INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

An Analysis of Metric-based Quality Design for Software Development

Mohit Kumar Sharma[1,2]*, Amardeep Gupta[3], Neeraj Puri[4]

1 Research Scholar, Computer Science, Faculty of Computational Sciences, GNA University, Phagwara
2 Assistant Professor (HOD), Computer Science, J.C. D.A.V. College, Dasuya, Hoshiarpur, 144205, Punjab, India
3 Principal, J.C. D.A.V. College, Dasuya, Hoshiarpur, 144205, Punjab, India
4 Assistant Dean Research, GNA University, Phagwara, 144401, Punjab, India

OPEN ACCESS

Received: 27.06.2020
Accepted: 19.09.2020
Published: 29.09.2020

Editor: Dr. Natarajan Gajendran

Citation: Sharma MK, Gupta A, Puri N (2020) An Analysis of Metric-based Quality Design for Software Development. Indian Journal of Science and Technology 13(35): 3600-3604. https://doi.org/10.17485/IJST/v13i35.1013

***Corresponding author**.

prof_mks@yahoo.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Background/Objectives: The main purpose is to find the significance of Object-Oriented metrics on software design; metrics are related to minimizing class unused attributes and methods to maintain quality and complexity. **Methods/Statistical analysis:** Experimental analysis is utilized for validating findings. Mathematical properties are used for validation and evaluation outcomes. The purpose of experimental validation is to understand whether a measurement quantifies what it should quantify. MS-Excel is utilized as a metric computation and graph generation for fifteen project designs. Statistical analysis through SPSS tool of different inheritance categories as single, multiple, multilevel, hierarchical, and hybrid are utilized for metric results. **Findings:** SSAAM is suggested to give measurement results for counting public and protected attributes of all super and subclasses in the class inheritance hierarchy. SSMAM explores numeric value for computing public and protected methods of all super and sub classes in the class inheritance hierarchy. Higher results are computed of SSAAM and SSMAM, it shows more inheritance hierarchy utilized in the design. It assist in a controlled use of public/protected members in software design to minimize complications. Software design complications can be enhanced due to more use of protected and public members as well as decreased use of the concept of encapsulation.

Keywords: Metrics; quality; protected and public specifiers

1 Introduction

The software development process is an idea of programming building to investigate a technique for necessity gathering, structure, advancement, testing, and execution of user requirements prerequisites to create programming. Various exercises attempted for the examination of programming, requirement gathering, cost estimation, and user request approval for programming advancement [(1)].

Object-Oriented Programming is a high-level language with English instructions that are easy to remember. It has significant highlights for protection as

information hiding[2]. Data and functions are joined together to form an entity and make it private for data covering up. Unauthorized cannot able to access that information[3].

Software measurement assumes a significant role in keeping up the nature of the quality of applications. It is the procedure in which numbers or symbols are allocated to attributes of real-world entities as per set conventions[4].

Software metrics are significant because of numerous reasons including estimating programming execution, arranging, and estimating the profitability of programming application. It is an estimation term for an application that has characteristics[5].

Object-Oriented metrics are an estimation procedure of product metrics in which computation is done on real-world entities to depict them as indicated by plainly characterized rules. These metrics encourage programming specialists to discover the profitability of the product application[6].

Software quality is an indispensable idea of software designing for ensuring and keeping up the standard level for user's acceptance worldwide through different institutionalization bodies such as ISO, IEEE, ANSI, etc[7]. Software quality in Object-Oriented software satisfies user necessities with guideline level. Productivity deals with quality characteristics, reliable, correct, user-friendly, reusability, portability, and efficient programming. Reliability ensures error-free programming.

Object-Oriented metrics have been used to give expected contribution to structure the models for estimating viability quality factor. Object-Oriented metrics are critical to discovering visibility, arranging, control, quality, and efficiency of programming advancement[8]. Object-Oriented metrics applied to examine the coding of Object-Oriented programming as a quality pointer.

Object-Oriented measurements by Chidamber and Kemerer is the scholarly research for Object-Oriented programming referred to as C-K metrics[9]. MOOD investigated by F.B. e Abreu and investigated a fundamental structural methodology considered object-based properties as hiding, reuse, polymorphism, and message passing. MOOD incorporates six unique measurements utilized for programming estimation and these measurements work at a system level, giving an overall performance of an application[10].

Object-Oriented measurements to discover appraisal and estimation of Object-Oriented programming quality. These Object-Oriented measurements are arranged into three as to size, inheritance, internal and external based measurements[11]. Several Object-Oriented metrics have been suggested by different scholars from time to time to enhance quality in past decades.

2 Materials and Methods

Research in metrics is a specialty of logical examination and search of information for appropriate data on a particular measurement. Experimental tools are utilized for the conduction of research with a set of rules. These are strategies or techniques utilized by specialists use in performing research activities.

2.1 Experimental analysis

Experimental analysis is utilized for validating findings. Mathematical properties are used for validation and evaluation outcomes. Experiential validation of the metric complements the theoretical validation. The purpose of experimental validation is to see whether a measurement quantifies what it should quantify. MS-Excel is utilized as a tool for suggested metric computation for fifteen project designs.

2.2 Statistical analysis

Fifteen project designs of Student Management System (SMS), Banking Management System (BMS), and Library Management System (LMS) in single, multiple, hierarchical, multilevel, and hybrid inheritance of object-oriented languages (C++/C#, etc.) are utilized as data-set for statistical analysis. SPSS tool is utilized for statistical analysis of different inheritance categories as single, multiple, multilevel, hierarchical**,** and hybrid for metric results.

3 Results and Discussion

The motivation of utilizing Superclass Subclass Attribute Access Metric (SSAAM) and Superclass Subclass Method Access Metric (SSMAM) is to find measurement values to maintain quality in projects. SSAAM gives measurement results for counting public and protected attributes of all super and subclasses of different inheritance categories as single, multiple, multilevel, hierarchical**,** and hybrid for metric results. SSMAM explores numeric value for computing public and protected methods of all super and subclasses of different inheritance categories as single, multiple, multilevel, hierarchical**,** and hybrid for metric results.

The evaluation and validation of metric-based quality design is experimentally done with fifteen project designs of Student Management System (SMS), Banking Management System (BMS) and Library Management System (LMS) in single,

multiple, hierarchical, multilevel and hybrid inheritance of object-oriented languages (C++/C#, etc.) are analyzed as data-set to understand the relation between different inheritance categories.

Student Management System (SMS) is a design of student-related information for fee collection assistance utilized in educational bodies. Banking Management System (BMS)is a design of bank-related information for customer, account, and loan-related assistance utilized in banks either private or public. Library Management System (LMS) is a design of library-related schema for library automation and record-related assistance utilized in libraries of any institution.

SSAAM gives measurement results for counting public and protected attributes of all super and subclasses in the class inheritance hierarchy and SSMAM explores numeric value for computing public and protected methods of all super and subclasses in the class inheritance hierarchy.

Fig 1. SSAAM and SSMAM

Statistical analysis of different inheritance categories as single, multiple, multilevel, hierarchical, and hybrid are utilized in metrics as SSAAM and SSMAM. Student Management System, Banking Management System, and Library Management System considered for a comparative study of fifteen project designs.

Table 1. Statistics analysis of inheritance based projects

Inheritance		Count	Min.	Max.	Mean	Median	Mode	Std. Deviation
Single	SSAAM	3	9	15	12	12	9	3
	SSMAM	3	7	17	12	13	7	5
Multiple	SSAAM	3	12	20	16	15	12	4
	SSMAM	3	11	27	18	16	11	8
Multilevel	SSAAM	3	10	22	14	11	10	7
	SSMAM	3	10	15	12	11	10	3
Hybrid	SSAAM	3	13	21	18	19	13	4
	SSMAM	3	12	23	17	17	12	6
Hierarchical	SSAAM	3	13	19	16	17	13	3
	SSMAM	3	10	25	16	13	10	8

Table 2. Comparative study using descriptive statistics

Sr. No.	Projects	Metric	N	Minimum	Maximum	Mean	Std. Deviation
1.	Student Management System	SSAAM	5	10	17	13.40	2.702
		SSMAM	5	15	27	21.40	5.177
2.	Banking Management System	SSAAM	5	9	19	12.80	3.768
		SSMAM	5	10	17	13.80	2.775
3.	Library Management System	SSAAM	5	15	22	19.40	2.702
		SSMAM	5	7	12	10.20	1.924

Pearson Correlation Coefficient between Superclass Subclass Attribute Access Metric (SSAAM) and Superclass Subclass Method Access Metric (SSMAM) of Student Management System, Banking Management System, and Library Management System are considered for comparative study.

Student Management System is calculated as the value of r is .879 and it is significant at the 5% level as .049. There is a correlation between excessive use of protected and public data with the protected and public methods in SMS designs in a class inheritance hierarchy.

Banking Management System is calculated as the value of r is .665 and it is not significant at the 5% level as .221. There is no correlation between excessive use of protected and public data with the protected and public methods in BMS designs in a class inheritance hierarchy.

Library Management System is calculated as the value of r is .943 and it is significant at the 5% level as .016. There is a correlation between excessive use of protected and public data with the protected and public methods in LMS designs in a class inheritance hierarchy.

Table 3. Comparative study using Pearson correlation

Sr. No.	Projects	N	Pearson Correlation	Sig. (2-tailed)	Statistically Significant < 0.05
1.	Student Management System	5	.879	.049	Yes
2.	Banking Management System	5	.665	.221	No
3.	Library Management System	5	.943	.016	Yes

A comparative study of various object-oriented metrics for fifteen project designs of Student Management System, Banking Management System and Library Management System are computed in Table 3.

Graphical representation for means of inheritance categories is shown in Figure 2.

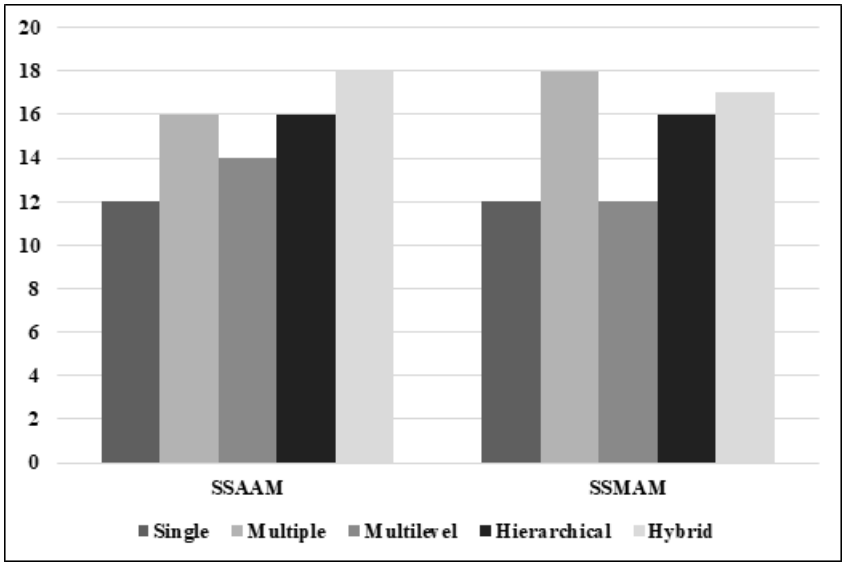

Fig 2. Means of Inheritance Categories

4 Conclusion

The fundamental target of metrics is to quantify and improve software size**,** and reduce complications. Inheritance measurements are excellent pointers with utilization to input in the proposal of software creation. These are utilized to discover metric-based quality for design and issues in a plan for further enhancements. SSAAM gives measurement results for counting public and protected attributes of all super and subclasses in the class inheritance hierarchy. SSMAM explores numeric value for computing public and protected methods of all super and subclasses in the class inheritance hierarchy. Metric-based quality design is interpreted for reliability with quality to deliver standard software. A productive measurement tool with a decision support system can be benefited based on this design for object-oriented software measurement and quality results. Metric based quality design can be utilized for quality enrichment in object-oriented software development. Superclass Subclass Attribute Access Metric (SSAAM) and Superclass Subclass Method Access Metric (SSMAM) are utilized to support this design with required attributes as well as methods in a class inheritance hierarchy.

Conflict of interest

The authors declare that they have no conflicts of interest. The authors alone are responsible for the content and writing of the paper. No Financial Aid or Grant is received from any Institution for conducting this study.

References

1) Akbar R. Tailoring Agile-Based Software Development Processes. *IEEE Access*. 2019;7:139852–139869. Available from: https://doi.org/10.1109/ACCESS.2019.2944122.
2) Gosain A, Sharma G. Object-oriented dynamic complexity measures for software understandability. *Innovations in Systems and Software Engineering*. 2017;13(2-3):177–190. Available from: https://dx.doi.org/10.1007/s11334-017-0304-3.
3) Ponnala R, Reddy CRK. Object-Oriented Dynamic Metrics in Software Development: A Literature Review. *International Journal of Applied Engineering Research*. 2019;14(22):4161–4172. Available from: https://www.ripublication.com/ijaer19/ijaerv14n22_10.pdf.
4) Kaur H, Verma GN. Software Complexity Measurement: A Critical Review. *International Journal of Engineering and Applied Computer Science*. 2016;1(1):12–16. Available from: https://doi.org/10.24032/ijeacs/0101/03.
5) Punia SK, Kumar P, Gupta A. A Review of Software Quality Metrics for Object-Oriented Design. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2016;6(8):359–367. Available from: www.ijarcsse.com.
6) Zighed N, Bounour N, Seriai AD. Comparative Analysis of Object-Oriented Software Maintainability Prediction Models. *Foundations of Computing and Decision Sciences*. 2018;43:359–374. Available from: https://dx.doi.org/10.1515/fcds-2018-0018.
7) Fernández SM. Continuously Assessing and Improving Software Quality With Software Analytics Tools: A Case Study. *IEEE Access*. 2019;7:68219–68239. Available from: https://doi.org/10.1109/ACCESS.2019.2917403.
8) Maheswaran K, Aloysius A. An Analysis of Object-Oriented Complexity Metrics. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 2017;2(4):768–775. Available from: https://doi.org/10.32628/CSEIT11724107.
9) Sharma RK, Gandhi P. Study of Reliability of Object-Oriented Structure Consuming CK Metrics. In: 6th International Conference on Computing for Sustainable Global Development, India. 2019;p. 828–831. Available from: https://ieeexplore.ieee.org/document/8991174.
10) Dhillon PK, Dhand P. Empirical validation of MOOD metrics to predict Software Reuse. *International Journal of Emerging Trends & Technology in Computer Science*. 2017;6(4):126–132. Available from: https://www.ijettcs.org/Volume6Issue4/IJETTCS-2017-07-25-31.pdf.
11) Kumar L, Rath SK. Empirical validation for effectiveness of fault prediction technique based on cost analysis framework. *International Journal of System Assurance Engineering and Management*. 2017;8:1055–1068. Available from: https://doi.org/10.1007/s13198-016-0566-4.