An Efficient Model for Software Quality Analysis Based on User and Developer Interaction

D.Naga Malleswari, D.Rakesh, K.Subrahmanyam, Divya Vadlamudi

Abstract: Software metrics have a direct link with measurement in software engineering. Correct measurement is the prior condition in any engineering fields, and software engineering may be an exemption, as those size and complicated nature of software increases, manual examination of software becomes a harder assignment. Most Software Engineers worry about the quality of software, how to measure and enhance its quality. The overall objective of this study was to assess and analysis software metrics used to measure the software product and process. In this study, the researcher used a collection of literatures from various electronic databases, available since 2008 to understand and know the software metrics. Finally, in this study, the researcher has been identified software quality will be a method for measuring how software is designed and how well the software conforms to that configuration. A percentage of the variables that we would be searching for software superiority and Correctness, item quality, Scalability, completeness and absence of bugs of those quality standard that might have been utilized from you quit offering on that one association will be unique in relation to others for this reason it may be better to apply the software measurements to measure the quality of software and the current is most common software metrics tools to decrease the partiality of faults during the valuation of software quality. The central influence of this study is an indication around software metrics to illustrate for development in this field by critical investigation about key metrics initiated on both developer and user interactiona unified definition of software quality management on User and Developer (SQMUD) is proposed

Keywords: Quality of software, software testing and faults, software metrics, software quality management on user and developer (SQMUD)

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!” [18] 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 [19]. Software metrics will reduce the subjectivity of faults during the assessment of software quality and it provides a measurable foundation for creating choices around the software quality. Metrics are the numerical value of software and it is used to predict the fault [3]. Software metrics occur file-level, class-level, component-level, method-level, process-level and quantitative values-level metrics [4], 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 designing we also develop metrics in order to count function point. Metrics applied at implementation phase are also used to measure software size [5]. 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 [23]. 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 [7]. 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 [5].

II. 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 [28]. According to Ming Chang et al [29] 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...
An Efficient Model for Software Quality Analysis Based on User and Developer Intraction

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 [30] presented the software fault prediction using artificial intelligence methods and research work focused on related work on software metrics particularly on AI approaches and software metrics.

Kunal Chopra et al [31] discussed about software metrics complexity using Ndepend to measure software product like size metrics, control flow metrics and data flow metrics. The final contribution by researcher was introducing the most commonly known and utilized software metrics projected and evaluatethe use of software metrics in creating simulations of software expansion procedure.

III. EXISTING WORK

• the proposed study involves an improved version of AZ-Model after obtaining the opinion of experts and obtaining expertise after proper implementation for various sizes of projects in organizations with different sizes. • Furthermore, statistical analyses are performed to examine the significance of AZ-Model.

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 a nonstop evaluation system which facilitates specific methods for task development with particular guidelines alongside documentation. The methods if be used to guanantee 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 made actualized further.

• Client requirement: necessities ought further bolstering a chance to be checked done whole task development transform to fulfill the 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 An differentiate administration less group thereabouts they canstay 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 those 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 embedded under those 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.

4.1 ENTROPY

Setting together a choice tree may be knownmas 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 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 situations. 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
There is an issue from claiming attempting should figure out those 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 a will reach through a set of possibilities known as the values (A), And that, for a 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. Those majority of the data pick up of quality A, relative should an accumulation of examples, S, will be computed Likewise:

\[\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)\]  

\[\text{Split Information}(S, A) = -\sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \log_2 \left( \frac{|S_v|}{|S|} \right)\]  

\[\text{Gain Ratio}(S, A) = \frac{\text{Gain}(S, A)}{\text{Split Information}(S, A)}\]

4.2 SQMUD PROCESS

SQMUD Methods give certification all around the undertaking or product management and development 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 a 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 those 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.
### V. RESULTS

**User Interaction**
- Frequency: 1.9883809546482875
- Entropy: [1, 1, 1, 3, 1]

**Developer Interaction**
- Entropy: [0.9842809546482875]
- User Entropy: 1.9883809546482875

**Graph**

- Graph details:
  - User Name: abcd
  - Job Name: efg
  - Organization Name: jkl
  - Frequency: 0.9883809546482875
  - Entropy: [1, 1, 1, 3, 1]
VI. CONCLUSION

In software development, software testing is highly desirable to ensure 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 assess 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 softwarequality would Correctness, result quality, Scalability, culmination And nonadherence 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.

REFERENCES

An Efficient Model for Software Quality Analysis Based on User and Developer Interaction


AUTHORS PROFILE

D. Naga Malleswari is a research scholar in Computer Science and Engineering of of Koneru Lakshmaiah Education Foundation, Vaddeswaram, India. She is a First class M.Tech graduate from JNTU, Kakinada. She has published 10 Scopus indexed papers and her research interest in Software Engineering, Risk Management.

D. Rakesh is a 4th year student in CSE as part of his project work he had studied many papers and done a project related to software implementation efficiency for this he surveyed many papers as part of literature and done the project successfully.

Dr. Kodukula Subrahmanyam, a Gold Medalist from Andhra University (1992-93) is currently working as a Professor (CSE) & Associate Dean in KLEF, Vaddeswaram, Guntur. He is in teaching profession for the past 25 years and prior to joining KLEF he worked as Programme Leader in the School of Engineering, Science & Technology at KDU University, Malaysia for about 10 years. While working in Malaysia, Prof. Subrahmanyam has been associated with many reputed Universities like Carnegie Melon University (USA), Murdoch University, Monash University (Australia), Northumbria University (UK) and gained international exposure on various teaching & learning pedagogies. He is the Founder Chairman of ACM Amaravathi Chapter and an active member of other professional societies like CSI, IEEE & CSTA, etc.. He has published more than 80 research papers in both National and international journals/conferences and attended various workshops/conferences in Malaysia, Singapore, USA & India. His research interests include Knowledge Engineering, Software Engineering & Soft Systems Methodologies. He has guided 3 scholars towards their PhD, 100 over students towards their Master’s and Bachelor Dissertations and currently guiding another 8 towards their PhD.

Mrs. Diyya Vadlamudi is currently working as an Assistant Professor(CSE) in KLEF, Vaddeswaram, Guntur. She is in teaching profession for the past 12 years. She has published 8 Scopus indexed papers and her research interest is towards Cloud computing and Software Risk assessment.