

# IMPLEMENTING THE MODEL FOR SOFTWARE QUALITY BASED ON INTERACTION BETWEEN USER AND DEVELOPER

Dr.S.V.N. Sreenivasu<sup>1</sup>, Dr. Nilamadhab Mishra<sup>2</sup>, Shafiqul Abidin<sup>3</sup>, Dr. Nripendra Narayan Das<sup>4</sup>,  
Dr.P. Sivakumar Ponnusamy<sup>5</sup>

<sup>1</sup>Professor, Department of Computer Science and Engineering, Narasaraopeta Engineering College (Autonomous), Narasaraopeta, Andhra Pradesh, India. E-mail: drsvnsrinivasu@gmail.com

<sup>2</sup>Assistant Professor (Sr.), School of Computing Science and Engineering, VIT Bhopal University, Madhya Pradesh, India. E-mail: nmmishra77@gmail.com

<sup>3</sup>HMRITM, (GGSIPU), Delhi, India. E-mail: shafiqulabidin@yahoo.co.in

<sup>4</sup>Associate Professor, Department of Information Technology, Manipal University Jaipur, Rajasthan, India. E-mail: nripendranarayan.das@jaipur.manipal.edu

<sup>5</sup>Associate Professor, Department of Information Technology, Sree Vidyanikethan Engineering College, Tirupathi, Andhra Pradesh, Indian. E-mail: drsivakumar.p@gmail.com

Received: 23.04.2020

Revised: 24.05.2020

Accepted: 21.06.2020

---

**ABSTRACT:** The present research paper proposes the software metrics have a direct link with measurement in software engineering. Here software engineering is not an exemption for measuring correct and suitable conditions as any other engineering fields require such as size, manual examinations and complicated nature of the increasing software so that it makes the software a harder assignment. Most of the engineers worry about the software measures and to enhance the quality of the software. The present paper studies about the asses and software analysis metric used for the measurement of product and process in software. In present study the researcher collects the literature depending on various electronic databases which are available since 2008 for understanding the software metrics and finally identifies the quality of software that which is used as a method for measuring the software designing and configurations. It measures the percentage of variables such as superiority, correctness, quality scalability and completeness of the item and also includes the absence of bugs which are high standard that might be used for quitting and offering one associated unique relation. The reason for that relation is to have better software measurement for measuring the quality of the software that is affected by current which is the most common software metric tool to decrease the partial faults during valuation of software quality. The main reason of this study is an identification of software metric around and illustrates the development in present field for critical investigation about the key metrics initiated on both user and developer integration with unified software quality management on user and developer (SQMUD).

**KEYWORDS:** Software metrics, Software testing and faults, Quality of software, Software quality management on user and developer (SQMUD).

---

© 2020 by Advance Scientific Research. This is an open-access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>) DOI: <http://dx.doi.org/10.31838/jcr.07.14.205>

## I. INTRODUCTION

Correct measurement is the prior condition in several engineering fields, and software engineering is not an exemption. Software metrics require a direct link with measurement in software engineering. According to De Marco; "You can't manage what you can't measure!" [5] And Campbell likewise gives significance of extent in software engineering by setting "If you aren't measuring, you aren't managing" — you're only along for the ride [1]. Software metrics will reduce the subjectivity of faults during the assessment of software quality and it provides a measurable foundation for creating choices round the software quality. Metrics are the numerical value of software and it is used to predict the fault [11]. Software metrics occur file level, class-level, component-level, method-level, process-level and quantitative values-level metrics [8], this helps the project manager and software engineers to find defects and making the prevention method for the defect. Software metrics can be applied to each software development phase. During requirement analysis software metrics can be developed, for instance, in order to determine cost estimation and resource needed. At the time of system design

we also develop metrics in order to count function point. Metrics applied at implementation phase are also used to measure software size [7]. According to Vikas Verma, having software metrics have a number of benefits such as provide a foundation for approximation and simplifies preparation by means of controlling status reporting, identifying risk areas and effectiveness and efficiency of testing[14]. Measuring the software project has a number of benefits for company it saves development effort, time and money. In addition to this for complex projects using metrics have easy to understand, identify common problems early, and manage resources [6].As mentioned above even if it has the benefit there is also drawbacks that are better understanding (knowledge) and need a lot of effort and time. Software metrics empower software developers to investigate their code and make upgrades assuming that required. Metrics could be developed for software size, cost estimation, software quality, maintainability, deformity analysis and software testing.

## **II. STATE OF PROBLEMS**

Software defect is not only the quality of software, but also increase costs and suspend the development schedule. In addition to above, there could be many reasons for the system to be faulty; most of them are because of the human factor, mistake and errors made in designing or coding by the peoples, data entry, documentation and communication failures. In order to solve the most common human factors we shall use software fault prediction before us taking the software systems to test and maintains. Software testing is one of the most those the majority basic and unreason ability stages from claiming product development. Venture supervisors compelling reason to know “when should stop testing?” and “which parts of the code on test?”. Those replies to these inquiries might specifically influence abandon rates and item personal satisfaction and in addition asset allotment (i. E. Experience from claiming test staff, know what number of individuals to dispense to testing) and the expense. Likewise that size And unpredictability about product increases, manual review about software gets a harder assignment. In this context, elective strategies that need aid used to anticipate possibility impacts unmistakably need aid product abandon prediction. Since trying commonly expends 40 - half from claiming development efforts, and expends additional exert for frameworks that oblige higher levels of reliability, it is a noteworthy and only the software engineering. [2] [3]. Software fault predictions used to solve the problem that reduces the qualities software product. In this paper, we studied the software tester to predict and fix the bug before it is delivered to customers that assures the quality of software and evaluate the state of art of software metrics; specifically focus on the quality metrics to measure software’s product and process.

### **i) Objective**

The primary objective of this study was to asses and analysis’s software metrics used to measure software quality particularly software product and process with respect to user interaction.

## **III.LITERATURE REVIEW**

The first survey on software metrics was done by Kafura in 1985 and he suggests existing code metrics, complexity metrics and validation metrics. Generally, in this survey work presented the major relations exist among the software metrics and quality aspects like comprehensibility of code, error features, length of coding time, and structural soundness [4].

According to Ming Chang et al [9] discussed the role of software metrics and software measurement for software quality. Authors also classified the software metrics according to various manners which are commercial, important, observation, measurement and software development. In addition to this, the author also discussed various methodologies which are around 15 measurement methodologies and 24 types of testing with their definitions, formula and effects.

Poornima Gupta et al [10] presented the software fault prediction using artificial intelligence methods and this research work focused on related work on software metrics particularly on AI approaches and software metrics.

Kunal Chopra et al [13] discussed about software metrics complexity using N depend to measure software product like size metrics, control flow metrics and data flow metrics. The final contribution by researcher was introducing the most common known and utilized software metrics projected and evaluate the use of software metrics in creating simulations of software expansion procedure.

## **IV.IMPLEMENTATION**

The SQMUD May be a screening procedure which may be used to guarantee the quality on whole software development lifecycle methodology. It will be likewise a nonstop evaluation system which facilitates specific methods for task development with particular guidelines alongside documentation. The methods if be used to guarantee personal satisfaction Conclusion (zero defects) Also venture victory. Toward a secondary level, the capacity from claiming SQMUD may be with perform those following:

Software undertaking planning: nature polishes ought to a chance to be arranged ahead of time which camwood make actualized further.

Client requirement: necessities ought to further bolstering a chance to be checked done whole task development transform to full fill client necessities.

Plan procedure & Coding: sure methodologies need aid emulated for configuration transform. Coding standard and rules must a chance to be made and actualized.

Software coordination and Testing: product joining and trying if be arranged also aggregated likewise for every prerequisite.

Direct irregular and planned audits: perform SQMUD audits to guarantee those fundamental controls are set up. The SQMUD procedure comprises of a mixed bag for stages for particular exercises. These exercises ought further bolstering make performed Toward a SQMUD less group which is answerable for product quality certification planning, analysis, And reporting weight. SQMUD is more successful the point when it reports dependent upon through a differentiate administration less group thereabouts they can stay dedicated of the procedure and remain objective of the deliverable. Those responsibilities of the SQMUD less group incorporate survey from claiming documentation to culmination and adherence to standards support previously, inspections, Audit for test results, also occasional audits for controls.

Past exploration fill in • the suggested examine includes a moved forward rendition about. AZ-Model then afterward acquiring that assessment for masters and acquiring smoothness after best possible usage for Different sizes for undertakings done associations with diverse sizes. And, measurable analyses were performed should. Analyse the noteworthiness of the recommended AZ-Model.

Recommended research worth of effort.

Those suggested study includes a moved forward versify from claiming. AZ-Model after getting that assessment for masters and acquiring dexterity then afterward fitting execution for Different sizes about undertakings clinched alongside associations with distinctive sizes. And, decision tree analyses were performed on. Inspect the noteworthiness of the suggested AZ-Model.

Product measurements and dependability product development will be an intricate also confounded methodology for which product faults would embed under that code toward mistakes throughout the advancement methodology or upkeep. It need been indicated that those example of the fault's insertion phenomena is identified with measurable qualities of the software, particularly for the product measurements. For example, an expansive software framework comprises for Different modules And every about these modules might make described as far as quality measures – it might remain calm helpful should have the ability with build “dangerous module” prediction models In view of these measurable qualities.

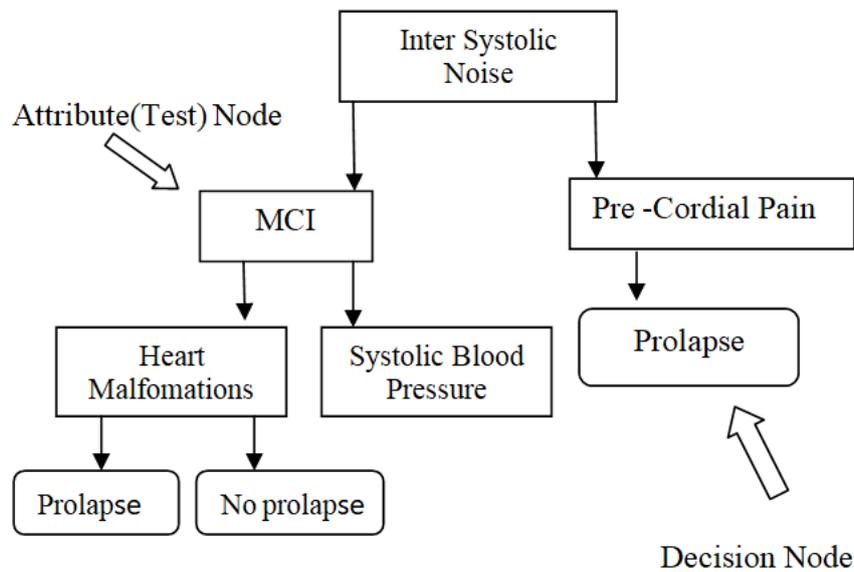


Fig.1: Prediction Models in View of these Measurable Qualities

**i) Entropy**

Setting together a choice tree may know a matter about picking which quality with test toward each hub in the tree. A measure known as majority of the data increase which will a chance to be used to choose which quality should test during each hub may be characterized.

It will be recognized that entropy will be a measure of the pollution on an accumulation from claiming preparation sets. Data get will be itself computed utilizing a measure known as entropy, which will be primary characterized on account of a double choice issue et cetera characterized for those general situation. Provided for a double categorization, C, Also An situated about examples, S, for which those extent from claiming cases sorted Similarly as certain By c will be p+ and the extent for samples sorted as negative By c will be p-, that point those entropy for encountered with urban decay because of deindustrialization, innovation developed, government entropy will be.:

$$Entropy(s) = -p + \log_2(p_+) - p - \log_2(p_-) \longrightarrow (1)$$

There is an issue from claiming attempting should figure out that best quality to pick for a specific hub done a tree. The taking after measure calculates a numerical worth for a provided for attribute, A, with admiration to a set of examples, encountered with urban decay because of deindustrialization, engineering concocted, government entropy.

Note that the qualities from claiming quality and will reach through An set of possibilities known as the values (A), And that, for An specific esteem starting with that set, v, it may be composed as Sv for those situated for illustrations which bring esteem v to quality a. That majority of the data pick up of quality A, relative should an accumulation of examples, S, will be computed Likewise:

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \longrightarrow (2)$$

$$Split\ Information(S,A) = - \sum_{i=1}^n \frac{|s_i|}{|s|} \log_2 \frac{|s_i|}{|s|} \longrightarrow (3)$$

and

$$Gain\ Ratio(S,A) = \frac{Gain(S,A)}{Split\ Information(S,A)} \longrightarrow (4)$$

**ii) Sqmud Process**

SQMUD Methods give certification all around the undertaking or product management Anddevelopment lifecycle. The consolidated project management life cycle (PMLC) and system development life cycle (SDLC)

comprises from claiming eight exceptional periods – initiation, planning, analysis, design, development, testing, implementation, And shutting. Every SQMUD life cycle stage holds an sentiment circle which gives majority of the data in regards issues discovered Throughout SQMUD exercises and ensures change.

Project Testing: The SQMUD less group if assess the execution for unit testing. A standout amongst the greatest dangers may be creating an item that doesn't meet the expressed necessities alternately creating an item full for defects.

The advancement controls also certification exercises would essential with relieve these dangers Also recognize defects early and the dangers connected with software advancement are reduced. The objective from claiming testing done SDLC is will discover also record defects.

The SQMUD group if Audit that usage arrange alongside those transformed management and guarantee that trying of the product or project throughout development stage may be finished also acceptable of the clients and stakeholders.

**Table 1:** Respondent Information

Section-A1 (Respondent Information)		
Full name (Optional)	Job Title	
	Position	
Have you ever been participated in an software development methodology	Yes <input type="checkbox"/>	No <input type="checkbox"/>
Working Experience in software development organization		
What is the scope of your company?	Client <input type="checkbox"/>	Vendor <input type="checkbox"/>
	Not Sure	other
E Mail Address		
Current address of your organization and Country		
How many years of industry/academic experience do you have		
Have you ever participated software development life cycle improvement project?	Yes <input type="checkbox"/>	No <input type="checkbox"/>
Sections A 2 (Organization Details)		
Name of Organization (optional)		
What is the primary business function of your organization?	Collocated software development <input type="checkbox"/>	Global software development <input type="checkbox"/>
	Research <input type="checkbox"/>	other <input type="checkbox"/>
Please specify the size of your organization	Small <input type="checkbox"/>	Medium <input type="checkbox"/>
	Large <input type="checkbox"/>	Not sure <input type="checkbox"/>
Please specify the number of employees	Less than 50 <input type="checkbox"/>	51-100 <input type="checkbox"/>
	101-150 <input type="checkbox"/>	Greater than150 <input type="checkbox"/>
Please specify the type of your organization	National <input type="checkbox"/>	Multinational <input type="checkbox"/>
	Not sure <input type="checkbox"/>	other <input type="checkbox"/>
Does your organization adopted software development life cycle process improvement standard or (CMMI/ISO)	CMMILevel-1 <input type="checkbox"/>	CMMILevel-2 <input type="checkbox"/>
	CMMILevel-3 <input type="checkbox"/>	CMMILevel-4 <input type="checkbox"/>
	CMMILevel-5 <input type="checkbox"/>	ISO <input type="checkbox"/>
	Other <input type="checkbox"/>	Not sure <input type="checkbox"/>
Which software development methodology your organization adopted		
Which factor is most important for you as a professional when adopting a method	Low cost <input type="checkbox"/>	Easy to handle <input type="checkbox"/>
	Great Productivity <input type="checkbox"/>	Great Reliability <input type="checkbox"/>

**V. CONCLUSION**

In software development, software testing is highly desirable to assure the quality of the software product. Software testing performed via manual and software metrics, the former one (manual) is costly and it required high time interval to perform it because of it now a day software engineer moves to systematic measurement method which is software metrics. This study conducted to reveal to asses and analysis's software metrics used to measure software quality particularly software product and process. Software metrics utilized to extent the software product and process. The researcher used a collection of literatures from various electronic databases which available since 2008 to understand and know the software metrics; the researcher has been identified Product personal satisfaction may be a method for measuring how product is intended what's more entryway great the software conforms to that configuration. Exactly of the variables that we need aid searching for software quality would Correctness, result quality, Scalability, culmination And nonattendance of bugs, In any case the quality standard that might have been utilized from one association will be unique in relation to others for this reason it will be better will apply those product measurements will measure the nature of product and the current most common software metrics tools. In the future the researcher recommends the specific application area of each software metrics and how can perform by the researcher to enhance the quality of software applications.

## VI. REFERENCES

- [1] Campbell, Luke, and B. K, "Software Metrics: Adding Engineering Rigor to a Currently Ephemeral Process " 1995.
- [2] Chayanika S, Sangeeta S, and R. S, "A Survey on Software Testing Techniques using Genetic Algorithm," *International Journal of Computer Science* vol. 10, pp. 381-393, January 2013.
- [3] C. Z. a. Q. SHICHAOZHANG, "Data Preparation for Data Mining" *Applied Artificial Intelligence* vol. 17, pp. 375–381 2003.
- [4] D. Kafura, "A survey of software metrics " presented at the ACM annual conference on The range of computing New York 1985
- [5] D. a. Tom, Controlling Software Projects. *New York: Yourdon Press*, 1986.
- [6] Fernando, Wijayarathne, M. D. Fernando, and I. Guruge, "The Importance of Software Metrics: Perspective of A Software Development Projects In Sri Lanka," *SAITM Research Symposium on Engineering Advancements*, 2014.
- [7] H. R. Bhatti, "Automatic Measurement of Source Code Complexity," *MASTER'S THESIS Computer Science and Engineering Luleå University of Technology*.
- [8] Malkit S and D. S, "Software Defect Prediction Tool based on Neural Network" *International Journal of Computer Applications*, vol. 70 2013.
- [9] Ming Chang Lee and T. Chang, "Software Measurement and Software Metrics in Software Quality," *International Journal of Software Engineering and Its Applications*, vol. 7, 2013
- [10] Poornima Gupta and P. Sahai, "A Review on Artificial intelligence Approach on Prediction of Software Defects," *International Journal of Research and Development in Applied Science and Engineering* vol. 9, February 2016.
- [11] Pooja P and D. A. Phalke, "Survey on Software Defect Prediction Using Machine Learning Techniques," *International Journal of Science and Research*, vol. 3, December 2014
- [12] S. Shivaji, "Efficient Bug Prediction and Fix Suggestions" *PhD A dissertation UNIVERSITY OF CALIFORNIA* 2013.
- [13] S. D. Conte, a. V. Y. Shen, and W. M. Zage, "A Software Metrics Survey" in Technical Reports P. U. C. Science, Ed., ed, 1987.
- [14] Vikas V and S. M, "Applications of Software Testing Metrics In Constructing Models Of The Software Development Process " *Journal of Global Research in Computer Science*, vol. 2 pp. 96-98 May 2011.