INHERITANCE METRICS FOR QUALITY ENRICHMENT IN CONTEXT OF OBJECT-ORIENTED SOFTWARE

Mohit Kumar Sharma[1,2]
[1]Research Scholar, Computer Science
FCS, GNA University, Phagwara
[2]Assistant Professor (HOD)
P.G. Deptt. of Computer Science
J.C. D.A.V. College, Dasuya, Punjab, India

Dr. Shailesh Kumar
Associate Professor
Computer Science
Faculty of Computational Science (FCS)
GNA University, Phagwara, India

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

ABSTRACT

Object-Oriented programming is a vital concept of the real-world entities and their features creation. Objects have their own internal data structure which defines their data and functions. In this competitive world of IT, different companies producing software products with minimum cost and maximum profit and timely availability in market. Software Quality is a mechanism that evaluates, assesses, and improves the accomplishment of software. Object-Oriented software applications are benefited with inheritance feature based on real world models. It has a capacity to solve complexity and make it reliable with the help of inheritance metrics. Inheritance metrics are part of Object-Oriented metrics for quality enrichment given by different researchers from time to time in past.

Keywords- Software quality; Object-Oriented metrics; Inheritance metrics.

I. INTRODUCTION

Software quality is a mechanism that evaluates, assesses, and improves the accomplishment of software. Software quality is explaining as the degree to which software meets requirements for reliability, maintainability, portability as contrasted with functional, performance, and interface requirements that are satisfied as a result of software engineering [1].

In the past decade, many IT companies have started to deploy Object-Oriented technology in their software development efforts. Object-Oriented Software Development is concept of the real-world entities and their features. Objects have their own internal data structure which defines their data and functions. Object-Oriented Design restrained all the properties and worth of software that is associated to any large or small project.

Object-Oriented metric is a measurement term of a degree to which a software system holds some characteristics. These metrics are based on actual project experiences; these are not law of nature. These are guidelines that give indication of the progress that a project has made and the quality of the design [2]. Moving from measurement to metrics is like moving from observation to understanding. Metrics are conceived by the user and designed to reveal a chosen characteristic in a reliable and meaningful manner [3]. Object-Oriented Metrics concentrate on measurement that can be applied to the class and the design characteristics as localization, encapsulation, inheritance, information hiding, polymorphism, messaging and object abstraction [4].

There is need for Object-Oriented Metrics due to visibility, planning and control, quality, productivity [5]. Software quality and reliability describes as fault-free software operation for a specified period of time [6].

II. RELATED WORK

Six Object-Oriented software metrics and these are the intellectual research for Object-Oriented software development. These metrics are utilized with a theoretical idea in measurement of objects, and which incorporate the experiences for software managers and evaluation done by these metrics against criteria for credibility and existing empirical data from commercial software to explain the features of metrics on real world applications and recommends the process in which these metrics may be utilized [7]. A set of Object-Oriented reuse metrics have been proposed at class level namely Method Reuse Factor, Attribute Reuse Factor, Method Reuse Factor with Inheritance and Attribute Reuse Factor with Inheritance [8].

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 [9].

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 [10].

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 [11].

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 behaviour of software [12].

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 [13].

Object-Oriented applications require 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 [14].

III. OBJECTIVE OF THE STUDY

To analyze the different inheritance metrics for quality enrichment in context of Object-Oriented software and explore results with validations.

IV. RESEARCH METHODOLOGY

In analytical research, evidences and information previously accessible and investigate them to interpret results for future work. Experimental research is also used and to reinterpret existing ones with results. Six software designs based on various inheritances of Object-Oriented languages are analyzed as data-set as a measurement tool to understand the relation with metrics to explore the significance of measurements for quality design using statically. Primary data is taken by six inheritance projects and secondary data for research are from reputed research papers, reference books, review papers, thesis, Object-Oriented software's testing tools and internet.

V. INHERITANCE METRICS

Reusability is important powerful reason for the use of inheritance in Object-Oriented software applications. Inheritance metrics are part of Object-Oriented metrics given by different researchers from time to time. Object-Oriented Metrics concentrate on measurement that can be applied to the class and the design characteristics as Localization, Encapsulation, Inheritance, Information hiding, Polymorphism, Messaging and Object abstraction. Moving from computation of metrics is like moving from observation to understanding. Inheritance metrics are conceived by the software engineer and designed to reveal a chosen characteristic in a reliable and meaningful manner for future use. Object-Oriented Metrics suggested by Shyam R. Chidamber and Chris F. Kemerer is the intellectual research for Object-Oriented software. Depth of Inheritance Tree and Number of Children are inheritance metrics also referred as C-K metrics [7].

Metrics for Object-Oriented Design (MOOD) explored a basic structural procedure of the Object-Oriented attributes as encapsulation, inheritance, polymorphisms, message passing. Method Inheritance Factor and Attribute Inheritance Factor are inheritance metrics [15].

Inheritance metrics to find assessment and measurement of Object-Oriented software quality. Attribute Count software metric is computation of total number of class attributes including both inherited class attributes and the

attributes defined in the class are computed. Method Count software metric is computation of total number of class methods including both inherited class methods and the methods defined in the class are computed. Number of Operation Overridden by subclass software metric is computation of total number of subclass operations overridden [16].

Inheritance metrics are part of Object-Oriented metrics for quality enrichment given by different researchers from time to time in past.

VI. STATISTICAL FINDINGS AND RESULTS

Inheritance have mainly five types – single, multiple hierarchical, multilevel and hybrid inheritance. Student Management System is a design of student-related information for fee collection assistance utilized in educational bodies. Six software designs based on various inheritances of Object-Oriented languages are analyzed as data-set as a measurement tool to understand the relation with metrics to explore the significance of measurements for quality design.

Interpretation of inheritance-based projects has been done for improved quality outcomes. Correlation between attributes, methods and complexity level is explored in results. If the number of inherited attributes increased without modular design, then the complexity level will be increased in design. Metric values explored the complexity level of structure and assist software engineers to build effective design. These project designs are taken into a research lab for computing metric values and results are as under: -

Table 1 - Metric Results

Projects	Inheritance Category	AIF	MIF	AC	MC	NOO
Project I	Without Inheritance	0	0	19	17	0
Project II	Single	0.38	0.40	19	18	1
Project III	Multiple	0.43	0.40	23	27	0.66
Project IV	Hierarchical	0.49	0.47	23	27	0.33
Project V	Multilevel	0.56	0.51	21	15	0.66
Project VI	Hybrid	0.60	0.58	27	23	0.50

Descriptive statistics interpretation of metric values for AIF, MIF, AC, MC, and NOO metrics have been calculated by statistics tool having a minimum, maximum, mean and standard deviation computed as under: -

Table 2 – Statistics Results

	N	Minimum	Maximum	Mean	Std. Deviation
AIF	6	.00	.60	.4100	.21652
MIF	6	.00	.58	.3933	.20452
AC	6	19.00	27.00	22.0000	3.03315
MC	6	15.00	27.00	21.1667	5.23132
NOO	6	.00	1.00	.5250	.33951
Valid N (listwise)	6				

Descriptive statistics are significant measuring outcomes, which assist to find the data is better or not and find the idea of an overall distribution of the observations in the data-set. Minimum denotes the bottommost value against each metric and maximum signifies the uppermost value against each metric. Mean signifies average value and standard deviation finds the typical distance between each metric data point and mean. Standard deviation is found to be the most vital index of variability against each metric of related inheritance-based project.

VII. CONCLUSION

In this competitive world of IT, different companies producing software products with minimum cost with maximum profit and timely availability in market. Software Quality is a mechanism that evaluates, assesses, and improves the accomplishment of software. Object-Oriented software applications are benefited with inheritance feature based on real world models. It has a capacity to solve complexity and make it reliable with the help of inheritance metrics. It also increases quality of software by measuring different components of Object-Oriented

software. Inheritance metrics are part of Object-Oriented metrics given by different researchers from time to time.

REFERENCES

[1] Anirban Basu, *Software Quality Assurance and Testing*, Prentice Hall of India, 2015

[2] Mark Lorenz and Jeff Kidd, *Object-Oriented Software Metrics*, Pearson Education, 1994

[3] C. Ranindranath Pandian., *Software Metrics*, Auerbach Publications CRC, 2011

[4] Stephen H. Kan, *Metrics and Models in Software Quality Engineering*, Pearson Education, 2nd ed. 2016

[5] Norman Fenton and James Bieman, *Software Metrics in Software Engg.*, CRC Press, Chapman Hall Book, 2014

[6] Glenford J. Myers, *Software Reliability Principles*, John Wiley & Sons, 2002

[7] Shyam R. Chidamber and Chris F. Kemerer, "A Metrics Suite for Object-Oriented Design", *IEEE Transactions for Software Engineering*, Vol. 20 No. 6, pp. 476-493, June 1994

[8] N. Sambasiva Rao, "Software Reuse Metrics for Object-Oriented Systems", *Ph.D. Thesis*, Anna University, Chennai, July 2007

[9] 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

[10] 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

[11] 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

[12] 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

[13] 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

[14] 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.

[15] K.P. Srinivasan and T. Devi, "A Comprehensive Review and Analysis on Object-Oriented Software Metrics in Software Measurement", *International Journal on Computer Science and Engineering*, ISSN: 0975-3397 Vol. 6 No.07 Jul 2014

[16] Mark Lorenz and Jeff Kidd, *Object-Oriented Software Metrics*, Pearson Education, 1994

ABOUT AUTHORS

Mohit Kumar Sharma is a Research Scholar of Computer Science with Faculty of Computational Science, GNA University, Phagwara, India. He is working as Assistant Professor (HOD), Post Graduate Department of 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 10 years' teaching and 2 years' research experience. He has 5 research paper publications in international journals, conferences and 3 book publications. His research interest is in Software Engineering and Object-Oriented Design and Analysis.

Dr. Shailesh Kumar is an Associate Professor of Computer Science with the Faculty of Computational Science, GNA University, Phagwara, India. He received his Ph.D. (Computer Sc.) degree from Banasthali Vidyapith, India. He has 16 years' teaching and 6 years' research experience. He has more than 25 research paper publications in international journals and conferences. His research interest is in Database Management, Data warehouse and Data Mining, Knowledge Management Systems, Software Quality Engineering and Object-Oriented Design.

Dr. Amardeep Gupta is the Principal of J.C. D.A.V. College, Dasuya, India. He received his Ph.D. (Computer Sc.) 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 20 years' teaching and 6 years' research experience. He has more than 32 research paper publications in international journals, conferences and 12 book publications. His research interest is in Parallel Processing and Object-Oriented Programming.