

# Design Metrics Behavior on Object Oriented Software

Kavita Srivastava

*Faculty of BCA, Department of Business Management  
And Entrepreneurship, Dr. R. M. L. A. University Ayodhya*

**Abstract-** Metrics is important factor to measurement of any software system. It provided help designer's capability to achieve the complete design. Metric based model for 'Assessment of Object Oriented Software' has evaluated the particular UML diagram. Metrics values have presented through specific formula and graph. The aim of this article is to assess the metrics values for inclusion of the assessment of object oriented software.

**Keywords-** Object Oriented Metrics, Software development life cycle, Software Efficiency.

## 1. INTRODUCTION

Different programming metrics related to software quality affirmation have been proposed in the past are so far being proposed [7]. A couple of books demonstrating such metrics exist [2]. By far most of these metrics are available to all programming dialects; a couple of metrics apply to a specific course of action of programming dialect. Among metrics of this sort, are those that have been proposed for protest- arranged programming dialect. Nowadays, a quality originator can investigate a huge measure of question - arranged metrics [1, 8]. The inquiry presented is not the nonappearance of metrics but instead the decision of those metrics which meet the specific prerequisite of each delicate product venture [4]. A quality draftsman needs to go up against the issue of selecting the fitting game plan of metrics for his item estimations [6, 9]. Different protest - arranged metrics misuses the data grabbed from metrics used as a piece of composed programming and change such estimations so as to satisfy the necessities of protest - arranged programming. On the other hand, other protest -situated metrics have been made especially for question - situated programming and

it is unimportant to apply them to organized programming [3, 5]. The abovementioned figure exhibits the different leveled structure of the metrics [10].

## 2. METRICS EVALUATION

There are framework metrics that can be gotten from class metrics with measurements, as relative measures, recognizing frameworks that go amiss from the standard. Irregular patterns or qualities of the framework under development and be spotted and amended. The number of metrics values has calculated through the UML diagram. The UML diagram presented the figure 1. Table 1 presented the metrics value through the UML diagram. Authors are encouraged to have their contribution checked for grammar.

The text is to be typeset in 10 pt roman, single spaced with baselineskip of 13 pt. Text area is 5 inches in width and the height is 8 inches (including running head). Final pagination and insertion of running titles will be done by the publisher. Upon acceptance, authors are required to submit their data source file including postscript files for figures.

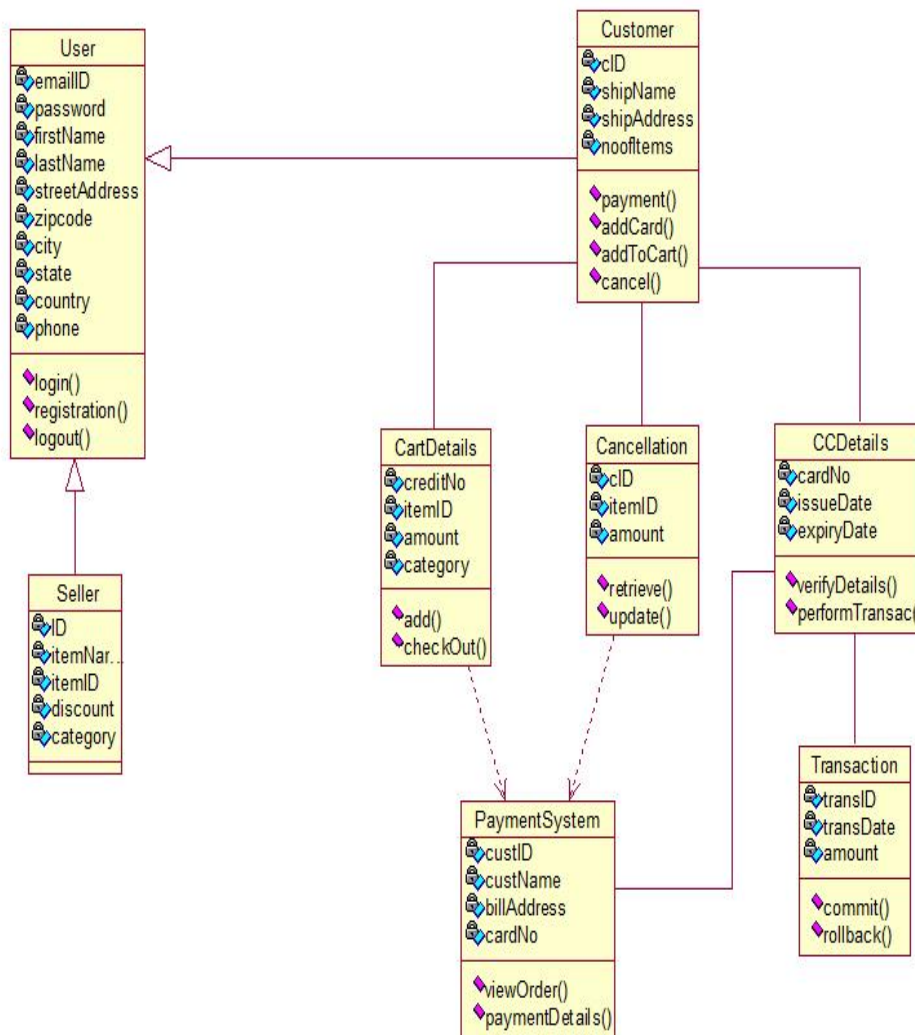


Figure 1 UML diagram of Online Shop

Table 1 Metrics Value				
Project	CC	MFA	LCOM	DIT
P <sub>1M</sub>	0.369	0.963	0.666	0.456
P <sub>2M</sub>	0.478	0.852	0.522	0.963
P <sub>3M</sub>	0.777	0.423	0.658	0.869
P <sub>4M</sub>	0.4444	0.258	0.258	0.856
P <sub>5M</sub>	0.744	0.366	0.436	0687
P <sub>6M</sub>	0.496	0.569	0.855	0.995
P <sub>7M</sub>	0.159	0.756	0.258	0784
P <sub>8M</sub>	0.888	0.693	0.555	0.7444
P <sub>9M</sub>	0.159	0.666	0.357	0.493
P <sub>10M</sub>	0.558	0.9930	.490	0.708
P <sub>11M</sub>	0.502	0.559	0.745	0.369
P <sub>12M</sub>	0.789	0.456	0.144	0.369
P <sub>13M</sub>	0.555	0.606	0.970	0.854

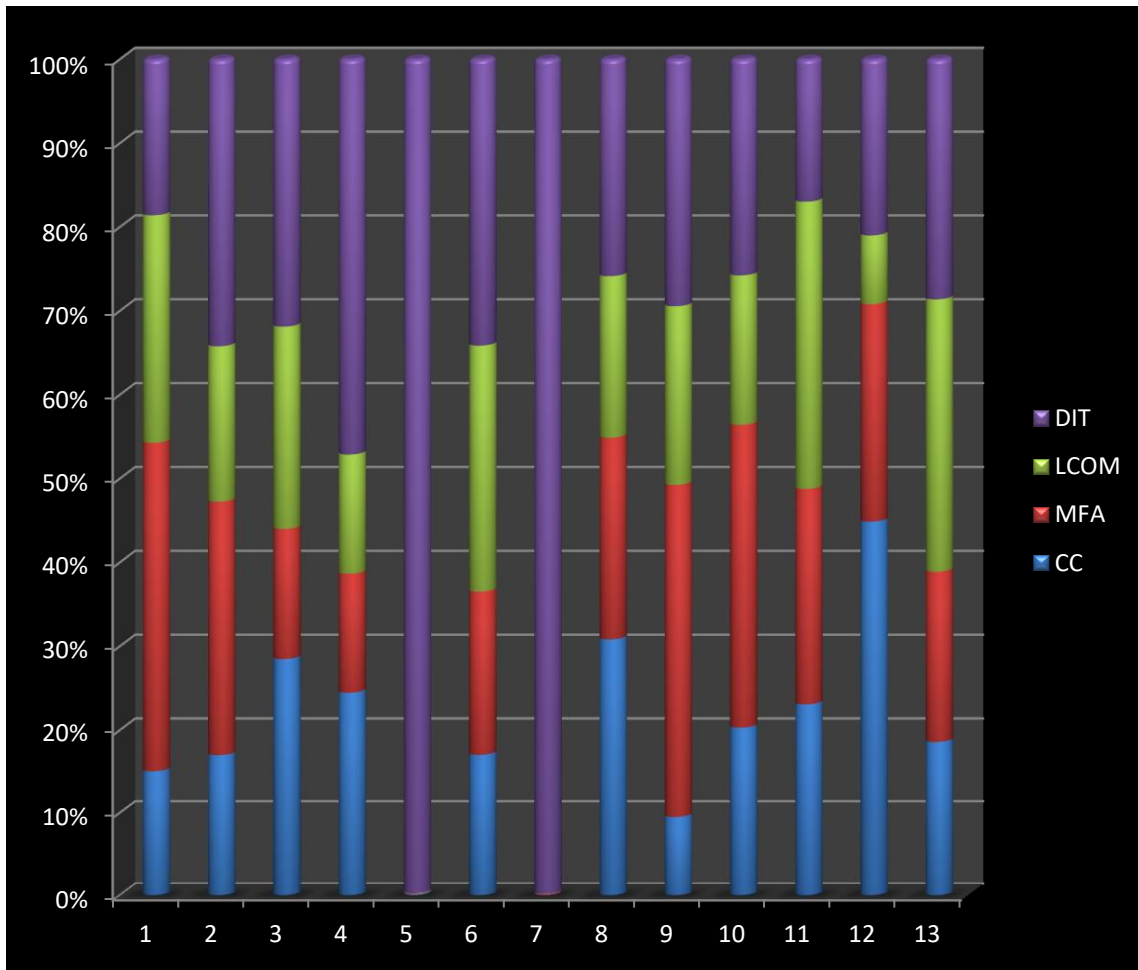


Figure 2 Comparative analyses of metrics Value

### 3. CONCLUSION

Object-oriented metrics have turned into a fundamental piece of object innovation and in addition programming building. Exploratory investigation of experimental information is given to relate the metrics to profitability, revamp exertion, and plan exertion on three business object-oriented frameworks. The observational outcomes recommend that the metrics give huge logical power for varieties in this UML diagram. We have pictured that increasingly metric is foresee the esteem classes and it is the most valuable to enhance the quality and dependability of the plan.

### REFERENCES

- [1] Brij Mohan Goel, Pradeep Kumar Bhatia, "Investigation of Reusability Metrics for Object-Oriented Designing", Proceeding of NCETCIT, GVM IT&M, Sonipat, pp. 104-110, May 2014.
- [2] Shyam R. Chidamber, Chris F. Kemerer, "A metrics suite for object oriented design", IEEE transactions on software engineering, vol. 20, no. 6, pp.476-493, June 1984.
- [3] ChidamberShyam, Kemerer Chris, Darcy David, "Managerial use of metrics for object-oriented software: an exploratory analysis", IEEE Transactions on software Engineering, vol. 24, no. 8, pp.629-639, August 2008.
- [4] Kramer, S. and Kaindl, H., "Coupling and cohesion metrics for knowledge-based systems using frames and rules", ACM Trans. on Soft. Engineering and Methodology (TOSEM), vol. 13, no. 3, July 2004, pp. 332-358
- [5] J. Bansiya, C. G. Davis "A Hierarchical Model for Object-Oriented Design Quality Assessment" IEEE Transactions on Software Engineering, 2014
- [6] C. Shyam and C. F. Kemerer "Towards a Metrics Suite for Object Oriented Design" Proceeding on Object Oriented Programming Systems, Languages and Applications Conference ACM, Vol. 26, Issue 11, Nov 2011

- [7] Subramanyam, R., Krishnan, M.S., “Empirical analysis of CK metrics for object-oriented design complexity: Implications for software defects”, *IEEE Transactions on Software Engineering*, vol. 29, no. 4, pp.297-310, April, 2014.
- [8] Ahmed M. Salem, Abrar A. Qureshi, “Analysis of inconsistencies in object-oriented metrics”, *Journal of Software-Engineering and Applications*, vol. 4, pp.123-128, January 2011.
- [9] Yuming Zhou, Hareton Leung, “Empirical analysis of object-oriented design metrics for predicting high and low severity faults”, *IEEE transactions on software engineering*, vol. 32, no. 10, pp.771 – 789, October 2006.
- [10] M.RizwanJameel Qureshi, Waseem Qureshi, “Evaluation of the design metric to reduce the number of defects in software development”, *I.J. Information Technology and Computer Science*, vol. 4, pp.9-17, 2009.