

Empirical Analysis of Metrics Using UML Class Diagram

Bhawana Mathur
Deptt. of CS&E
JECRC University Jaipur

Manju Kaushik
Deptt. of CS&E
JECRC University Jaipur

Abstract—Lots of Organizations before they are setup survey the maintainability of programming frameworks. To give quality program design there exists a critical strategy called Object-Oriented Framework. Object-Oriented estimations might be utilized to study the judgment skills suite of a class diagram's structure particularly programming valuations and how the models have been developed and portrayed. The UML Class Diagram metrics maintain the Object-Oriented software. It is maintained through the investigation of the association among object oriented metrics and maintainability. This paper shows the effects of a scientific evaluation of software maintainability forecast and metrics. The research aims at the software quality attribute of maintainability as opposite to the method of software maintenance. It also aims to find out the vital correlation between structural complexity metrics and maintenance time. Several investigators have done copious in it, got lots of theoretical outcomes, and subsequently established a chain of practical uses. Due to dynamic changes in object-oriented technology, in today's scenario the class diagram is an essential UML model, as, researcher must first get to know the use of software in a scientific manner. It is an affordable strategy which has had an exceptional result in recent times. This paper is related to UML class diagram metrics through which a way is provided to maintain UML class diagram complexity weights. UML Class diagram's qualities will efficiently and technically show the complexity of Object -Oriented Software. A more specific research study has shown that the technique is associated with individual's experience and also can be useful to improve software quality.

Keywords—UML Class diagram; Maintainability; Object Oriented System; CK Metrics suite; Model; Software; UML

I. INTRODUCTION

The aim of the research is to find out the significant correlation between structure complexity during maintenance time and the role played by metrics in class diagrams for maintainability. In the arena of software engineering, the aim is to lessen the price and energy in developing a software which has improved features but low maintainability. Because of this, software maintainability of a code is a difficult task which each user is facing. The class diagram plays a significant role in software maintenance. An upgraded version of a software will help in solving various defects and problems and the users will become accustomed to the newer environment. With reference to the ISO/IEC-9126 maintainability, the capacity of a software should be changed as per the requirement [1]. The changes will be remedial, adaptive or perfective in nature and the software systems will purposefully fulfill the requirements of a user. The present

requirement of a user is to acquire better features with low maintainability which poses a big challenge for researchers. Maintenance usually absorbs 40% to 80% of a software system's value. Subsequently, software maintenance is essential; it is significant for the designers to think about the maintainability side throughout the process of code changing. Failing to satisfy users in this regard can lead to an increase in their maintainability value. That is why software system metrics have become vital in the field of software engineering practice. Researchers are more inclined towards making quantitative software traits that would evaluate the labor cost, cost efforts, energy, etc.

Regardless of the problems that may come on the way, evaluating maintainability of a code continues to be an awfully troublesome and difficult task. In any case, some of the developers or users are of the view that practical valuation may appear to be difficult in software development process. As a result of the many necessary conclusions along with investigation associated with software metrics as well as maintainability, it is vital to ascertain various associations among object-oriented metrics and their maintainability. Chidamber and Kemerer (CK) object-oriented metrics is incontestable with its durable influence on programming feature and attempts a supportive outfit of item arranged software metrics [2].

II. OBJECTIVES

- A. To use various metrics in UML class diagram to find out maintainability.
- B. To demonstrate a controlled analysis so as to assess if there is a clear imprint in maintaining the integrity of the specifications.
- C. To use the experiential data as structure models which portrays UML class diagram maintainability
- D. To take into account the particular models to anticipate UML class diagram maintainability in the OOIS advancement life-cycle.

III. LITERATURE REVIEW

Varied kinds of maintainability models have been prescribed. Prototypes have expected maintainability with code as well as plan estimations; however certain are concentrating just on layout level estimations [3]. Nowadays, the principle points for programming building are to expand the properties of programming ancient rarities. Nearby is a

conjoint contract that the quality assertion are crucially assured since the beginning periods of the product development life cycle concentrating on high-level design articles like class diagrams. In the improvement of Object Oriented Information System, the class diagram is a significant initial object that sets the basis of completely imminent model and application task. Later, class diagram attribute is a vital problem that is required to be assessed (better-quality if essential) in form to get attributes OOIS, that is the prime thing of current era programming augmentation organizations. The underlying accentuation on class diagram quality is permitted with the help of IS innovators for enhanced OOIS, denied of inessential modification at future periods of the improvement when changes are further sumptuous and further hard to accomplish. In this field where software analysis is a significant part since the initial obtainability of metrics provides a new assessment to class diagram attributes and unbiased approach staying away from inclination in the quality assessment process. Besides, measurements convey an esteemed and fair-minded vision into accurate procedures of enhancing the greater part of the product qualities. Shown that maintenance was (then will remain) the key supply customer of the entire programming life cycle, common sense has made one of the item vestige qualities highlights that item change affiliations further focus on pretty much. Maintainability is not limited to code; it is nature of the divergent programming. However, researchers are cognizant that maintainability is an outside trademark that must be considered in the OOIS circuitry. In this way, it is essential to have starting indications of such standards based on the physical attributes of class outlines. Most advanced of the obtainable works on OO methods is associated with methods which can first be real-world once a software artifact is finished or approximately finished. The above- mentioned data also makes the framework in OOIS. Later full examination of certain diagrams like reachable OO quantum, fitting to class diagrams at strong configuration stage researchers have future set up of UML class diagram structural complexity that helped over through viable methodology about UML relations, such as associations, generalizations, aggregations and dependencies[4]. However, use of metrics is of no worth if their applied procedure is not validated analytically, whichever in tricks of case studies occupied against tangible task beyond meticulous research. Observed justification is vitally aimed at achievement of some software evaluation task. During this research, class diagram is additionally perceived in experiential exploration, certainly attained initial signs of class diagram maintainability are obtained [4]. The results have shown good initial conclusions for the OOIS researchers to proceed in the OOIS improvement circuitry subsidizing to the ascent of well component OOIS. A preceding skillful research has been achieved. The autonomous variables are present in the UML class diagram with structure complexity metrics. Prior examinations show that there are indigent variables of practicality particular understandability and modifiability using client appraisals gathered on seven semantic marks scale. In object oriented

software, to investigate measurement like structure complexity metrics in UML ,Class diagrams are used to collect observational data in features like union and coupling [4] [5].

IV. BASIC CONCEPT

There dependably has been an interest to give proficient, compelling and amazing programming. There are numerous maintainability items to convey enhanced maintainability. The component of respectable programming arranges seriously strong element of programming. Finally, some finishing up comments and future patterns in measurements for OO models are displayed [3]. Quite recently specific estimations have been described, which can be associated with class diagram level when all is said and done (see Table 4). Additionally customary eminent like, the quantity of classes, quantity of properties and quantity of techniques are considered.

V. EMPIRICAL DATA COLLECTION

Information sources are exhibited and given a point by point portrayal of information accumulation technique through inspection of 22 distinctive UML Class diagrams.

Following steps have been followed:

- A. The self-representing variables are the vital whim and the extent of UML class diagrams, measured concluded the 11 estimations showed up in Table I.
- B. Individually, choose an inside of a subject class diagram test, i.e., each of the tests (trial undertakings) must be understood by all of the subjects. The trials are to be found as an alternate request used for every theme.
- C. Themes are assumed a careful instructional meeting before the analysis occurred. Be that as it may, the subjects didn't know about what highlights are imagined to ponder. Nor are they mindful of the genuine speculation expressed.
- D. So as an association between the measurements presented in Table 1 and the class diagram maintainability is clear.

VI. CASE STUDY

Twenty two different UML class diagrams have been examined and maintainability through these Size metrics and structure complexity metrics have been found. Size metrics contains Response for a class, NOA, Number of Method, Total Number of methods, Weight methods per class. Structure Complexity metrics contains Number of Children, Number of classes, Number of Relation, NGen, MaxDIT, NAggH, NGenH [6][7].

VII. SIGNIFICANCE OF STUDY

Particularly measurements ratio the structure unpredictability of UML Class diagrams as a result of the use of associations, for instance, affiliations, speculations, conglomerations and conditions [3]. In like manner standard measurements, for instance, the number of classes and the amount of characteristics et cetera are considered (See Table 1).

VIII. THESE CLASS DIAGRAM STRUCTURAL COMPLEXITY MEASURES PERMIT OO CREATORS

- A. Numerical examination of outline choices, thus a target choice among various class diagram changes with proportional semantic substance.
- B. The forecast of outer quality attributes, similar to maintainability in the starting periods of the IS period of existence is also an asset to assignment taking into account these expectations.

IX. OBSERVATIONS

- A. Analyze: UML class diagram structural complexity metrics
- B. Determination of Assessing: Concerning their capacity of imperativeness castoff as class diagram maintainability markers.
- C. As of the perspective of Information system draftsmen: Under the guidance of Research Scholar Computer Science and Associate Professor of the Computer Science in the JECRC University.
- D. Arranging. – Background choice. The examination is not permanent and this one highlights on UML class diagram structural complexity metrics. The maintainability is improved since the particular setting is also progressed underneath one researcher [8] [9]. The Investigation reports are the genuine issue, particularly thae pointers can be utilized to measure the maintainability of class diagram. In the end, it inspects the connection between metrics and maintainability.
The components remain in UML class diagrams. The independent variable is controlled by the valuations. The destitute variable stays restricted all the times.
- E. Variables determination. The free fickle in the UML class diagram is essentially unpredictable. The maintainability is called penniless fickle in UML class diagram.
- F. A researcher called time as “maintenance time.” Maintenance time incorporates an immaculate opportunity to get a hold on the class diagram so as to examine the compulsory variations and to execute them. The doubt here is that, for the similar adjustment assignment, the faster a class diagram can be changed, the less troublesome it will be to keep up.

- G. Hypothesis formulation. Researcher desires to trial the subsequent hypotheses:
- H. The Null hypothesis, H0: There is no significant correlation between structural complexity metrics like RFC, NOA, NOM, WMC, NOC, NC, NOR, NGen, MaxDIT, Nagg, NAggH, NGenH and maintenance time.
- I. An Alternative hypothesis, H1: There is a significant correlation among structural complexity metrics (RFC, NOA, NOM, WMC, NOC, NC, NOR, NGen, MaxDIT, Nagg, NAggH, NGenH) and maintenance time.

TABLE I. MEASUREMENTS FOR UML CLASS DIAGRAM BASIC INTRICACY

Metric Name	Metric definition
Response for a Class (RFC)	Response for a Class(RFC)
Number of Attributes (NOA)	The Number of Attributes (NOA)
Total Number of Methods (NOM)	The Total Number of Methods (NOM)
Weight Method per Class (WMC)	The Weight Method per Class (WMC)
Number of Children (NOC)	The Number of Children (NOC)
Number of Classes (NC)	The Number of Classes (NC)
Number of Relation(NOR)	The Number of Relation(NOR)
Number of Generalizations (NGen)	The total number of Generalization relationships within a class diagram (each parent-child pair in a generalization relationship).
Maximum DIT(MaxDIT)	It is the maximum of the DIT (Depth of Inheritance Tree) values obtained for each class of the class diagram. The DIT value for a class within a Generalization Hierarchy is the longest path from the class to the root of the hierarchy.
A Number of Aggregations Hierarchies (NAggH)	The total number of Aggregation hierarchies (whole-part structures) within a class diagram.
A Number of Generalisations Hierarchies (NGenH)	The total number of Generalization hierarchies within a class diagram.

- J. Experiment design. An inside of subject configuration test, i.e., completely the tests (trial undertakings) must be determined by all of the themes. The point indicates the trials in a various request.

X. OPERATION

Preparation. : While the examinations stayed finished, the subjects had involved two methods for Software Engineering. Twenty-two UML class outlines of exceptional use that were absolutely not hard to be appreciated by each of the topics. The charts have various basic multifaceted nature, all things considered of metric qualities. Each and every graph had an encased scan that integrated a quick depiction of what the outline meant and two new presents for the category chart. Every last zone anticipated that it would adjust the class diagrams permitting the novel necessities and demonstrate the begining and end time. The separation among the two is anything we call time (passed on in minutes and seconds). The

alterations based on every classification graph have been similar in qualities, procedures, classes, et cetera.

Execution: The topics had been common and revealed figure out by what method to do the checks. The major part of the learning with the changed class diagrams is composed with the maintenance time acquired after the reactions of the checks and the estimations prices without a doubt figured by the process for a metric instrument orchestrated.

Information Validation: When the data was collected, it was assessed if the checks have been done and if the movements had been finished precisely. The expert used the information gathered as a touch of asking for to test the hypotheses definite. (see Table II) [4] [7].

Analyzing the Spearman's correlation coefficients: There is a high association (rejecting hypothesis H0) among most of the UML class diagram's metrics and maintenance time is surmised. The way that every one of the metrics has an association more prominent than 0.5 is expected, which is a typical edge to survey correlation values. NOR is the emerge that has a lesser relationship, yet this could be cleared up by the route that in most of the class diagrams NOR appropriated the value 0 (see Table II).

TABLE II. MAINTENANCE TIME BETWEEN METRICS

	RFC	NOA	NOM	WMC	NOC	NC	NOR	NGen	MaxDIT	NAggH	NGenH
Maintenance	0.56	0.75	0.83	0.56	0.54	0.89	0.41	0.58	0.72	0.68	0.69

XI. VALIDITY EVALUATION

The few problems that undermine the validity of the experiential gain of knowledge are pondered [4] :

A. Threats to inference validity

The inference legitimacy describes the level to which inferences are quantifiably immense. The fundamental constraint that would affect the quantifiable legitimacy of this learn is the degree of the case know-how (242 qualities and 22 subjects), that potentially are not suitable for each parametric and non-parametric size output [4] [10].

B. Threats to develop Validity

The dependent variable castoff as maintenance time, i.e., the time every subject spent implementation.

C. Threats to inside Validity

The inward legitimacy portrays the level of trust in an intention impact relationship amongst additional items of leisure activity and the experiential result.

The running with issues have been distributed with contrasts between subjects [9]:

In this examination, in building UML class diagrams. This is shown through the examination of the descriptive statistics in light of the aggregate maintenance time for every subject (see Table III). As the Kurtosis qualities are more prominent than zero and conclude that there are no great contrasts between in the investigation.

TABLE III. DESCRIPTIVE STATISTICS FOR THE TOTAL MAINTENANCE TIME

	Min.	Max	Mean	Std. Deviation	Skewness	Kurtosis
Total maintenance time (minutes)	35	45	40.9	2.9	1.3	2.1

1) Mastery of the creation of exchange amongst type diagrams [3] .

2) Accuracy in the time standards

3) Knowledge impacts

4) Exhaustion impacts

5) Determination impacts

6) Subject inspiration

B. Threats to outer Validity

Outer legitimacy is that UML diagrams used as an arrangement object for making OOIS and to various examination settings[1].

XII. RESULT

From the descriptive measurements, we noticed some observations and so actions were involved. Some of them have been said as takes after [13][14]:

- As NAggH Median and Mean value are minimum in 22 UML Class diagrams, so we conclude that the use of Aggregation in 22 UML Class diagrams is limited.
- We have removed NOM, WMC, MaxDIT and NGenH from 22 UML Class diagrams because it has been observed that all data points for NOM, WMC, Maximum DIT and NGenH are zeros in the 22 UML Class diagrams.
- We have observed that the classes, attributes between classes in 22 UML Class diagrams were high in RFC, NOM, and WMC 22 UML Class diagrams.
- Values of Mean and Median of NC are almost same in 22 UML Class diagrams that mean UML Class diagrams have almost similar classes.
- If WMC quality is high, it upsets the re-utilization of the class and to stay away from this we have to decrease the quantity of strategies or their complexity [13].
- If the DIT is more than six, it expands the design complexity and subsequently decreasing the inheritance utilization while coding.

TABLE IV. UML CLASS DIAGRAM WITH SIZE METRICS AND STRUCTURE COMPLEXITY METRICS

Name of Software	Size Metrics				Structure Complexity Metrics						
	RFC	NOA	NOM	WMC	NOC	NC	NOR	NGen	MaxDIT	NAggH	NGenH
U1	7	6	7	7	3	8	9	5	4	0	2
U2	22	7	22	22	2	5	4	4	2	1	2
U3	8	13	8	8	3	4	4	3	0	0	2
U4	9	6	9	9	1	3	2	3	0	0	1
U5	8	11	8	11	3	4	3	3	2	2	5
U6	5	13	5	5	2	3	8	8	2	6	2
U7	14	23	18	14	3	7	7	6	0	0	3
U8	30	23	30	30	5	6	8	5	0	1	3
U9	18	7	18	18	3	4	3	3	0	1	1
U10	6	4	6	2	3	4	5	4	2	0	2
U11	9	12	6	9	7	8	8	3	0	0	2
U12	16	2	16	16	6	7	7	7	5	0	4
U13	7	2	2	7	2	3	2	1	2	0	0
U14	3	3	3	5	3	4	4	2	4	0	4
U15	5	11	5	6	3	4	4	2	3	0	2
U16	10	6	10	10	4	5	5	3	4	3	0
U17	30	23	30	30	5	6	7	2	6	1	0
U18	5	13	5	5	6	7	7	2	2	1	3
U19	11	11	0	0	4	5	4	2	2	1	1
U20	8	8	8	8	2	3	2	2	2	1	0
U21	12	9	12	12	4	5	7	3	4	3	0
U22	16	27	17	17	8	7	9	2	2	0	4

TABLE V. DESCRIPTIVE STATISTICS

	RFC	NOA	NOM	WMC	NOC	NC	NOR	NGen	MaxDIT	NAggH	NGenH
Min	3	2	0	0	1	3	2	1	0	0	0
Max	30	27	30	30	8	8	9	8	6	6	5
Mean	12.2	11.2	11.5	11.7	3.8	5.1	5.4	3.5	2.3	1.1	3.0
Median	9	10	8	9	3	5	5	3	2	0.5	2.0
Std. Deviation	7.6	7.2	8.4	8.0	1.8	1.7	2.3	1.8	1.8	1.5	1.5

G. If the NOC is high, weakening of abstraction of classes happens, and subsequently, it diminishes NOC as if an occurrence of DIT is applied here.

H. If the RFC of a class is high, it again achieves more measures of testing attempts that are an augmentation in the test gathering and over design complexity of the

framework increments. Consequently, the number of operations that will be implemented because of a message received must be lessened.

I. An exceptionally compelling part is played by the quantities of subclasses in influencing the maintainability of software.

J. UML demonstrating is obvious to be high priced and not as a subject of course financially savvy.

XIII. CONCLUSION

It is extensively acknowledged that the element of OO Software must be assessed from the underlying levels of its development life cycle. This statistic clue to describe a set of metrics for evaluating the structural complexity of UML class diagrams, through the knowledge that they are related through the maintainability of diagrams. The results obtained through research shows that most of the metrics like NAggH suggested. It is a good indicator of class diagram maintainability and their attributes because NAggH median and mean value are minimum in 22 UML Class diagram and the use of Aggregation is limited in these class diagram. NOM, WMC, MaxDIT, and NGenH have been removed from these 22 UML class diagrams because it has been observed that all data points are zero. Investigational outcomes design that the metric is to a great degree identified with human knowledge and can effectively quantify the complexity of object-oriented software measurement. On the other hand, the outcomes of this research can help the software quality mechanism and assessment demonstrating based on software measurement, increase the accuracy and can be useful to increase software quality, software maintenance work and it can enhance and improve software measurement research. Similarly, WMC quality is high so, the quantity of their complexity has been decreased. In the same way, if DIT is more than six, therefore decreasing the inheritance utilization while coding has been removed. The software maintainability is a noteworthy natural for an application model which is planned to lessen a framework's inclination, and to indicate when it creates much less high-priced and not more hazardous to change the code alternatively to alter it. The high-quality maintainable knowledge of maintainability allows for the aid crew to know, on what module to middle amid preservation.

XIV. FUTURE WORK

Even though the results acquired in this experiment are hopeful. It's major to place on these measures to data got from "actual tasks".

Various modifications that could be prepared to develop the research obtainable are:

A. Expansion the size of the class diagrams. By increasing the size of the class diagrams. Also, as the cases will be more genuine and if work with professionals, could make improved use of their possible ability and accomplish that the outcomes will more common.

B. Increase the change between the values of the metrics. This option could lead to more definite outcomes about the

metrics and their relationship with the factor will try to control.

C. Carry out the research in a further skillful environment.

D. Research with real data. One more way to improve the legitimacy of the outcomes is by functioning with real data acquired from engineering environment .But, the lack of such data continuous to be difficult so must invention other ways to challenge validating metrics.

ACKNOWLEDGMENT

The JECRC University, Jaipur, Rajasthan, India supports this research.

REFERENCES

- [1] Genero B., M., Moody D.L. and Piattini M. (2005), 'Assessing the capability of internal metrics as early indicators of maintenance effort through experimentation', *Journal of software maintenance and evolution: Research and Practice*, 17(3), pp.225-246.
- [2] Dubey S.K. and Rana A. (2011), 'Assessment of maintainability metrics for object-oriented software system', *ACM SIGSOFT Software Engineering Notes*, 36(5), pp.1-7.
- [3] Genero M., Piattini M. and Jiménez L. (2001), 'Empirical validation of class diagram complexity metrics', *In Computer Science Society, 2001. SCCC'01; XXI International Conference of the Chilean*, Chile, 9-9 Nov.2001. IEEE ,pp. 95-104.
- [4] Genero M., Olivas J., Piattini M. and Romero F. (2002), 'Assessing object-oriented conceptual models maintainability', *In Advanced Conceptual Modeling Techniques*, Volume 2784 of the series Lecture Notes in Computer Science, Berlin, Germany,October 7,2002. Springer Berlin Heidelberg, pp. 288-299.
- [5] Dubey S.K., Sharma A. and Rana A. (2011), 'Analysis of Maintainability Models for object-oriented System', *International Journal of Computer Science and Engineering*, 3(12), pp.3837-3844.
- [6] Kuhl, Frederick S.(1990), 'Object-oriented programming applied to a prototype workstation', *Software: Practice and Experience*, 20(9), pp. 887-898.
- [7] Genero M., Piattini M. and Calero C. (2002), ' Empirical validation of class diagram metrics', *International Symposium on Empirical Software Engineering*,Nara ,Japan, October 3-4,2002. IEEE ,pp. 195-203.
- [8] Genero M., Olivas J., Piattini M. and Romero F. (2001), 'Using metrics to predict OO information systems maintainability', *In Advanced Information Systems Engineering* , Volume 2068 of the series Lecture Notes in Computer Science, Sweden,June 28,2001. Springer Berlin Heidelberg ,pp. 388-401.
- [9] Genero M., Jiménez L. and Piattini M. (2002), 'A controlled experiment for validating class diagram structural complexity metrics', *In Object-Oriented Information Systems*, Montpellier, France,September 2,2002.Springer Berlin Heidelberg ,pp.372-383.
- [10] Genero M., Poels G., Manso E. and Piattini M. (2005), 'Defining and validating metrics for UML class diagrams', *Metrics for Software Conceptual Models*, London, UK, January 4, 2005. Imperial College Press,pp.99-159.
- [11] Genero M., Piattini M., Manso E. and Cantone G.(2003), 'Building UML class diagram maintainability prediction models based on early metrics', *Ninth International Symposium in Software Metrics*, Sydney ,NSW,Australia ,September 3-5,2003.IEEE ,pp. 263-275.
- [12] Fernández-Sáez A.M., Genero M. and Chaudron M.R.(2011), 'Does the level of detail of UML models affect the maintainability of source code?', *Models in Software Engineering* , Wellington, New Zealand, October 16-21,2011. Springer Berlin Heidelberg, pp. 134-148.
- [13] Al Dallal J. (2013), 'Object-oriented class maintainability prediction using internal quality attributes', *Information and Software Technology*, 55(11), pp.2028-2048.
- [14] Genero M., Poels G. and Piattini M.,(2008), 'Defining and validating metrics for assessing the understandability of entity–relationship diagrams', *Data & Knowledge Engineering*, 64(3), pp.534-557.