iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** VII **Month of publication:** July 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

Empirical Analysis of Mccall's Quality Factors Using Analytic Hierarchy Process Agile Perception

Dr.V. Padmakar[1], Dr.B.V. Ramana Murthy[2], Jogannagari Malla Reddy[3]
[1]*Professor, Dept. of. CSE, Visvesvaraya College of Engineering and Technology, Hyderabad, Telangana State, India,*
[2]*Professor, Dept. of. CSE, AAR Mahaveer Engineering College, Hyderaba, Telangana State, India,*
[3]*Research Scholar, Dept. of. CSE, Lingaya's University, Faridabad, Haryana State, India*

Abstract: Software Quality is the degree of which a system components and process meets the system requirements, in other words software quality is user satisfaction, in this paper, we are trying to find out weightages of McCall quality factors, which are applied on development process using agile methodology. An empirical analysis has been found out using Analytic Hierarchy Process on McCall factors with reference to agile methodology applications has been carried out. The process of empirical techniques provides the weightages of McCall factors, which provides insights to the work flow, work process required to attain quality standards

Keywords: Quality Assurance, Quality factors, Quality Model, Analytic Hierarchy Process

I. INTRODUCTION

Software quality is the degree of which a system components, process, meets, specified requirements and customer or user needs or expectations requirements [1]. The question, Quality is a complex concept, it means different things to different people and it is highly context dependent. Gravin has analysed how software quality is perceived in different ways in different domains, such as philosophy, economics, marketing, and management. The software quality is difficult to define, it depends on the viewpoint of the observer. Kitchenham and Pfleeger's emphasizes the views in comprehensive manner as follows[13].

Transcendental view: it visualizes quality as something that can be recognized but is difficult to define. The transcendental view is not specific to software quality alone but has been applied in other complex area of everyday life.

User View: It perceives quality as fitness of purpose. According to this view, while evaluating the quality of a product, one must ask the key question " Does the product satisfy user needs and expectations?

Manufacturing View: here quality is understood as conformance to the specification. The quality level of a product is determined by the extent to which the product meets it specifications.

Product View: in this case, quality is viewed is viewed as coupled to the inherent characteristics of the product. A product's inherent characteristic that is internal qualities determines its external qualities.

Value-Based View: quality in this perspective depends on the amount a customer is willing to pay for it.

The connect of software quality and the effort to understand it in terms of measurable qualities date back to the mid -1970. McCall, Richards, and Walters were the first to study the concept of software quality in terms of quality factors and quality criteria

Usually a quality factor represents a behavioural characteristic of a system. Some examples of high level quality factors are correctness, reliability, efficiency, testability, maintainability and reusability. A quality criterion is an attribute of a quality factor[2] that is related to software development. For example, modularity is an attribute of the architecture of a software system[5]. Highly modular software allows designers to put cohesive components in one module thereby improving the maintainability of the systems.

II. SOFTWARE QUALITY FACTORS

A. The software industry have been anxiously targeted for improve the product quality with innovative methods. Software quality plays the vital role in the software development in the various phases of the software development life cycle. The faulty software products has significant cost to the suppliers and unsatisfaction to end-users which fails to meet their goals. Any system behavioural characteristic represent quality factors [13][14]. The Jim McCall developed the model for US Air Force identified with following three main perspective for characterizing the quality attributes of software product of the following.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

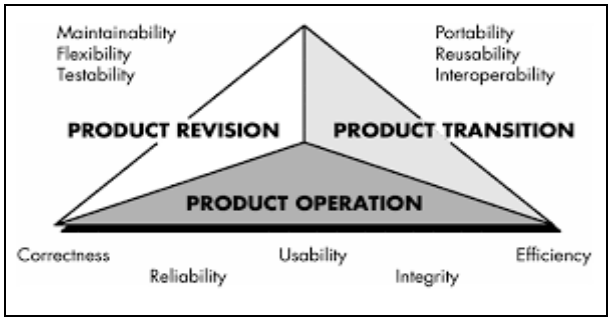


Figure 1. McCall's Quality factors of the software product

1) *Product Operation:* The product operational characteristics shows the operational behaviour of the software product like Correctness, Reliability, Usability, Integrity and Efficiency.
2) *Product Revision:* The product revision is ability to Change, test and maintain the software product as per change management . Example : Testability, Flexibility, Maintainability.
3) *Product Transition:* The product transition have the adaptability and comfortably features with the new environments.. Example. interoperability, portability, and reusability etc

B. Software developers and quality assurance engineering are interested in different quality factors to a different extent. For example customer may want efficient and reliable software with less portability[15]. The developers are required to meet customer needs by plan their system efficient and reliable for the same time making the product portable and reusable to reduce the cost of software development. The software quality assurance team is interested in the testability of a system so that some other factors such as correctness, reliability and efficiency can be easily verified through testing. The testability factor is important to developers and customers as well[9].

1) *Correctness:* **S**oftware systems satisfy all the functional requirements the system is said to be correct. A correct software system may still be unacceptable to customer if the system fails to meet unstated requirements such as stability performance and scalability[7].
2) *Reliability:* It is difficult to construct large software systems which are correct. But the software may still be acceptable to customers because the execution scenarios causing the system to fail may not frequently occur when the system is deployed. Reliability is a customer perception and incorrect software can still be considered to be reliable.
3) *Efficiency*: Efficiency concerns to what extent a software system utilizes resources such as computing power memory disk space communication bandwidth and energy. A software system must utilize as little resources as possible to perform its functionalities[16]. For example by utilizing less communication bandwidth base station in a cellular telephone network can support more users.
4) *Integrity*: A system is integrity refers to its ability to withstand attacks to its security integrity refers to the extent to which access to software or data by unauthorized persons or programs can be controlled. Integrity has assumed a prominent role in today's network based application.
5) *Usability*: A software systems is considered to be usable if human users find it easy to use. User put much stress on the user interface of software systems. Software fails too often no good user interface can keep it in the market.
6) *Maintainability*: In general maintenance refers to the upkeep of products in response to deterioration of their components due to continued use of the products. Maintainability known to how easily and inexpensively the maintenance task can be performed.
7) *Testability*: It is important to be able to verify every requirement both explicitly stated and simply expected. Testability means the ability to verify requirements. At every stage of software development it is necessary to consider the testability aspect of a product.
8) *Flexibility*: Flexibility is reflected in the cost of modifying an operational system. As more and more changes are effected in a system throughout its operational phase subsequent changes may cost more and more. It the initial design is not flexible it is highly likely that subsequent changes are every expensive in order to measure the flexibility of a system[17] and easily one can add a new feature to a system.

9) *Portability:* Portability of a software system refers to how easily it can be adapted to run in a different execution environment. Is a board term encompassing hardware platform operating system distributedness and heterogeneity of the hardware system to name a few. Portability is important for developers because a major adaptation of a system can increase its market potential.
10) *Reusability:* Reusability means if a significant portion of one product can be reused may be with minor modification in another product economically it may not be viable to reuse small components. Reusability saves the cost and time to develop and test the component being reused.
11) *Interoperability*: Interoperability feature can binds different type of environments with the computer networking.

III. ANALYTICAL HIERARCHY PROCESS

The Analytic Hierarchy Process considers a set of evaluation criteria, and a set of alternative options among which the best decision is to be made. It is important to note that, since some of the criteria could be contrasting, it is not true in general that the best option is the one which optimizes each single criterion,[20] rather the one which achieves the most suitable trade-off among the different criteria.

The Analytic Hierarchy Process generates a weight for each evaluation criterion according to the decision maker's pair wise comparisons of the criteria[21]. The higher the weight, the more important the corresponding criterion. Next, for a fixed criterion, the Analytic Hierarchy Process assigns a score to each option according to the decision maker's pair wise comparisons of the options based on that criterion. The higher the score, the better the performance of the option with respect to the considered criterion. Finally, the Analytic Hierarchy Process combines the criteria weights and the options scores, thus determining a global score for each option, and a consequent ranking[22]. The global score for a given option is a weighted sum of the scores it obtained with respect to all the criteria.

A. *Implementation of the Analytic Hierarchy Process*

The Analytic Hierarchy Process can be implemented in simple consecutive steps:

1) Prepare the table
2) Fill the pair wise comparison table
3) Summation of columns
4) Each column Summation
5) Repeat for all other columns.
6) Each entry I divided by summation of columns i
7) Repeat for all entries
8) Summation of rows
9) Summation of rows divided by no of columns equal to weight ages of attributes.
10) Summation of weight ages of attributes equal to 1

TABLE I
PAIR WISE COMPARISON TABLE

Agile	C	Re	E	I	U	M	T	F	P	R	Io
C	1.0	0.3	5.0	0.3	5.0	6.0	0.2	3.0	4.0	4.0	4.0
Re	3.0	1.0	4.0	0.3	0.3	3.0	0.2	5.0	6.0	5.0	5.0
E	0.2	0.3	1.0	0.3	4.0	3.0	0.2	4.0	3.0	3.0	3.0
I	3.0	4.0	4.0	1.0	5.0	3.0	3.0	7.0	6.0	7.0	7.0
U	0.2	0.3	0.3	0.3	1.0	0.3	0.2	3.0	3.0	3.0	3.0
M	0.2	0.3	5.0	0.3	4.0	1.0	0.3	4.0	5.0	4.0	6.0
T	0.2	5.0	5.0	0.3	5.0	3.0	1.0	4.0	6.0	5.0	6.0
F	0.3	0.2	0.3	0.1	0.3	0.3	0.3	1.0	3.0	5.0	5.0
P	0.3	0.2	0.3	0.2	0.3	0.2	0.2	0.3	1.0	3.0	3.0
R	0.3	0.2	0.3	0.1	0.3	0.3	0.2	0.2	0.3	1.0	0.3
Io	0.3	0.2	0.3	0.1	0.3	0.2	0.2	0.2	0.3	3.0	1.0

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

TABLE II
PAIR WISE COMPARISON TABLE AND SUMMATION OF COLUMN

Agile	C	Re	E	I	U	M	T	F	P	R	Io
C	1.0	0.3	5.0	0.3	5.0	6.0	0.2	3.0	4.0	4.0	4.0
Re	3.0	1.0	4.0	0.3	0.3	3.0	0.2	5.0	6.0	5.0	5.0
E	0.2	0.3	1.0	0.3	4.0	3.0	0.2	4.0	3.0	3.0	3.0
I	3.0	4.0	4.0	1.0	5.0	3.0	3.0	7.0	6.0	7.0	7.0
U	0.2	0.3	0.3	0.3	1.0	0.3	0.2	3.0	3.0	3.0	3.0
M	0.2	0.3	5.0	0.3	4.0	1.0	0.3	4.0	5.0	4.0	6.0
T	0.2	5.0	5.0	0.3	5.0	3.0	1.0	4.0	6.0	5.0	6.0
F	0.3	0.2	0.3	0.1	0.3	0.3	0.3	1.0	3.0	5.0	5.0
P	0.3	0.2	0.3	0.2	0.3	0.2	0.2	0.3	1.0	3.0	3.0
R	0.3	0.2	0.3	0.1	0.3	0.3	0.2	0.2	0.3	1.0	0.3
Io	0.3	0.2	0.3	0.1	0.3	0.2	0.2	0.2	0.3	3.0	1.0
∑ Columns	8.9	11.9	25.6	3.3	25.6	20.1	5.9	31.7	37.7	43.0	43.3

TABLE III
SUMMATION OF ROWS

	C	R	E	I	U	M	T	F	P	R	Io	∑ Row
C	0.11	0.03	0.20	0.10	0.20	0.30	0.03	0.09	0.11	0.09	0.09	1.35
Re	0.34	0.08	0.16	0.07	0.01	0.15	0.03	0.16	0.16	0.12	0.12	1.40
E	0.02	0.02	0.04	0.07	0.16	0.15	0.03	0.13	0.08	0.07	0.07	0.84
I	0.34	0.34	0.16	0.30	0.20	0.15	0.51	0.22	0.16	0.16	0.16	2.69
U	0.02	0.02	0.01	0.07	0.04	0.01	0.03	0.09	0.08	0.07	0.07	0.53
M	0.02	0.03	0.20	0.10	0.16	0.05	0.06	0.13	0.13	0.09	0.14	1.09
T	0.02	0.42	0.20	0.10	0.20	0.15	0.17	0.13	0.16	0.12	0.14	1.79
F	0.04	0.02	0.01	0.04	0.01	0.01	0.04	0.03	0.08	0.12	0.12	0.52
P	0.03	0.01	0.01	0.05	0.01	0.01	0.03	0.01	0.03	0.07	0.07	0.33
R	0.03	0.02	0.01	0.04	0.01	0.01	0.03	0.01	0.01	0.02	0.01	0.21
Io	0.03	0.02	0.01	0.04	0.01	0.01	0.03	0.01	0.01	0.07	0.02	0.26

TABLE IV
WEIGHTAGES OF ATTRIBUTES

	C	Re	E	I	U	M	T	F	P	R	Io	∑ Row	Weightages = ∑Rows/no of attributes
C	0.11	0.03	0.20	0.10	0.20	0.30	0.03	0.09	0.11	0.09	0.09	1.35	0.12
Re	0.34	0.08	0.16	0.07	0.01	0.15	0.03	0.16	0.16	0.12	0.12	1.40	0.13
E	0.02	0.02	0.04	0.07	0.16	0.15	0.03	0.13	0.08	0.07	0.07	0.84	0.08
I	0.34	0.34	0.16	0.30	0.20	0.15	0.51	0.22	0.16	0.16	0.16	2.69	0.24
U	0.02	0.02	0.01	0.07	0.04	0.01	0.03	0.09	0.08	0.07	0.07	0.53	0.05
M	0.02	0.03	0.20	0.10	0.16	0.05	0.06	0.13	0.13	0.09	0.14	1.09	0.10
T	0.02	0.42	0.20	0.10	0.20	0.15	0.17	0.13	0.16	0.12	0.14	1.79	0.16
F	0.04	0.02	0.01	0.04	0.01	0.01	0.04	0.03	0.08	0.12	0.12	0.52	0.05
P	0.03	0.01	0.01	0.05	0.01	0.01	0.03	0.01	0.03	0.07	0.07	0.33	0.03
R	0.03	0.02	0.01	0.04	0.01	0.01	0.03	0.01	0.01	0.02	0.01	0.21	0.02
Io	0.03	0.02	0.01	0.04	0.01	0.01	0.03	0.01	0.01	0.07	0.02	0.26	0.02
													1.00

B. Analysis of Results

From the above table it has been determined that the integrity have much significance of 24% followed by testability(16%), Reliability(13%), Correctness(12%), Maintainability(10%), Efficiency(8%) and portability(3%). The remaining quality factors usability and flexibility occupies the equal significance of 5% similarly other quality factors reusability and interoperability with less significance of 2% in the product.

C. Interpretation of Results

The results shows the pair wise comparison of attributes using AHP which gives sound and strong empirical relation to map into real world, the weightages given the insights of the business objection work flows to be carried out while developing application using agile method and to attain quality standards, focus of work weightages in terms of product perceptive must be given in the following order to attain McCall Factors.

TABLE IV
WEIGHTAGES OF ATTRIBUTES IN ORDER OF SEQUENCE WITH PRIORITY IN PERCENTAGE

Sl. No	Quality factor	Significance	Weightages in %
1	Integrity	0.24	24
2	Testability	0.16	16
3	Reliability	0.13	13
4	Correctness	0.12	12
5	Maintainability	0.10	10
6	Efficiency	0.08	8
7	Usability	0.05	5
8	Flexibility	0.05	5
9	Portability	0.03	3
10	Reusability	0.02	2
11	Interoperability	0.02	2

From the research findings concludes that, the quality significance in the software product can be visualize with the following bar graph figure.2

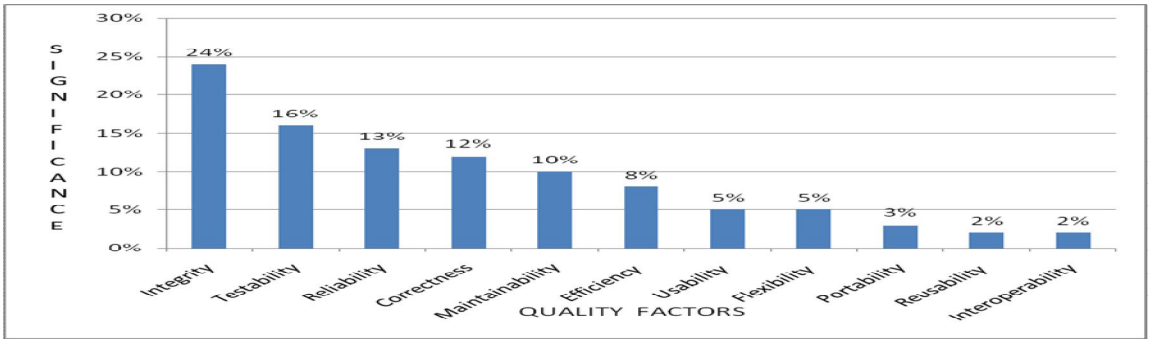


Fig 2 Quality significance in the software product

IV. CONCLUSIONS AND FUTURE SCOPE

An prototype of one sample taken from industry project manager, the process can be consolidation and further research can be done by framing a questioner and getting sample from medium scale industry. Quality is always depends on the design of the product. Any product is designed it is very important to develop to have primary knowledge of factors which will affect the quality of the product. The paper is the study and comparison of quality models as it tells the factors and criterion. Along with this AHP is to the developer for developing a quality product. The Analytical Hierarchy process is effective decision making method for multi criteria

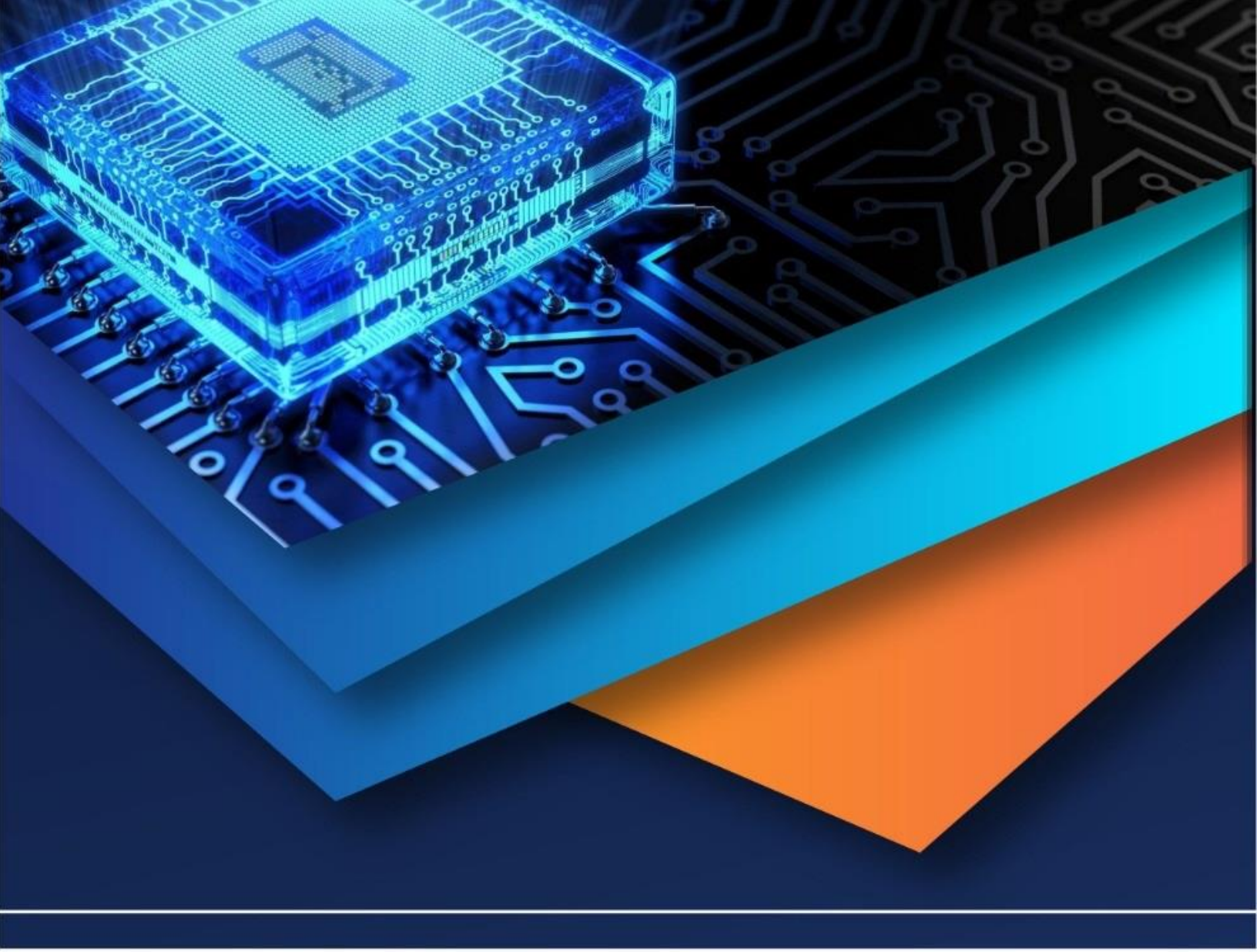
problems.. There are many inherent complexities in the prioritization. With the numerical example. the authors suggest that when some alternatives to be very close to other with sensitive. The decision maker needs to very cautious in prioritization. There is a need of extensive research in area of Multi Criteria Decision Making methods which is useful for scientific and Software engineering applications.

V. ACKNOWLEDGEMENT

We sincerely acknowledge to all the faculty of the department of Computer Science & Engineering, Mahaveer group of Institutions, Hyderabad for their support, guidance, and cooperation during the period of study

REFERENCES

[1] Software Engineering - Product Quality, Part 3, 4, Version 1.0 (ISO/IEC TR 9126), url:http://www.iso.org/iso/home/store /catalogue_tc/, visited:7th July, 2012 .
[2] Al-Qutaish, R. E., "Qualit y Models in Software Engineering Literature: An Analytical and Comparative Study," Journal of American Science , 2010. Vol. 6, no. 3, pp. 166 - 175.
[3] Kitchenham, B., Pfleeger, S. L. "Software Quality: the Elusive Target,"1996. IEEE Software , vol. 13, no. pp. 12-21.
[4] Wirth, N. "A Brief History of Software Engineering,"IEEE Annals of the History of Computing, 2008. Vol. 30 no.3, pp. 32- 39.
[5] Boehm, B. W.; Brown, J. R.; Kaspar, H.; Lipow, M.; McLeod, G.; and Merritt, M., "Characteristics of Software Quality, "North Holland Publishing, Amsterdam, The Netherlands, 1978, vol., no., pp.
[6] Deissenboeck, F.; Juergens, E.; Lochmann, K.; and Wagner, S. 2009. Software quality models: purposes, usage scenarios and requirements. In Proceedings of the 7th ICSE conference on Software quality (Munich, Germany, 2009).
[7] Dromey, R. G., "A Model for Software Product Quality," IEEE Transactions on Software Engineering, 1995, vol. 21, no., pp 146 - 162.
[8] History of Software Engineering. In Wikipedia . Retrieved June 27 2012, from http://en.wikipedia.org/wiki/History_of_software_engineering.
[9] Software Engineering - Product Quality, Part 3, 4, Version 1.0 (ISO/IEC TR 9126), url:http://www.iso.org/iso/home/store/ catalogue_tc/, visited: 7th July, 2012
[10] Bawane Neelam and Srikrishna, C.V. 2010. A Novel Method for Quantitative Assessment of Software Quality. International Journal of Computer Science and Security 3, 6, 508 - 517
[11] Alvaro Alexandre, E.S. de Almeida and S.R. deLemos Meira 2010. A Software Component Quality Framework. ACM SIG SOFT SEN 35, 1(Nov. 2010), 1 – 18.
[12] Kalaimangal Sivamuni and Srinivasan Rengaramanujam 2011. Q'Facto12 - An improved quality model for COTS Components. ACM SIGSOFT SEN 35, 2(Mar. 2010), 1 – 4.
[13] Upadhyay Nitin, Despande, Bharat M. and Agrawal, Vishnu P. 2011. Towards a Software Component Quality Model . Springer Press CCIS ACST .131, 3 , 398 – 412 DO I: 10.1007/978 -3 – 642 – 17857 - 3_40
[14] Bassem Anas AL-Badareen 2011. Software Quality Evaluation: User's View. International Journal of applied mathematics and informatics 5, 3(2011), 200-207
[15] Bawane, Neelam and Srilrishna, C.V, "A Novel Method for Quantitative Assessment Of software Quality", international Journal of Computer Science and Security, Vol 3,Issue 6,2011.
[16] Wu,B.H "An evolutionary approach to evaluate quality of software systems", 2011
[17] A. P. Mathur, Foundations of Software Testing, Second. Pearson, 2013.
[18] P. R. Suri and H. Singhani, "Object Oriented Software Testability Survey at Designing and Implementation Phase," International Journal of Science and Research, vol. 4, no. 4, pp. 3047– 3053, 2015.
[19] P. R. Suri and H. Singhani, "Object Oriented Software Testability (OOSTe) Metrics Analysis," International Journal of Computer Applications Technology and Research , vol. 4, no. 5, pp. 359–367, 2015.
[20] . Khanna, "Testability of Object- Oriented Systems : An AHP -Based Approach for Prioritization of Metrics," in International Conference on Contemporary Computing and Informatics(IC3I) , 2014, pp. 273 – 281.
[21]] P. K. Singh, O. P. Sangwan, A. Pratap, and A. P. Singh, "Testability Assessment of Aspect Oriented Software Using Multicriteria Decision Making Approaches, World Applied Sciences Journal, vol. 32, no. 4, pp. 718–730, 2014.
[22]] P. K. Singh, O. P. Sangwan, A. P. Singh, and A. Pratap, "A Quantitative Evaluation of Reusability for Aspect Oriented Software using Multi-criteria Decision Making Approach," World Applied Sciences Journal, vol. 30, no. 12, pp. 1966–1976, 2014.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY