International Journal of Engineering & Technology, 7 (1.5) (2018) 13-18

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Software metric evaluation on cloud based applications

A. Phani Sheetal [1*] , K.Ravindranath [2]

[1]Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522502
[2]Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur Andhra Pradesh, India-522502
**Corresponding author E-mail: uma.15phani@gmail.com*

Abstract

Unbound growth in the cloud computing service models have motivated the companies building traditional software to be migrated into the clouds. During the high demand of the traditional applications, the performance and quality of the software were evaluated by the popular and globally accepted metrics. Nevertheless, after the migration of the same applications into the cloud, the expectation and definition of performance and quality has been changed. The beneficiaries of these applications are setting new milestones for the applications. Hence, the recent demand of the research trend is to build new software metric models to match the trade of between the new expectations from the beneficiaries and the software quality policies for organization or individual or state. Thus this work makes an attempt to understand the traditional software quality metrics and try to justify the applicability of these parameters in the trend of cloud based software applications. This work also proposes a novel metric method for performance evaluation for the migrated applications into the cloud, with the intension of formalizing and standardizing the cloud based metric methods unlike the recent trends.

Keywords: *Software Metrics; Direct Analysis; Indirect Analysis; Cloud Data Centre Performance; Architectural metric model; Design metric model; Quality Index*

1. Introduction

The quality control for the software applications have became the most important issue. Most of the applications in use are majorly deployed for making precise calculations and dealing with mission critical data [1]. It is been demonstrated by various researchers that, the poor quality of the software or the extreme negligence of software quality may lead to extreme high maintenance for the software applications. Henceforth it is natural to understand the importance of software measurements techniques and paradigm in all the stages of software development. The measurement of software quality, performance and other issues can be formulated in multiple highly popular techniques called metrics.
Halstead et al. introduce the software metrics in the year of 1972 [2] with the intension of standard methods for measurement in software applications. The main objective of the metric is to improve the software quality, performance and maintainability. From the time of inception of software metrics focused on two major types of attributes as internal attributes and external attributes [2]. Based on the types of the attributes, the analysis of the software applications are also classified into two types as direct measurements and indirect measurements [3]. The direct assessments are calculated using the internal attributes and the indirect assessments are calculated using external attributes. Further, the analysis of quality can be achieved using both the assessments. Hence a number of research attempts are made to map these two attributes [4]. The outcomes of McCall et al. and Boehm et al. are notable in this direction. They identified reliability, usability and maintainability as most prominent factors for quality [5]. Similarly, the mapping between the attributes and classify the same into sub attributes groups are also a notable work by Dromey et al. [6].
However, the migrations of legacy software into cloud computing service models are becoming highly popular due to the pay per use modularity. This work also analyses types of computing in order to achieve the higher understanding the cloud computing paradigms shift for data centre applicability [7] [8] [9].
This understanding will help in formulating the attributes of legacy software metric into cloud based software application metrics.
The rest of the paper is organized such that, in Section – II the software metric current improvements are been analyzed, in Section – III the cloud service models are been analyzed, in Section – IV traditional cloud based data centre performance models are been analyses, in Section – V the current tradeoffs are been discussed, in Section – VI proposed cloud based software metric models are been presented, in Section – VII the results are been analyzed and the work presents the conclusion in Section – VIII.

2. Software metrics

The software metrics are classified into multiple classifications and each of the categories is having advantages and shortfalls. A number of research attempts are been made to establish the classifications and the classifications are made standard. The classification standards are listed in priority for many of the book authors as well [10] [11]. The different classes of metrics are deployed for different categories of facets of the software applications.

The first classifications of software metrics are demonstrated by G. Eason et. al. based on the simplicity, objectivity, obtainability and robust nature [12]. The following are classification categories are furnished here:

- **Process Metric**: The process metrics are deployed to evaluate the software development process costs or duration or methodology attributes.
- **Project Metric**: The project metrics are deployed to evaluate the software project situations, cost, and number of employees and skill sets of the employees.
- Product Metric: The product metrics are deployed to evaluate the product at any phase of the development. The project metrics are also classified in to static metric and dynamic metric [13].

Henceforth, the detailed understandings of the types of metrics are listed in this work.

A. Syntactic metrics

A majority of the metrics are based on syntactical aspects of the software in the modern day of research and development. These metrics focuses on the way programs are written rather how they function. The highly popular metrics are LOC as represented line of code and cyclamate complexity [14]. The LOC metric determines the complexity of the code based on the size and in the other hand the cyclamate complexity metric determines the complexity of each module. Overall the syntactic metrics are designed to analyse the source codes for quality based on various attributes. The quality measures mean the reduction of bugs in the source code [15]. A problem of syntactic metrics is to showcase of the different metric parameters based on the structural aspects of source analysed [16].

B. Semantic metrics

The basics limitations in terms of the acceptability of the attributes in syntactic metrics demands higher order of metrics are to be deployed for sufficient amount of analysis. Henceforth, the Semantic metrics are been understood in this work. Gall et. al. suggests an approach using semantic metrics to provide insight into software quality early in the design phase of software development by automatically analysing natural language (NL) design specifications for object-oriented systems is presented. Semantic metrics are based on the meaning of software within the problem domain [17] [18]. Thus these basic understandings will help this work to build the cloud based metrics for the software applications migrated from traditional to cloud based. However, the metrics are closely depended on the cloud service models.

3. Cloud service models

The motivation from the benefits obtained from cloud computing have enforced the demand to push various legacy applications to the cloud. The recent advancements and studies by many researchers have demonstrated the benefits of all the cloud service models, which are discussed in this work. Significantly high availability, pay per cost model for cost reduction and manageability of the applications and data influences the adoption of three major cloud service models such as infrastructure, platform and software services [19].

The fundament designs of cloud service models are to justify the higher availability of the cloud based applications and data during the high demand segments. With the introduction of virtual machines and capabilities of virtualizing every hardware resources provides the capability of maximizing to the application owners and at the same point, enables the cloud service providers to bill the application owners based on the usages rather than dedicating the complete hardware resource pools. The virtualization also provides the separation and isolation of application, data and auditing process for each and every application. As the computing infrastructure is hosted on the service provider premises, thus the requirements for onsite physical security and need for the maintainability also reduces for the application owners. Nonetheless, the application owns can also have a defined control on the hardware resources over internet access [Figure – 1].

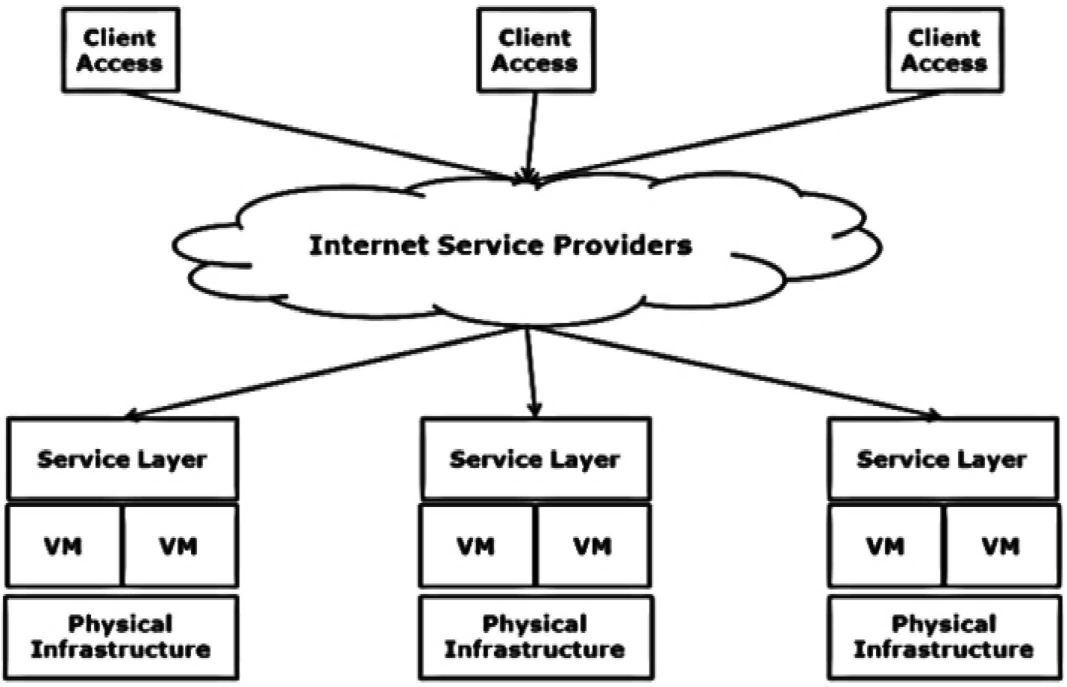

Fig. 1: Generic Grid Computing Architecture

The uncountable benefits of the cloud computing eventually depend on the service model. Few of the benefits are discussed here:

- The feature of on demand provisioning makes the cloud service models effective for agile development.
- The cloud service provider makes the application development easy by introducing multiple APIs for the developers.
- The scalability capability of the cloud service models can scale the infrastructure based on the need of the application owner for small, medium and high availability.
- The seamless access methods over the internet make the application owner, auditors and mostly the consumers access the services from anywhere.

The cloud service models can be classified based on various scientific and commercial needs. The followings are three major classifications:

- **Software as a Service**: The application owners host the application or the part of the application on the cloud.
- **Platform as a Service**: The runtime is provided as a service for the application developers to build and host the application.
- **Infrastructure as a Service**: Customers are allowed to rent and use the infrastructure provided by service providers.

3.1 Traditional data center based software performance metric

The traditional cloud based or data centre based applications are mainly depended on the hardware performance of the data centre. Henceforth, firstly it is important to analyse the hardware or architecture metric of the cloud based data centre. This work formulates the hardware metric of the data centre [Table – 1].

Table 1: Data Centre Hardware Performance Metric

SNO	Details of the Metric Attributes	
	Attributes Name	**Description**
1	Node CPU	The number of Tera Flops in the CPU
2	Memory Bandwidth	The product of the memory clock, the transfers per clock based on the memory type, and the memory width
3	Network Bandwidth	The Transfer speed of the network interconnecting the nodes of the data centre
4	Power Consumption	The total power consumption of the data centre
5	Node Power Consumption	The average power consumption of any node in the data centre

Table 2: Metric Parameters for data Centre Based Applications

SNO	Details of the Metric Attributes	
	Attributes Name	**Description**
1	Resource Provisioning	The duration for provisioning the virtual resources connected to the software application
2	System Availability	The availability of the software application over cloud that is the service level agreements
3	Resource Tracking	The number of virtual resource allocated to the application and the detailed usage of these resources
4	Energy Consumption	The consumption of the energy for running and maintaining the application on the data centre
5	Inventory Size	The amount of physical resources are available to be allocated to the application during high demand
6	Issue Tracking	The time needed for tracking the issue and provide solution

Thus these attributes of the hardware performance metric will have significance influence on the deployed software application in data centres. Nevertheless, the software application performance met-
With this understanding of the metric parameters, it is natural to understand the gaps in available metrics for improvements. In the next section of this work, the challenges are been highlighted.

3.2 Challenges on the Present Research

In this section of the work, the current challenges are been highlighted in case of software performance metrics on cloud environment.

After the in detail understanding of the problems in the data centres, we propose the following solutions in order to justify the proposed framework for cloud based data centre component optimization:

With the recent advancements, the architecture of data centre along with cloud based data centre needs to be studied in order to propose a novel optimized distributed cloud based data centre architecture.

1. The advancements of networking components for cloud-based data centre needs to be understood and propose a novel optimized distributed communication path along with the parameters for performance evaluation.
2. The load balancing techniques to be studied for data centres and cloud architecture and proposes a novel framework for optimized load balancing for the workloads.

1.

However, the challenges of the software metrics are also to be understood in this work. Thus, the challenges of semantic metrics are presented here [20].

rics will have distinguished parameters in the metric [19]. This work also demonstrates the attributes with description [Table – 2].

- Firstly, the metrics are majorly focuses on the software applications, which are traditions. Hence incorporation of the various platforms such as cloud based data centre oriented parameters is to be incorporated.
- Secondly, the semantic metrics majorly focuses on the attributes collected during the development phase, thus the real performance evaluation can be a challenge. Hence it is strongly suggested that the attributes are to be collected during design phase to the implementation phase and also in the maintenance phase for better understanding during upgrade.
- Finally, the mathematical models are to be built for the better proof of the considerations and providing flexibilities in the analysis using countless available technologies.

In order to avoid the challenges, this work made a preliminary step by suggesting a cloud based data centre oriented software performance metric. The details are been formulated in the next section.

4. Proposed cloud based software metric model

The recent demand for the software metric model for cloud based data centre oriented cannot be ignored. Thus this work proposed a two phase software metric model for application evaluation. The first phase of the software metric model called architectural metric deals with the cloud based data centre parameters and the second phase called design metric deals with the software design issues on distributed architecture like data centres. The description is presented here [Table – 3]. After formulating the novel metric model, this work demonstrates the use on various random data sets. The applications under taken in the results and discussion section are arbitrary.

Table 3: Proposed novel Metric Model for Cloud Based Software Application

SNO	Details of the Metric Attributes		
	Attributes Type	**Attribute Name**	**Description**
1	Architectural Mod-el	CloudLet_ID	Sequence number of the request for Resources
2		CloudLet_Status	Status of the request fulfilment
3		DataCentre_ID	Simulated data centre id
4		VM_ID	Virtual infrastructure id
5		Duration / Time	The throughput time
6		Start_Time	Start time of the job
7		Finish_Time	End time of the job
8	Design Model	Number_Of_Modules	Total number of modules in the source code
9		Independent_Modules	Total number of independent modules
10		Preceding_Modules	Total number of modules where the function depends on the previous module on another node
11		DB_Items	Total number of data base items
12		Unique_DB_Items	Total number of unique database items
13		DB_Segments	Total number of database segments
14		Signle_Function_Exit_Points	Total number of modules with single entry and exit point
15		Quality Index	The final quality index value based on Parameter 8 to 14.

5. Results and discussions

During the testing of this work, a total of 15 random applications are been considered and a total of 110 cloud let jobs are been considered. Hence firstly, this work records the architectural model attribute values [Table – 4].

Table 4: Architectural Model Attribute Valve Analysis

CloudLet_ID	CloudLet_Status	DataCentre_ID	VM_ID	Duration / Time (In NSec)	Start_Time (In NSec)	Finish_Time (In NSec)
100	SUCCESS	3	100	320	200.1	520.1
105	SUCCESS	3	100	320	200.1	520.1
102	SUCCESS	3	102	320	200.1	520.1
107	SUCCESS	3	102	320	200.1	520.1
104	SUCCESS	3	104	320	200.1	520.1
109	SUCCESS	3	104	320	200.1	520.1

The values are also analysed graphically. Firstly the success rate for the responses is been demonstrated [Fig – 2].

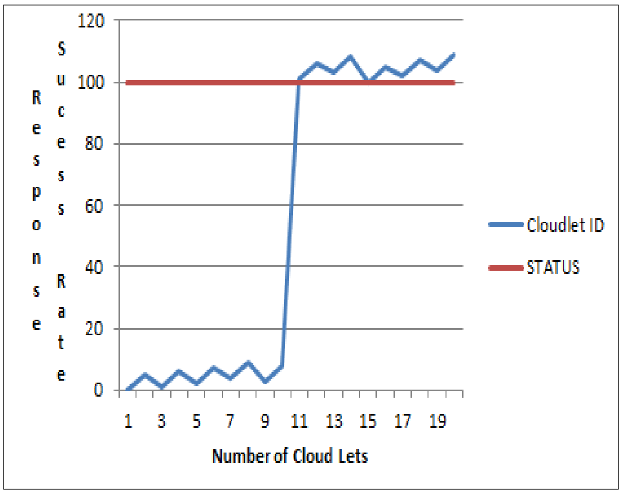

Fig. 2: Response Success Rate Analysis

Secondly, the start and end time analyses of the requests submitted for execution are been demonstrated [Fig – 3]. This analysis demonstrates the responsive nature of the application, which is one of most important criteria for the beneficiaries of the application.

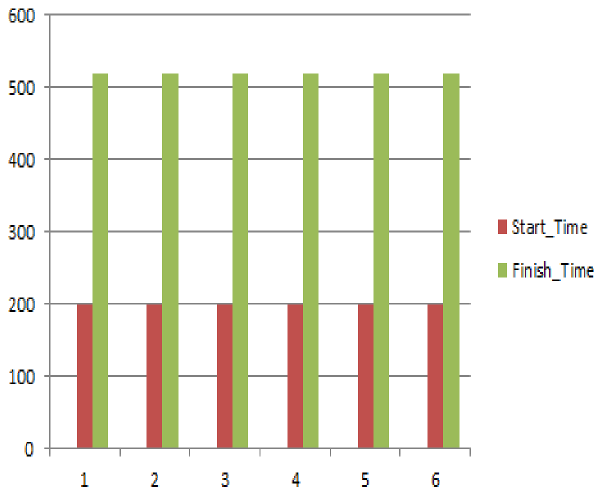

Fig. 3: Execution Time Analysis

Further, the design metric model parameters are been demonstrated. As mentioned a total of 15 random applications are been considered for demonstration [Table – 5].

Table 5: Design Model attribute Valve Analysis

Application Name	Number Of Modules	Independent Modules	Preceding Modules	DB Items	Unique DB_Items	DB Segments	Signle Function Exit_Points	Quality Index
RA1	31	17	23	2334	799	10	14	0.42788
RA2	186	14	174	3556	3147	10	74	0.38079
RA3	110	27	43	3797	119	10	72	0.60074
RA4	125	29	52	3374	2132	10	22	0.52613
RA5	70	37	56	1916	395	10	61	0.40539
RA6	40	19	5	2602	1297	10	0	0.57992
RA7	155	22	112	3334	1954	10	84	0.45539
RA8	48	12	35	2186	653	10	45	0.45022
RA9	198	48	85	2483	2128	10	75	0.45597
RA10	134	12	65	2532	2204	10	63	0.46357
RA11	52	33	10	1927	539	10	16	0.54739
RA12	130	87	120	1499	1379	10	53	0.25611
RA13	56	36	0	2650	488	10	31	0.57886
RA14	168	24	158	3179	793	10	117	0.46349
RA15	186	186	183	3661	1493	10	29	0.30580

In the represented dataset RA denotes any "Random Application" and the arbitrary values are been mentioned. The quality index value is a weighted average of the parameters.
It is observable that ranging from a smaller application to huge applications, the proposed metric models are significantly correct in providing the quality index values. The quality index values are expected to near zero and a higher value towards one, is understood as inappropriate design for cloud based data centre software applications.
The results are also been analysed graphically. Firstly, the module-oriented applications are been analysed with the quality index [Fig – 4].

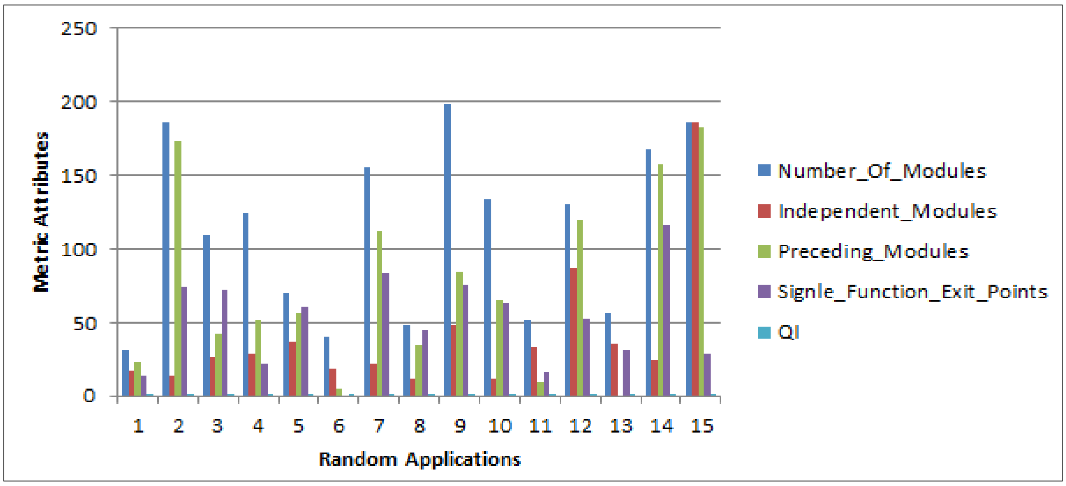

Fig. 4: Modules and QI Analysis

This analysis proves that the applications ranging from 40 modules to 200 modules can be analysed using this metric method. Significantly, this is to be understood that the generic applications ranging from healthcare to insurance to tactical are limited to 150 numbers of modules, which are deployed on the cloud.
Secondly, the analyses of database attributes related to the applications are been analysed [Fig – 5].

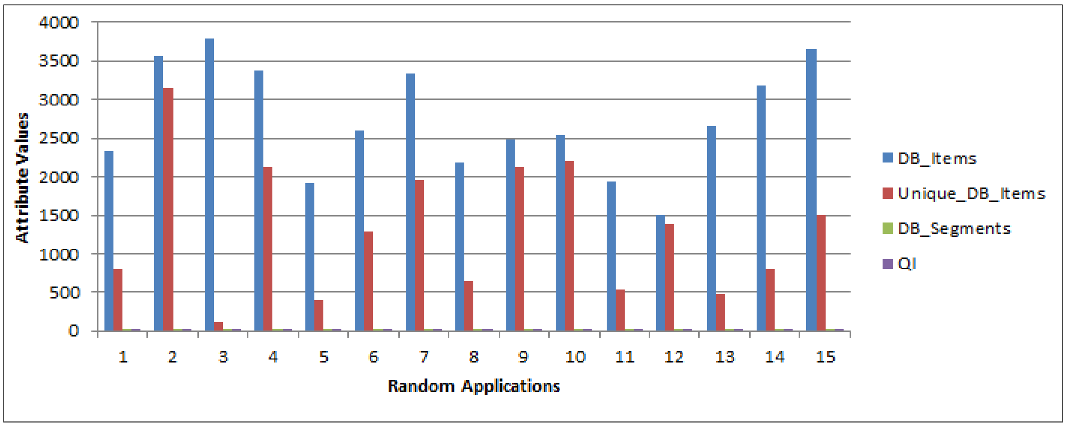

Fig. 5: Modules and QI Analysis

Normal cloud based data centre database configurations are ranging from 1 to 10 segments. It is to be observed that, the proposed metric model can analyse the database segments till 10 segments.

6. Conclusions

Encountering the trade-off between the customer demand and the demand of the software quality improvements, this work analyses the recent improvements in the researches on software quality metrics. During the course of study, this work understands the basic software metrics and highlights the improvement points. Understanding the recent trend of migrating traditional application into the cloud data centres, the proposed novel cloud based data centre oriented software application quality metric model is been demonstrated. The proposed model is consisting of a total of fourteen unique parameters, classified into two categories as architectural model and design model. The proposed novel metric model is also tested on 110 node cloud data centre and 15 random applications. The notable outcome of this work is to provide a generic new trend of analysing applications, which are migrated from traditional environment to the cloud data centre, to match the demand of modern research and customer satisfaction.

References

[1] Norman E. Fenton, 1991, Software Metrics, A Rigorous Approach, Chapman & Hall, London

[2] A Survey of Software Metrics, Fabrizio Riguzzi, July 1996, DEIS Technical Report no. DEIS-LIA-96-010.

[3] A.Mili .et.al. Semantic software metrics.2013.

[4] Bohem, B.W, Brown,j.R. and Lipow, M, “Quantitve evaluation of software quality” proceedings of the second international conference on software engineering, 1976.

[5] Norman E.Fenton, shari Lawerance Pfleeger. Software metrics Arigorous and practical approach. Second edition.PWS Publishing company. 20 park plaza Boston.

[6] Dromy. R.G, “cornering the chimera” , IEEE software ,31(1),33-34,1996.

[7] Z. Zhang, X. Cheng, S. Su, Y. Wang, K. Shuang and Y. Luo "A unified enhanced particle swarm optimization-based virtual network embedding algorithm", Int. J. Commun. Syst., vol. 26, no. 8, pp.1054 -1073 2013

[8] X. Cheng, S. Su, Z. Zhang, K. Shuang, F. Yang, Y. Luo and J. Wang "Virtual network embedding through topology awareness and optimization", Comput. Netw., vol. 56, no. 6, pp.1797 -1813 2012

[9] G. Sun, V. Anand, H. Yu, D. Liao, Y. Cai and L. Li "Adaptive provisioning for evolving virtual network request in cloud-based data centers", Proc. IEEE GLOBECOM, pp.1617 -1622 2012

[10] Christof Ebert and Reiner Dumke. Software Measurement: Establish, Extract, Evaluate, Execute. Springer V erlag, 2007

[11] Norman E. Fenton and Shari Lawrence Pfleeger. Software Metrics: A Rigorous and Practical Approach. PWS Publishing Company, 1997.

[12] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lips Software Metrics SEI Curriculum Module, SEI-CM-12-1.1, December 1988.

[13] A Survey on Metric of Software Complexity Sheng Yu, Shijie Zhou.

[14] Survey on Impact of Software Metrics on Software Quality - (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 1, 2012

[15] McCabe T. A complexity metric. IEEE Transactions on Software Engineering, 1976, 2(4): 308-320.

[16] Torn, A., T. Andersson and K. Enholm, 1999. A complexity metrics

[17] Gall, C. S. Inf. Technol. & Syst. Center, Univ. of Alabama in Huntsville, Huntsville, AL Lukins, Stacy K.; Etzkorn, Letha H.; Gholston, Sampson; Farrington, Phillip A.; Utley, Dawn R.; Fortune, J.; Virani, Shamsnaz Semantic Metrics, Conceptual Metrics, and Ontology Metrics: Volume: 2 , Issue: 1 Page(s): 17 – 26.

[18] Eric S. Raymond, the Art of Unix Programming, Addison-Wesley, New York, 2004.

[19] Kunja Nagamani ; Ch. G. V. N. Prasad ; K. Shahu Chatrapati, A novel framework for optimal component based data center architecture, International Conference on Information Communication and Embedded Systems (ICICES), 2016, IEEE

[20] Helali, Rasha Gaffer M. "Software semantic metrics: A Survey."

[21] Dr. Seetaiah Kilaru, Harikishore K, Sravani T, Anvesh Chowdary L, Balaji T “Review and Analysis of Promising Technologies with Respect to fifth Generation Networks”, 2014 First International Conference on Networks & Soft Computing, ISSN:978-1-4799-3486-7/14,August2014.

[22] S.V.Manikanthan and V.Rama“Optimal Performance Of Key Predistribution Protocol In Wireless Sensor Networks” International Innovative Research Journal of Engineering and Technology ,ISSN NO: 2456-1983,Vol-2,Issue –Special –March 2017.

[23] T. Padmapriya, V.Saminadan, “Performance Improvement in long term Evolution-advanced network using multiple imput multiple output technique”, Journal of Advanced Research in Dynamical and Control Systems, Vol. 9, Sp-6, pp: 990-1010, 2017.

[24] Rajesh, M., and J. M. Gnanasekar. "Path observation-based physical routing protocol for wireless ad hoc networks." International Journal of Wireless and Mobile Computing 11.3(2016): 244-257.