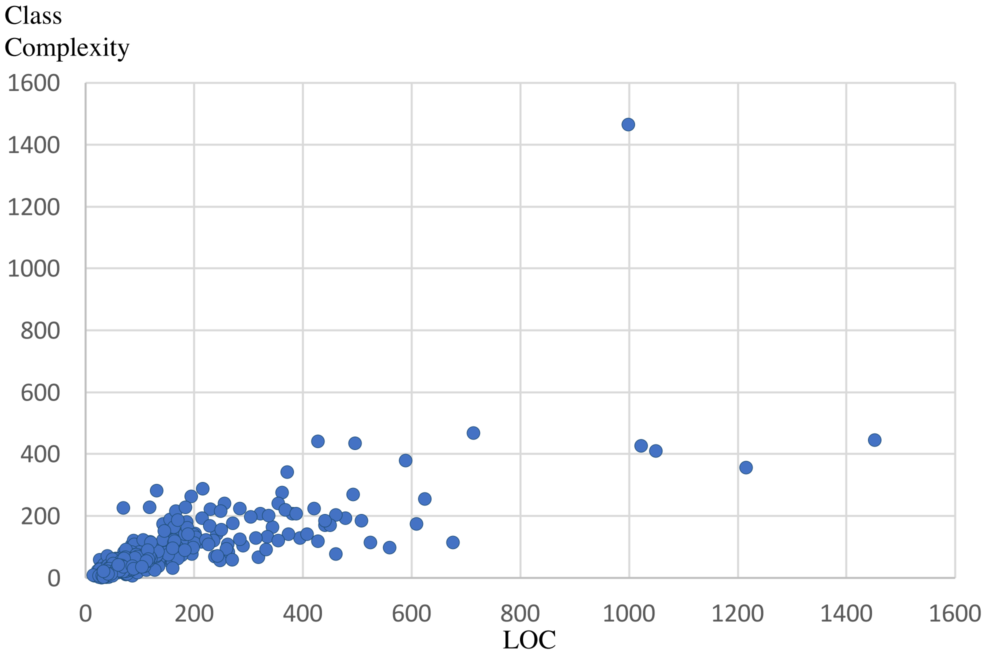
Expected Lines of Code → $ELOC(Method) = Method_{comp} * Method(LOC_{factor})$

Lines of Code Factor → $Method(LOC_{factor}) = Method_{comp}(Average)$

Evaluation Studies

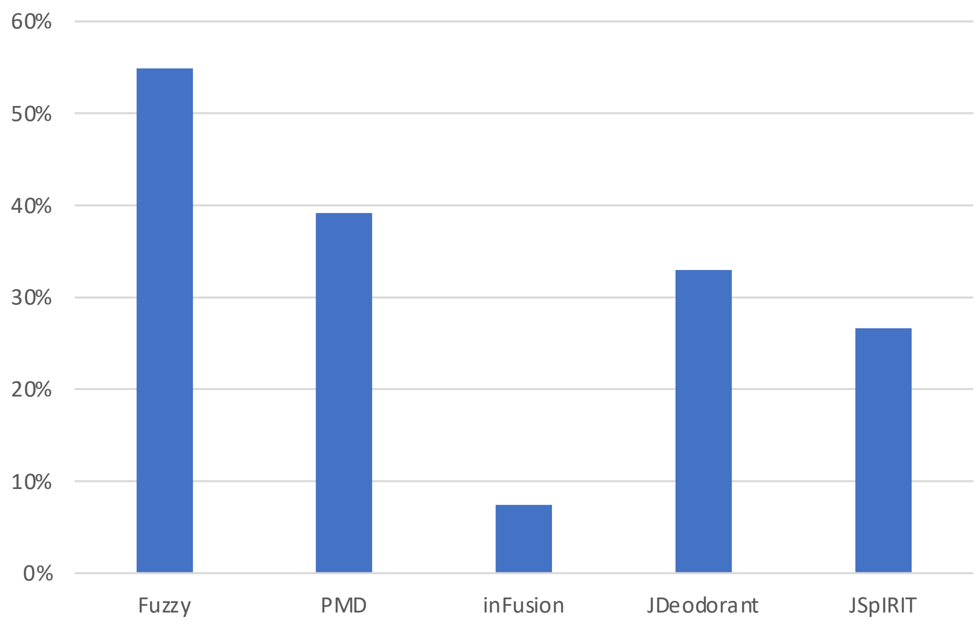
- Complexity
  - To investigate whether there is a correlation between the proposed UML Class Diagram complexity and the associated code
- Fuzzy
  - Comparative study between the proposed approach and other detection tools

Results (Complexity Metrics)



Correlation between class complexity and Lines of Code (LOC) is 0.82

Results (Fuzzy Metrics)



The total F1 Score for classes of each tool.

Our approach with the highest F1 score with 55%

Future Work

- Theoretical evaluation of the complexity metrics by using Weyuker's nine properties model.
- Evaluate whether the metrics derived from software designs provide a better characterization of codebase quality and sustainability than alternate traditional metrics.
- Quantify thresholds for the fuzzy code smells derived from the software design.
- Compare our new fuzzy code smells with code smells resulting from code smells detection tools for different codebases.