

Object-Oriented Metrics for Quality Improvement of Object-Oriented Software

Mohit Kumar Sharma, Sandeep Ranjan, Amardeep Gupta

Abstract: Software metric is a computation of characteristics of a programming applications for quality enhancement. Software metrics are used to access the productivity and efficiency of a software product. These are helpful to understand the technical procedure used to create software. Software metric enables software engineers to assess software quality, software process improvement, monitoring and controlling of software. Object-Oriented Software Development is related to real-world objects and their characteristics creation instead of working in software applications. Class Objects have their own internal data structure, which defines their data and methods. Object-Oriented design restrained all the properties and worth of software that is allied to any large or small project. Object-Oriented metric is a measurement term in which a Object-Oriented Software holds features. These are guidelines that give an indication of the progress that a project has made quality.

Index Terms: Object-Oriented Metrics, Quality, Complexity

I. INTRODUCTION

Software metric is a measurement of the characteristics of a software product for quality improvement. Software metrics are used to access the productivity and efficiency of a software product. These are helpful to understand the technical procedure used to create software. Software metric enables software engineers to assess software quality, software process improvement, monitoring and controlling of software [1].

Software metric as “A quantitative measure of the degree to which a system, component, process possesses a given attribute” [2].

Object-Oriented Design is related to real-world objects and their characteristics creation instead of working in software applications. Class Objects have their own internal data structure, which defines their data and methods. Object-Oriented metric is a measurement term in which a Object-Oriented Software holds features. These are guidelines that give an indication of the progress that a project has made quality [3].

Object-Oriented design restrained all the properties and worth of software that is allied to any large or small project. [4].

Software quality and reliability is concerned with failure free application operation. There is exigency for

Object-Oriented Metrics due to visibility, planning and control, quality, productivity [5].

II. REVIEW OF LITERATURE

Reliability in software development is the probability of fault-free application working in a particular work area. Software reliability of Object-Oriented software products based on their structure with Object-Oriented metrics as C-K metrics. Object-Oriented metrics are utilized to estimate software reliability and final software reliability is resulted from these estimated metric values and proposed different steps for methods followed for computation of metrics. WMC, LOC, DOI, NOC and RC are utilized for maintaining software reliability [6].

Conventional static metrics to be insufficient for modern Object-Oriented software, as object-oriented dynamic metrics contain features such as inheritance, polymorphism, dynamic binding etc. Dynamic metrics are preferred in place of conventional static metrics due to different features. Object-Oriented dynamic metrics computed based on data collected during execution of applications and measure the dynamic behavior of software [7].

Different approaches are utilized through software application cost estimated. Tools are used to compute LOC, functions points and Object-Oriented points. QMOOD metrics, GQM metrics, MOOSE metrics and EMOOSE metrics are suggested [8].

Evaluation of software complexity metrics is applied for Object-Oriented software by calculating source codes created in Java programming. The complexity of Java classes in software is computed by using software tools. The output shows that these complexity metrics can be used to estimate software quality [9].

Interpretation of metrics from coding is now a challenging activity and to find percentage of coding is utilized from the present one using inheritance by using C-K metrics. Object-Oriented applications by parameterized constructor in C++ programs is more reusable [10].

Automated software metrics support for software engineers and users to measure the design or coding of the open access applications and find quality of the software with hierarchical metric model. The main objective is to the automatic quality evaluation of source code for open access applications by study metrics and develop a tool. It will be based on software metrics that, combined and configured by experts, will provide an automatic investigation of features such as flexibility, precision, modularity, and software maintenance [11].

Software metrics plays vital role to measure software quality. A number of metrics related to software complexity, quality, reusability, reliability, maintainability have been developed in the past and are still being proposed. Software metrics

Revised Manuscript Received on January 05, 2020.

Mohit Kumar Sharma, Research Scholar, Faculty of Computational Science, GNA University, Phagwara, Punjab, and Assistant Professor (HOD), Computer Science, J.C. D.A.V. College, Dasuya, Punjab, India

Dr. Sandeep Ranjan, Assistant Professor, Faculty of Computational Science, GNA University, Phagwara, Punjab, India

Dr. Amardeep Gupta, Principal, J.C. D.A.V. College, Dasuya, Punjab, India

are tools for controlling software complexity. Cognitive with non-cognitive complexity metrics in Object-Oriented design with respect to the complexity of a class, code, inheritance, interface and polymorphism are discussed [12].

Object-Oriented applications requires reliable software metrics to find software importance. An empirical study had been done by public dataset KC1 from NASA project database. Statistical techniques like correlation analysis and regression analysis are conducted to validate results. After interpretation, metrics SLOC, RFC, WMC and CBO are important and resulted as quality indicators [13].

Object-Oriented metrics analyze the difference between all metrics through the comparison table. A case study has been taken that how to evaluate the reusability by using machine learning regression algorithms and proved. Finally, the comparison of novel regression algorithms and mentioned with a tabular form then WEKA software to compare and plotting the graph used [14].

Several cohesion complexity metrics for Object-Oriented programming with the primary objectives as to propose several Object-Oriented software metrics that identifies the types and level (such as low, medium and high) of cohesion exist in the module and to assess the complexity of the software modules and to perform the theoretical, empirical and statistical validations of the proposed metrics and to develop an automated software tool for the proposed cohesion metrics and find difference between results of the suggested cohesion metrics for the state of the art cohesion complexity metrics. The proposed cohesion complexity measurements also improve the functional quality, productivity, reusability and organization aspects of the project [15].

Possible causes of bugs affecting software quality and the major contributors of software bugs with expert opinions backed with data set. To develop software bug prediction models using SRGM through software metrics and significantly correlate bug indicators as model inputs. To compare the relative effectiveness of bug indicators towards the prediction of bugs using Object-Oriented projects. To predict various types of bug severities on the basis of increasing level of inheritance using various algorithms of the neural network on open source data sets and industry data sets. Measure the accuracy of these algorithms to identify the best algorithm for bug prediction using Object-Oriented projects [16].

C-K metrics are evaluated to compute quality by open access software as Statcato. Computations did in every class and every software version and to interpret the software quality [17].

Method complexity, message complexity, attribute complexity, weighted class complexity, and code complexity metrics are presented. Metrics suite was evaluated theoretically using measurement theory and Weyuker's properties, practically using Kaner's framework and empirically using thirty projects [18].

A Complete survey on software metrics and its applications has been carried in the past and the main objective of this survey is to point out reusability factor rather than software coding [19].

Object-Oriented software computation results to be a important tool for software effectiveness and reliability. Interpretation of different Object-Oriented metrics done and are helpful for ensuring software quality design with high cohesion and low coupling and metric quality design

discussed and software quality factors affecting with results [20].

Reusability is an important characteristic in Object-Oriented Software Development, that involves requirement specification, analysis, design, coding, testing and installing of existing software to rebuilt in the same procedure to create a new updated software application with reliable cost and on-time. It is a systematic procedure of creation of software applications from existing to enable more powerful features to get productivity, quality and effective performance based on the present need of the hour.

Inheritance is an important concept of reusability that is used in Object-Oriented programming for creating new classes from existing classes. Object-Oriented Software applications are benefited from inheritance-based features for real-world models. It has the capacity to solve complexity and make it reliable with inheritance metrics. An interpretation has been done with different inheritances on Object-Oriented Inheritance metrics that evaluated different computation values to explore, solve and reduce software complexity [21].

III. RESEARCH METHODOLOGY

Analytical research utilized for literature interpretation of existing object oriented metrics proposed by different researchers, and make an evaluation of the metric quality design for software growth. In analytical research, facts and information already available and analyze them to interpret results for future work. Metric design involves the generation of metric and test using programs in object oriented programming. Experimental research can be further used for evaluating results and to reach conclusions. Data sources for research are reputed research papers, reference books, review papers, thesis, object oriented software's testing tools and internet.

IV. RESULTS AND FINDINGS

After reviewing the literature, Current research is going on in the field of quality enrichment by different software development companies through expert researchers.

Object-Oriented Software Development is being used mostly for the creation of software having enhanced characteristics for real-life problems. An object is the main building block that is a combination of both data and functions. Software objects are similar to real-world entities because it contains both data and methods. Object stores its information in data and use methods to operate on this data.

A Class is a collection of objects that have common data and behavior. Object-Oriented Metrics are quality indicators and utilized to get feedback about the design of Object-Oriented software. These are used to find deficiencies and problems in design for future improvements.

Object-Oriented metrics include various inheritance metrics to enhance the reusability approach with quality measurements and computations. Software Complexity is explained as in which a design is hard to understand, long time execution and test.

Superclass has a collection of data and methods with an access modifier, that specifies the scope of any data and methods as Private, Protected and Public. Private data and methods of a base class cannot be inherited in subclass.

Protected data and methods have a scope between private and public members of a class and accessible inside a class.

Superclass is a base class that has a combination of data and functions utilized for a specific purpose. Superclass is used to create other classes according to program requirements and these subclasses have access to superclass data and functions with its own data and functions.

Superclass Subclass Attribute Access Metric is a combination of superclass and subclass attribute access values in the context of protected and the public to reach a specific result. Attributes of the class can be declared as private, protected and public access based on the requirement of program and scope of private access is within the class and protected access within classes inherited and public access can be accessed anywhere throughout the program. Superclass Subclass Attribute Access Metric is computed as Total of Protected Attribute Members + Total of Public Attribute Members

Superclass Subclass Method Access Metric is a combination of superclass and subclass functions access values in the context of protected and the public to reach a specific result. Functions or methods of a class can be declared as private, protected and public access based on the requirement of program and scope of private access is within the class and protected access within classes inherited and public access can be accessed anywhere throughout the program. Superclass Subclass Method Access Metric is computed as Total of Protected Method Members + Total of Public Method Members.

Table 1 – Object-Oriented Metric Results for Software Quality

Sr. No.	Metric Interpretation	Quality Results
1	Decision making	Metrics are helpful to software engineers to take correct decision for software development and maintenance.
2	Significant Improvement of software	Metrics makes important changes in software to update new version based on user requirements and needs.
3	Reliability	Metrics ensures reliability as error free software to customer.
4	Estimating the effort involved for maintenance	Metrics are helpful in estimating cost and feasibility of software product as well as for maintenance.
5	Quality	Metrics are used for making good quality software based on correct measurements.
6	Measure complexity of the code	Metrics ensures to measure complexity level of software to make it high cohesive and low coupled design.
7	Prediction of faults	Metrics helps to predict failures and errors to make accurate software code.
8	Process Efficiency	Metrics makes software development process efficient as per correct user requirements.
9	Product Effectiveness	Metrics ensures quality product by removing ambiguities and flaws in design.

V. CONCLUSION

Software metrics are necessary for the knowledge revolution regarding cost and effort estimations for a software product. These are explored as force multiplier due to quality improvements in existing software. Problem solving is the prime good quality of metrics for real-world entities. Software metrics is also used to refer to the discipline of using quantitative values to assist software creation,

implementation and maintenance. Object-Oriented Metrics are quality indicators and utilized to get feedback about the design of Object-Oriented software. These are used to find deficiencies and problems in design for future improvements. Object-Oriented metrics include various inheritance metrics to enhance the reusability approach with quality measurements and computations. Software Complexity is explained as in which a design is hard to understand, long time execution and test.

REFERENCES

- N.S. Gill, Software Engineering: Software Reliability, Testing and Quality Assurance, India, Khanna Book Publishing Co. New Delhi., 2016
- IEEE, IEEE Standard Glossary of Software Engineering Terminology (ANSI), Std 729-1983
- C. Ranindranath Pandian., Software Metrics, Auerbach Publications CRC, 2011
- S.H. Kan, Metrics and Models in Software Quality Engineering, Pearson Education, 2nd ed. 2016
- N. Fenton and J. Bieman, Software Metrics in Software Engg., CRC Press, Chapman Hall Book, 2014
- S.K. Misra and B. Roy, "Assessment of Object-Oriented Metrics for Software Reliability", International Journal of Engineering Research & Technology, Vol. 4, Issue 01, January 2015, pp. 432-435
- B.M. Goel and S.B. Gupta, "A Comparative Study of Static and Dynamic Object-Oriented Metrics", International Journal of Information Technology & Systems, Vol. 5, No. 1, Jan-June 2016, pp. 7-14
- S.K. Punia, P. Kumar and A. Gupta, "A Review of Software Quality Metrics for Object-Oriented Design", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 6, Issue 8, August 2016, pp. 359-367
- M.A. Job, "Analysis of Software Complexity Using Object-Oriented Design Metrics in Java Application", International Journal of Advanced Research in Computer Engineering & Technology, Vol. 5, Issue 10, October 2016, pp. 2492-2499
- N. Padhy, S. Satapathy and R. P. Singh, "Utility of an Object-Oriented Reusability Metrics and Estimation Complexity", Indian Journal of Science and Technology, Vol 10(3), January 2017, pp. 1-9
- A.A. Hussein, "Object-Oriented Software Metric Tool to Evaluate the Quality of Open Source Software", International Journal of Computer Science and Network Security, Vol.17, No.4, April 2017, pp. 345-351
- K. Maheswaran and A. Aloysius, "An Analysis of Object-Oriented Complexity Metrics", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 2, Issue 4, July-Aug 2017, pp. 768-775
- B. Kochar, S.S. Gaur and D.K. Bhardwaj, "Identification, Analysis & Empirical Validation (IAV) of Object-Oriented Design Metrics as Quality Indicators", International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 5, Issue 8, August 2017, pp. 31 – 40.
- N. Padhy, R.P. Singh, S. Satapathy and J. Sethlani, "A Systematic Literature Review of an Object-Oriented Metrics Components: Case Study for Evaluation of Reusability Criteria", International Conference on Advance Studies in Engineering and Sciences, Dec 2017, pp. 49-61
- H.B. Vincenraj, "An Optimized Cohesion Metric for Improving Quality of Object-Oriented Software Product", Ph.D. Thesis, Bharathidasan University, Tiruchirappalli, Dec 2017
- Gupta, "Software Bug Identification and Prediction Through Software Metrics in Object-Oriented Projects: An Empirical Analysis", Ph.D. Thesis, Christ University, Bengaluru, Dec 2018
- Binanto, "Measuring the Quality of Various Version an Object-Oriented Software Utilizing CK Metrics", in 2018, International Conference on Information and Communications Technology, 2018, pp 41-44
- S. Misra, A. Adewumi, L. Fernandez-Sanz and R. Damaseviciu, "A Suite of Object-Oriented Cognitive Complexity Metrics", IEEE Access, Vol. 6, January 2018, pp. 8782-8796
- N. Padhy, S. Satapathy and R.P. Singh, "State-of-the-Art Object-Oriented Metrics and Its

Reusability: A Decade Review”, Springer Nature, Smart Computing and Informatics, Smart Innovation, Systems and Technologies, 2018, pp. 431-441

20. M.K. Sharma, S. Kumar and A. Gupta, “A Comprehensive Interpretation of Object-Oriented Metrics for Quality Refinement in Software Development”, International Journal of Advanced Research in Computer Science, Vol. 9, No. 2, March 2018, pp. 390-396
21. M.K. Sharma and A. Gupta, “Interpretation of Inheritance with Object-Oriented Metrics in context of Software Complexity”, International Journal of Engineering and Advanced Technology, Vol. 8, Issue-5, June 2019, pp. 1951-1955

AUTHOR PROFILE



Mohit Kumar Sharma is a Research Scholar in Computer Science with Faculty of Computational Science, GNA University, Phagwara, India. He is working as Assistant Professor (HOD), Computer Science, J.C. D.A.V. College, Dasuya, Punjab, India. He received his M.Phil. (Computer Sc.) degree from Madurai Kamaraj University, India. He has 11 years’ teaching and 2 years’ research experience. He has 9 research paper publications in International reputed journals, 15 paper presentations in International /National conferences/ seminars and 3 Book publications. His research interest is in Software Engineering and Object-Oriented Design and Analysis.



Dr. Sandeep Ranjan is working as Assistant Professor, Faculty of Computational Science, GNA University, Phagwara, Punjab, India. He received his Ph.D. (Computer Sc. & Engg.) degree from I. K. Gujral Punjab Technical University, India. He has 12 years’ teaching experience and has published 12 research papers in International journals, conferences. He has been to the University of Tartu, Estonia as an International doctoral scholar. His research interest areas include social network sentiment analysis and Object-Oriented Programming.



Dr. Amardeep Gupta is the Principal of J.C. D.A.V. College, Dasuya, Punjab, India. He received his Ph.D. (Computer Sc. & Engg) degree from I. K. Gujral Punjab Technical University, India. He was former Head of the Department, Post Graduate Department of Computer Science, DAV College, Amritsar, Punjab. He has 21 years’ teaching and 6 years’ research experience. He has more than 35 research paper publications in International journals, conferences and 12 Book publications. His research interest is in Parallel Processing and Object-Oriented Programming.