

# Evaluate The Software Through Object Oriented Paradigms

Ahmad Mateen  
University of Agriculture Faisalabad,  
38000 Pakistan

Marriam Nazir  
University of Agriculture Faisalabad,  
38000 Pakistan

## ABSTRACT

Programming assumes an essential part in today's electronic world. The developer utilizes diverse dialects to build up the product. Keeping in mind the end goal to create programming, it needs a few stages, for example, Analysis, Design, Implementation, Testing and Maintenance. Of these few stages, the investigation and outline get to be fundamental, since these are the most crucial component in the advancement of the product. Presently a-days, a large portion of the product is article stated, in light of the fact that the item arranged dialects give a simple approach to create and keep up the system. This item situated project is comprised of a few divisions based upon the reason. Every division performs some capacities subordinate upon the code. Every one of these divisions are then incorporated to give the single system. In the event that any blunder happened in any part of the project implies, it is important to change the influenced part of a system to expel the mistake. To keep away from this sort of pointless change with the long time term, the engineer must diagram and tests the underlying stage, for example, investigate and configuration. These sorts of tests on examination and outline for an article arranged project did by an innovation called the OOP (Object Oriented Paradigms). In this paper, an approach proposed for examination, the configuration to be completed in the improvement of the system, before begin to execute. These measurements developed by taking or gathering substantial volume of information keeping in mind the end goal to give the measurements reasonable to a wide range of Object-Oriented dialects, for example, Python, Ruby, .Net and R software for big data analysis. The proposed work gives the summing up procedure which termed as GenM (Generalized Measurement) strategy on the item situated programming. This is finished by executing the arrangement record for recognizing the mistake rate. Along these lines, this paper gives proficient systems to OOP.

## General Terms

Error Rate, Object-Oriented, Paradigms, Software, Design, Testing.

## Keywords

Analysis, OOP, Implementation, Configuration File, Phases, Maintenance.

## 1. INTRODUCTION

Presently a-days, everything gets to be mechanized. With a specific end goal to change the things automated, every one of the things are changed over into programming design. Numerous sorts of programming development by method for booking dialects. Of these dialects, a wide range of sorts exists based upon the kind of difficulty. At first, the programming dialects did not take after any structure and hence it was termed as Unstructured Programming

Languages. The following stride is to conquer the unclear downside by method for presenting Structured Programming Language. In this organized programming dialect, as the length of the system builds, meaningfulness gets to be troublesome. That sort of dialect is additionally termed as Procedure-Oriented Language (POL).

The Software quality is now a significant part in software design and style, helping the designer to take care of the complexity of large systems [1]. According with today's program development, object-oriented design and style is a favorite concept. It features proved its value for a program that preserved and modified [2]. Current concepts connected with Structured in addition to Object-Oriented analysis are generally not comparatively new but these approaches still play a very essential role in the production, programming paradigm [3]. By "evaluation" it suggests a methodical method for analyzing a program in detail. The goal of such an examination is to analyze the top quality of the software product concerned, not the high quality of the process made use of to create the product [4].

SDLC, Software Development Life Cycle is a procedure utilized by programming industry to plan, create and test amazing programming projects. The SDLC plans to deliver a high quality programming that meets or surpasses client desires, achieves culmination inside times and cost gauges. With a specific end goal to create programming, the real key strides are as per the following: Analysis, Design, Implementation, Testing and Maintenance.

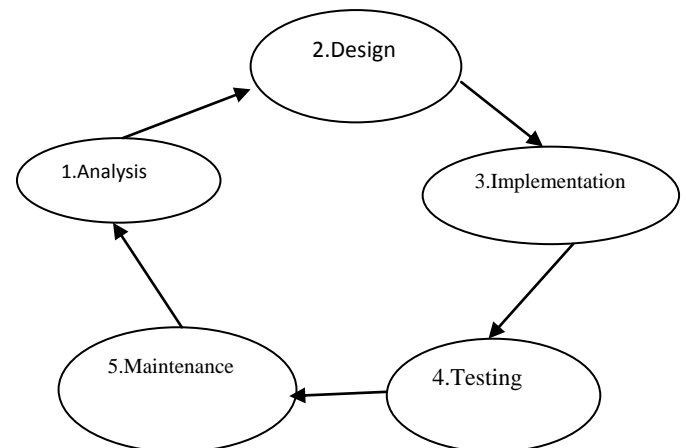


Figure1. System Development Life Cycle (SDLC)

The product advancement life cycle (SDLC) is a system characterizing errands performed at every progression in the product improvement process. ISO/IEC 12207 is a global standard for programming life-cycle forms. It expects to be the standard that characterizes all the undertakings required for creating and keeping up programming.

SDLC is a procedure took after for a productive venture, inside a product association. It comprises of a nitty gritty arrangement portraying how to create, keep up, supplant and modify or improve particular programming. The life cycle characterizes a philosophy for enhancing the quality of programming and the general advancement process.

Of these, the real care must be taken in the piece of Analysis and Design. These are the underlying strides to build up the product. In the event that any progressions held in these strides, the entire Software must be influenced. With a specific end goal to quantify the nature of programming from the underlying stage and to accept the product to conclude the convenience of the product based upon the mistake rate is done by utilizing the new method called the OOP (Object-Oriented Paradigms).

The Object Oriented Paradigms objectives are as per the following[19]:

- It is easier to change the software when the programmer or client needs to change it according new circumstances or new demands and new requirements, which changed with the passage of time.
- A programmer changes the class or adds new functionality according to the different demands, it is also easy to change design according business rules or modification in the design.
- When programmer changes one set of application, the other set of application also changed because the influence of change of one set on another.
- Manage dependencies between classes and packages of classes to minimize the effect of progress on different parts of the program.

Quality contains all attributes and huge elements of an item or a movement which identify with the wonderfulness of giving necessities [5]. Quality viewpoints are diverse properties of source code, for example, attachment or coupling, test-related properties, for example, scope, or design related properties, for example, correspondence orders or information access imperatives. A scope of every one of these perspectives is just conceivable when SQA includes diverse partner bunches [6].

Design metrics play a great clicks rule throughout helping developers recognize design aspects regarding software hence, improve software quality and also developer productivity [7]. In any case, a slight code blunders in a space transport's direction PC may be mission basic and jeopardize human lives. Programming quality in PC hones has turned out to be profoundly essential. Seeing the infiltration of the PC code into ordinary articles like clothes washers, cars, iceboxes, toys and even things like the blemishes wanderer, any framework be is it a huge one or a little framework, guaranteeing the largest amounts of programming quality is fundamental. [8].

## **2. PREVIOUS WORK**

A decent programming programmer utilized measurements to survey the standard and quality of software, application, source program code and examination ways. Metrics applied to improve software productivity and efficiency by fixing errors, and then these errors are removed from the software by programmers to establish quality and accuracy. Quality represents the uniqueness which increased its usefulness. Such qualities denoted a point of fineness that differentiated the

item from other folks. Proficiency, Complexity, Understandability, Reusability, Testability, viability is the diverse amazing components used to survey nature of a product stock.

Formal particulars associated with project testing, streamlining, refactoring, documentation, and, above all, investigating and repair. Be that as it may, they were hard to compose physically, and programmed mining methods experience the ill effects of a lot of false positive rates. To address the issue, they proposed to expand a fleeting property excavator by consolidating code quality measurements. They quantified the code quality by extricating extra data from the product designing process and utilizing data from code that will probably be right, and additionally code that was more averse to be right [9].

The issues which faced within the industry to develop the most effective item with decreased rate are following: In order to decide the top quality also to boost the product providers, the industry seems to be converging in the direction of a set of common metrics, such as however, not limited to, apparatus supply, enough time to end situations, agencies had embraced the same along with continual concentrate on the internal techniques, and so forth. They learnt the tradeoff among the price of an administration (as portrayed by the staffing level) and the related top quality measurements (subject towards assistance amount agreement). In particular, they learnt the price of assistance top quality by using a great design that took under consideration the constraints along with charge components typically found in a supplier environment.

The model started with an arrangement of low-level security measurements in view of conventional outline qualities of item situated classes, for example, information epitome, attachment and coupling. At last, the whole project's security was abridged as a solitary security record esteem. The model was accepted by means of a trial, including five open source Java programs; utilizing a static examination device they had created to consequently separate the security measurements from incorporating Java byte code [10].

Software quality often a higher level characteristic that provides a comprehension the internal durability and stability towards used on occasion. Since, it is often a higher level high quality attribute, it is just not possible to immediately measure this buy should be inferred indirectly over the design properties involving software.

At this time, there is no doubt quality is the most essential element in any sort of business. Accomplished a respectable position in the worldwide business sector in IT industry, an organization must need to deliver brilliant items. Rivalry is high and one can't manage the cost of remedying mistakes in the wake of transporting the items to the client [11].

Service-Oriented Structures (SOA) ways that improved programming interoperability through uncovering dynamic applications as administrations. To judge the configuration associated with administrations in administration based techniques, quality estimations wound up crucial to choose tradeoffs in the middle of SOA quality attributes. In the archive they presented your auxiliary quality associated with administration granularity for that investigation of extra inward basic application characteristics: intricacy, union alongside coupling. Thusly, measurements more often than not proposed for ascertaining SOA inside qualities utilizing linguistic structure code.

The utilization of execution measures in business was scarcely totally new. Organizations had likewise been measuring costs, top quality, amount, process duration, effectiveness,

profitability, and numerous others, of items, suppliers and procedures. Open-source advancement forms experienced rose was a powerful strategy to lessen process duration alongside decline outline, execution, and quality affirmation costs for a few shots in regards to programming, particularly frameworks, structure programming, including working frameworks, compilers alongside dialect preparing apparatuses, distributors, and syndication middleware. Second, they exhibited the aspirations and strategy in the Skoll errand, which focused on creating alongside experimentally approving new open-source application quality true serenity and improvement procedures to determine urgent open-source challenges.

Tried the speculation that nonexclusive recuperation strategies, for example, process sets, survived most application deficiencies without utilizing application-particular data. They analyzed in point of interest the shortcomings that happen in three, substantial, open-source applications: the Apache web server, the GNOME desktop environment, and the My SQL database [12]. Estimation in programming, building can be best depicted as far as measurements, as a quantitative measure of the degree to which a framework, a segment or process has a given characteristic. The desire for programming measurements permits the examiners, fashioners, coders, analyzers and the supervisors imagine the product improvement forms and survey the framework [13]. In the item situated environment, one of the significant viewpoints having a solid impact on the nature of coming about programming framework is the configuration many-sided quality [14].

The idea of many-sided quality starts from framework hypothesis. It could be characterized as "a matter of an article with numerous entwined components, viewpoints, subtle elements, or characteristics [15]. An arrangement of measurements is proposed amid this study as a feature of an endeavor to assess the coupling, attachment and multifaceted nature of the administration, situated configuration and hypothetically approved [16]. It also applied to point their education regarding the interdependence among the ingredients by providing precious opinions concerning top qualities, for example maintainability, modifiability along with understandability. Time with regard to software program upkeep generally incorporated a higher correlation using complexity regarding the style.

Facet Oriented Software package Layout is a good rising technique that provides effective brand new approaches to raise modularity regarding software program coming from the style. Measuring programming is a device that permits the improvement of value programming for its whole life cycle. For programming, estimation programming measurements are utilized [17]. A key goal of programming, building is the improvement of programming quality. Be that as it may, the current SOA quality metric spotlight on the expansive estimations of outside basic programming, administration properties, (for example, multifaceted nature, reusability and execution) [18]. They quantified the code quality by removing extra data from the product designing process and utilizing data from code that will probably be right, and in addition code that was more averse to be right [19].

### 3. PROPOSED WORK

OOP is the system to quantify the nature of the product utilizing distinctive OOP measurements. OOP measurements are excessively various, making it impossible to gauge the product in various ways. In this proposed work, it is created with new OOP measurements to decide the quality and

blunder rate of the product to improve the nature of the product.

To robotize the testing on the Object-Oriented Design by OOP, OOP Metrics are proposed in this paper. These measurements are developed by taking or gathering substantial volume of information with a specific end goal to give the measurements appropriate for a wide range of Object-Oriented dialects, for example, Python, Ruby, .net and R. This OOP metric depicts about the examination, a plan that is completed in an item arranged dialect. The interest of value programming is expanding step by step because of social reliance on the customers to the product. For example, engineering, interface and combination and so forth are the fundamental programming outline deformities. Any issue in programming brings about money related misfortune and time delays. Today's product must guarantee reliability and mistake free execution at whatever point it is utilized. Programming outline is a critical part in the nature of the product. Poor configuration brings about the more prominent revamp and higher expense.

### 3.1 Experimental Design

Based upon the OOP measurements, the nature of the product is assessed. The proposed work is giving the summed up approach which termed as GenM (Generalized Measurement) system on the item situated programming measurements and it is better, to decide the project quality.

This OOP metric examination the nature of the project by taking the system and afterward investigation the system by distinguishing the kind of dialect utilized, number of classes utilized as a part of the system and the sort of the classes utilized as a part of the project. Every one of these things is given as Configuration documents.

Aside from these things, the arrangement record additionally contains the data about the blunders found in the system, sort of the mistake and the mistake rate. The blunder is likewise characterized into a few classes based upon the kind of the mistake. These things put away as takes after:

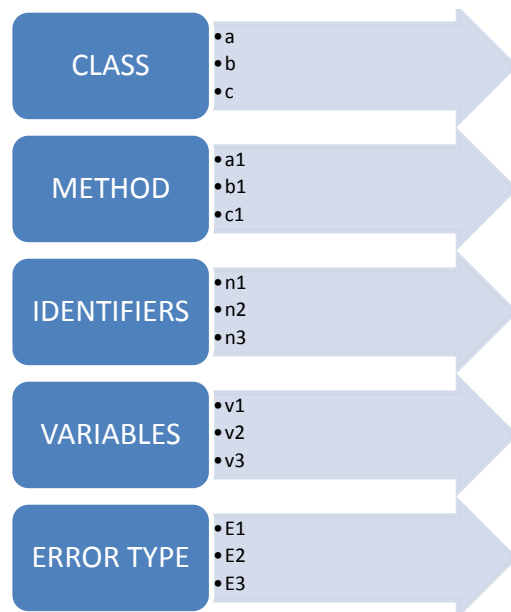


Figure2. Parts identification of program

From this setup document practically everything about the system is put away by examination the outline. Based upon the mistake sort, the blunder rate is resolved and the nature of

the project is figured. The mistake rate is separated for various sorts of blunders, for example, sentence structure blunder, run-time blunder et cetera. A portion of the blunder rates is characterized underneath.

The fundamental thought is to evaluate the software quality for judging the different kind of errors in a software application, which cause bad or low performance. To improve the software quality and performance, it is very necessary to fix the bugs and error ratio in the affected part of software, which results huge problem in the software for excellent performance.

**Table 1. Types of Error**

ERRORS IDENTIFICATION (%)	
Syntax Error	6%
Logical Error	0%
Compilation Error	4%
Run-Time Error	8%
Data-Type Mismatch	2%
Executable Files	5%
Other Errors	10%

The above chart shows the percentage of different kinds of error, e.g. (syntax error, run-time error, data-type mismatch, compilation error, executable files and other types of error). That indicates the quality of application by show the error rate.

Quality is a very important factor. Software quality basically based on different factors, but the most important factor is customer satisfaction which is directly concerned applying standards. In this area achieving quality software is very important because of the high customer software industry and improving day by day. Meanwhile, developing countries like Pakistan are struggling with software quality.

While computer professionals, people make an effort to build a system that is useful and that work; as computer software engineers, people encounter the task of developing complex system with less computing in addition to human resource. Besides, the present time is regarded as an entire world of objects. These kinds of objects exist in nature, with man-made entities, in business, and within the products that most of us use. They are categorized, described, arranged, combined and manipulated in addition to creating. Therefore, an object-oriented view is a picture creation of such methods.

The setup record is worked as takes after with all the fundamental attributes. This setup record comprises of three segments, for example, <compile>, <OOP> and <error>.

- compiling the programming file.
- OOP is used for analysis and design of the program
- error rate found with in the software.

```

<configuration-file>
<config>
<compile>
<program-type> Python, .net, Ruby and R.
<OOP>
Class-Number, Class-Type, Method, Identifiers,
variables and data type
<error>
Syntax Error, Logical Error, Compilation Error, Run-
Time Error, Data-Type Mismatch, Executable Files,
Other Errors
</error>
</OOP>
</compile>
</config>
</configuration-file>

```

**Figure 3. Configuration file**

This proposed system additionally comprises of a calculation to decide the nature of the project. The calculation is given beneath with appropriate clarification.

```

Check Input
Measurement of the design by applying OOP
Examine it if mistake , Recognize the mistake in result
then sort the fault
If no error ll error <find the threshold value so
Quality = "good"
Otherwise
Based upon the fault rate, the nature of the project is
determined
End if

```

**Figure 4. GenM Algorithm**

## 4. RESULTS

The Proposed strategy is extremely productive that is utilized to assess the nature of the product. To test the effectiveness of the technique, different exploratory setups are developed and the outcome is dissected. The trial setup is made by taking the arrangement of clients with two classes, for example, with programming aptitudes and without programming abilities. The client with programming aptitudes can examine the sort of system and to incorporate the project. At that point they utilize the arrangement document for examination the OOP Metrics and to reason the mistake rate.

At last, the outcome is submitted to the client reporting the nature of the product and the ease of use of the product is dictated by the client based upon the blunder rate.

In this manner, this proposed approach gives better trial results when it is actualized on a programming. In this way, it is extremely helpful for a wide range of clients to examination the nature of programming. The accompanying table-1 demonstrates the yield of the example program created

utilizing this proposed approach to assess the blunder rate in a given project.

**Table 2. Classification of Errors**

REPORT	
Error type	Count
Syntax error	1
Semicolon Expected	4
Invalid Object	2
Invalid Identifier	1
Invalid Conversion	3

There are different kinds of error that exist in a program. These error types are classified in table-2 e.g. syntax error, semicolon expected, invalid object, invalid identifier and invalid conversion. This report identifies the count of errors. The table-2 shows numbers of error of each error type that are existing in a sample program. According to this report, syntax errors and invalid identifier are comparatively lower than invalid object. While invalid object is comparatively lower than semicolon expected and invalid conversion. This is very helpful to measure the quality of a software by fixing the errors. It's quite easy to measure the quality of a software by determining errors and then classifying them within accurate number of errors.

Taking into account the trial setup, different results are distinguished. Likewise to guarantee the nature of the proposed philosophy, numerous examinations are attempted with the current methods of investigation the rate of execution.

**Table 3. Quality measurement methods and performance rate**

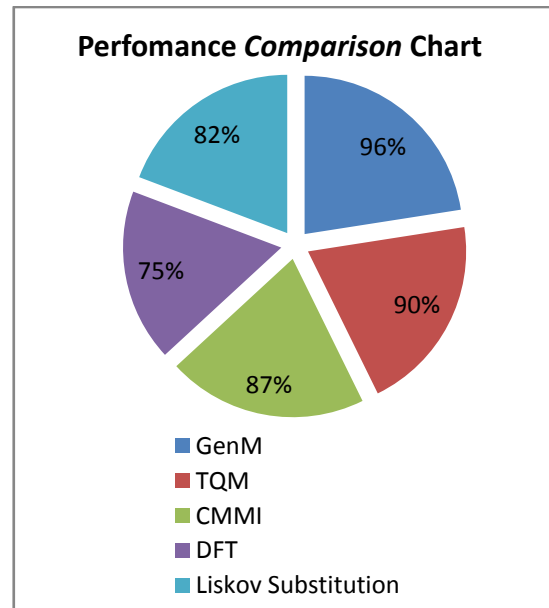
Quality Measurement Methods	The Rate of Performance (%)
GenM (Generalized measurement)	96%
TQM (Total quality management)	90%
CMMI (Capability maturity model integrated)	87%
DFT (Design for test)	75%
LS (Liskov substitution)	82%

It is very important to compare and analyze the proposed methodology with existing techniques that are used for measuring the quality of software. Here is some techniques and their performance ratio in table-3 that are illustrating comparison with the high ratio performance of *GenM (Generalized measurement)*.

In table-3, it is defined that TQM, CMMI, DFT and Liskov Substitution (LS) has a low performance rate than GenM. According to the table-3, ratio of TQM (total quality management) is 90% and the ratio of CMMI (capability maturity model integrated) is 87% and the ratio of DFT (design for test) is 75% and the ratio of LS (liskov substitution) is 82%. While the ratio of GenM (Generalized

measurement) is 96%, Which is the highest ratio from TQM, CMMI, DFT and Liskov Substitution (LS). Thus, these results identify that GenM is best to gauge the nature of a product.

The proposed system is tentatively confirmed by executing the algorithm. The proposed algorithm is implemented and the efficiency of the program is determined. First of all, the program is compiled for its results. From the results, the efficiency is determined through the implementation of the algorithm. From the experimental result, it is proven that the proposed method performs well. The above table shows the higher performance rate (96%) of *GenM (Generalized Measurement)* technique by comparing different quality measurement methods.



**Figure 4. Performance Comparison Chart**

The figure 4 is a description of five techniques's percentages. In this chart, highest performance of GenM (96%) is indicated by blue section while TQM (90%), CMMI (87%), DFT (75%) and Liskov Substitution LS(82%) techniques are indicated with different colors like red, green, purple and zinc. The figure-4 is a demonstration of performances in (%) of five techniques GenM, TQM, CMMI, DFT and LS.

Object oriented programming is a modern and powerful technology which is leading a successful path in this today's programming world. This study presents some object oriented metrics that are used to measure the program efficiency. The results of the object oriented metrics is implemented on various experiments and the results are comparatively good as above performance comparison chart is elaborating the performance rate of CMMI, TQM, DFT, Liskov Substitution (LS)&GenM.

## 5. CONCLUSION

Object-oriented Paradigms (OOP) is encouraging programming industry with more solid and sensible programming and its outline. The nature of the product outline is measured through various strategies and approaches. In the event that any mistake happened in any part of the project implies, it is important to change the influenced part of a system to uproot the bug. It is possible only by identifying errors and by measuring the quality of software. Along these lines the nature of the product is measured and the outcome is mechanized by the proposed philosophy of this examination.

This philosophy gives a proficient and appropriate instrument to quantify the nature of the product. This estimation is made by breaking down the outline utilizing the design document give by the examination work. Different OOP measurements are proposed alongside the accumulation method for various sorts of dialects and the different blunder systems. Along these lines, this exploration work is finished effectively with a productive strategy proposed in this work.

## **6. REFERENCES**

- [1] A. Chandrasekar, S. Rajesh and P. Rajesh, "A Research Study on Software Quality Attributes", *International Journal of Scientific and Research Publications*, 2014, 4(1), 14-19.
- [2] R. Huang, M. Li and Z. Li, "Research of Improving the Quality of the Object-Oriented System", *International Journal of Information and Education Technology*, 2013, 3(4), 3-6.
- [3] S. K. Thakkar, K. K. Thakkar and N. M. Satra, "Object Oriented Designing and Modeling", *International Journal of Advanced Research in Computer Science and Software Engineering*, 2015, 5(1), 905-912.
- [4] D. L. Parnas, M. Lawford, "The Role of Inspection in Software Quality Assurance", *IEEE transactions on software engineering*, 2003, 29(8), 674-676.
- [5] A. B. Tomar, V. M. Thakare, "A systematic study of software", *International Journal of Software Engineering and Applications*, 2011, 2(4), 61-70.
- [6] Martin, B. 2013. *Fostering Software Quality Assessment*. Software Engineering (ICSE) 35th International Conference. 3(13), 1393-1396.
- [7] R. Subramanyam, M. S. Krishnan, "Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects", *IEEE transactions on software engineering*, 2003, 29(4), 297-310.
- [8] S. M. Rawat, "Survey on Impact of Software Metrics on Software Quality", *International Journal of Advanced Computer Science and Applications*, 2012, 3(1), 137-141.
- [9] C. L. Goues, W. Weimer, "Measuring Code Quality to Improve Specification Mining", *IEEE Transactions on Software Engineering*, 2012, 38(1): 175-190.
- [10] Alshammari, B., Fidge C. and Corney, D. 2011. *A Hierarchical Security Assessment Model for Object-Oriented Programs*. Faculty of Science and Technology, Queensland University of Technology, Australia, 11th International Conference on Software Quality, 4(2): 218-227.
- [11] A. K. Shah, "How to improve software quality assurance in developing countries", *Advanced Computing: An International Journal*, 2012, 3(2), 17-28.
- [12] Chandra, S. and Chen, P. M. 2000. *Whither Generic Recovery from Application Faults? A Fault Study using Open-Source Software*. Computer Science and Engineering Division, Department of Electrical Engineering and Computer Science, University of Michigan. 7(2): 97-106.
- [13] Yeresime, S., Jayadeep, P. and Santanu, K. R. 2012. *Effectiveness of software metrics for object-oriented systems*. 2<sup>nd</sup> International Conference on Communication, Computing and Security.
- [14] Selvarani, R., Gopalakrishnan, N. and Kamakshi, P. 2009. *Estimation of defect proneness using design complexity measurements in object-oriented software*. International Conference on Signal Processing Systems.
- [15] Hycham, A. and Bruno, M. 2015. *Measuring the complexity of a higraph-based system model: formulism and metrics*. Conference on Systems Engineering Research.
- [16] Arafat, A. M. E. and Radziah, M. 2014. *Metrics for evaluating the quality of service-oriented design*. 8<sup>th</sup> Malaysian Software Engineering Conference.
- [17] Alberto, N., Hector, G. P., Juan, C. C. and Carlos, S. 2013. *A methodology for obtaining universal software code metrics*. The Iberoamerican Conference on Electronics engineering and Computer Science.
- [18] Saad, A., Ed, Z. and David, C. D. R. 2011. *A metrics framework for evaluating SOA service granularity*. IEEE international Conference on Service Computing.