International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

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

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 not 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 asses 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 indicationaround software metrics to illustrate for development in this field by critical investigation about key metrics initiated onboth 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)

Revised Manuscript Received on December 30, 2019.
* Correspondence Author

D.Naga Malleswari*, CSE, Koneru Lakshmaiah Education Foundation ,Guntur, Andhra Pradesh, India, nagamalleswary@kluniversity.in

D.Rakesh, CSE, Koneru Lakshmaiah Education Foundation ,Guntur, Andhra Pradesh, India , India, rakesh543555@gmail.com

Dr.K.Subrahmanyam, CSE, Koneru Lakshmaiah Education Foundation ,Guntur, Andhra Pradesh, India , India, smkodukula@kluniversity.in

Divya Vadlamudi, CSE, Koneru Lakshmaiah Education Foundation ,Guntur, Andhra Pradesh, India, divya.movva@kluniversity.in

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 measurablefoundation for creatingchoices around the software quality. Metrics are the numerical value of software and it is used to predict the fault [3]. Software metrics occur filelevel, 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 VikasVerma , having software metrics have a number of benefits such as provide a foundation for approximation and simplifiespreparation 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 aspectslike 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 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 this 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 softwaredevelopment 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 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 further bolstering a chance to be checked done whole task development transform to fulfillthe 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.

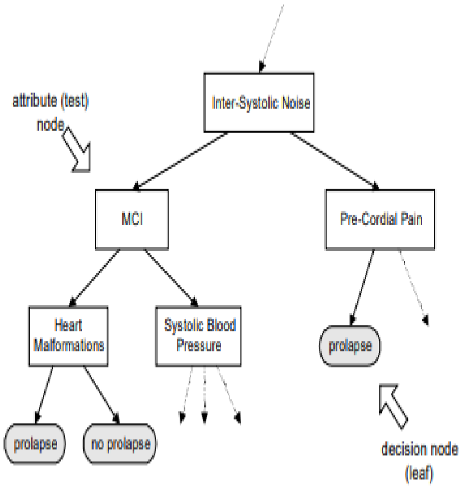
Past exploration fill in • the suggested examine includes a moved forward rendition about. AZ-Model then afterward acquiring that assessment for masters and acquiring smoothness after best possible usage for Different sizes for undertakings done associations with diverse sizes. • And, measurable analyses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

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

$$\text{Split Information}(S,A) = -\Sigma_{i=1}^{n} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \longrightarrow (3)$$

and

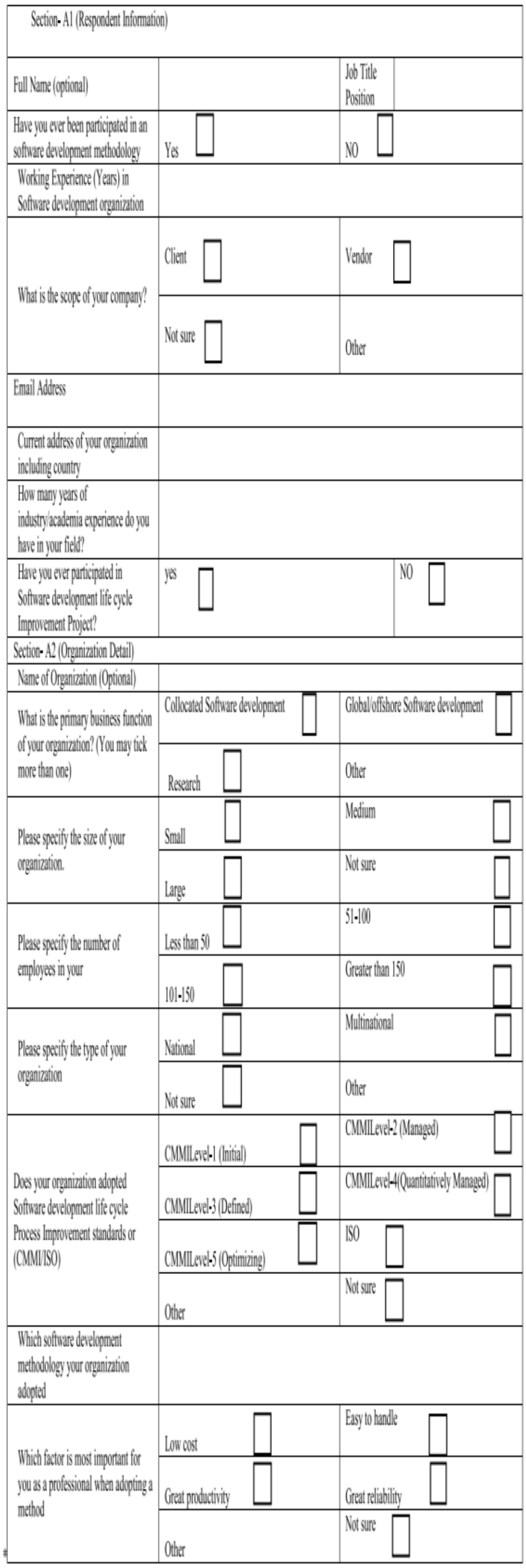
$$\text{Gain Ratio}(S,A) = \frac{\text{Gain}(S,A)}{\text{Split Information}(S,A)} \longrightarrow (4)$$

4.2 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 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.

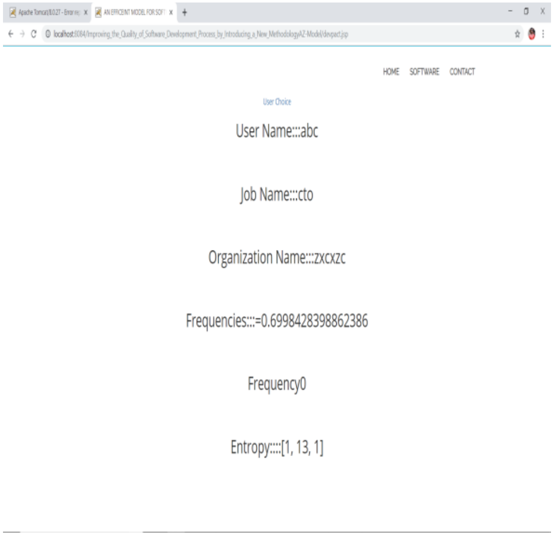
The SQMUD group if Audit those usage arrange alongside those transformed management and guarantee that trying of the product or project Throughout development stage may be finished Also acceptable of the clients And stakeholders.



Section- A1 (Respondent Information)			
Full Name (optional)		Job Title Position	
Have you ever been participated in an software development methodology	Yes □	NO □	
Working Experience (Years) in Software development organization			
What is the scope of your company?	Client □	Vendor □	
	Not sure □	Other	
Email Address			
Current address of your organization including country			
How many years of industry/academia experience do you have in your field?			
Have you ever participated in Software development life cycle Improvement Project?	yes □	NO □	
Section- A2 (Organization Detail)			
Name of Organization (Optional)			
What is the primary business function of your organization? (You may tick more than one)	Collocated Software development □	Global/offshore Software development □	
	Research □	Other	
Please specify the size of your organization.	Small □	Medium □	
	Large □	Not sure □	
Please specify the number of employees in your	Less than 50 □	51-100 □	
	101-150 □	Greater than 150 □	
Please specify the type of your organization	National □	Multinational □	
	Not sure □	Other	
Does your organization adopted Software development life cycle Process Improvement standards or (CMMI/ISO)	CMMILevel-1 (Initial) □	CMMILevel-2 (Managed) □	
	CMMILevel-3 (Defined) □	CMMILevel-4(Quantitatively Managed) □	
	CMMILevel-5 (Optimizing) □	ISO □	
	Other	Not sure □	
Which software development methodology your organization adopted			
Which factor is most important for you as a professional when adopting a method	Low cost □	Easy to handle □	
	Great productivity □	Great reliability □	
	Other	Not sure □	

Section: B –Questions related to six pointed star model. The aim of this section is to collect factor wise responses from the participants which make it possible to analyze the proposed model according to six pointed star model provided by the project management Body of knowledge (PMBOK 4.0). The factor wise collection of data make us enable to check that the proposed model is efficient for what factor to what extent. Please rank each question according to your own understanding and experience about AZ-Model of software development.						
SDA= strong disagree, DA= Disagree, N= Neutral, A=Agree, S A=Strong agree						
Factors	Questions	SDA	DA	N	A	SA
Scheduling	By using this model project team are aware of the project status	□	□	□	□	□
	Project teams get the satisfactory requirements form the customer.	□	□	□	□	□
	Project is delivered on time according to schedule.	□	□	□	□	□
Scope	Project usually has well defined scope	□	□	□	□	□
	Project management methodology is effective to make the scope clear	□	□	□	□	□
Budget	Project completed within budget	□	□	□	□	□
	The project provides good Return on Investment.	□	□	□	□	□
Risk	Project risk and opportunities are managed	□	□	□	□	□
	Business objectives are meet	□	□	□	□	□
Resources	Human and material resources are mostly available	□	□	□	□	□
	Team can work well together to achieve expected results	□	□	□	□	□
	Maximum utilization of available resources	□	□	□	□	□
Quality	Quality requirements are met	□	□	□	□	□
	Client satisfaction is met	□	□	□	□	□
	The project is successful overall	□	□	□	□	□
Section: C-Add your comment if any						

V. RESULTS


Apache Tomcat/8.0.27 - Error re
AN EFFICEINT MODEL FOR SOFT
localhost:8084/Improving_the_Quality_of_Software_Development_Process_by_Introducing_a_New_MethodologyAZ-Model/devpact.jsp
HOME
SOFTWARE
CONTACT
User Choice

User Name:::abc

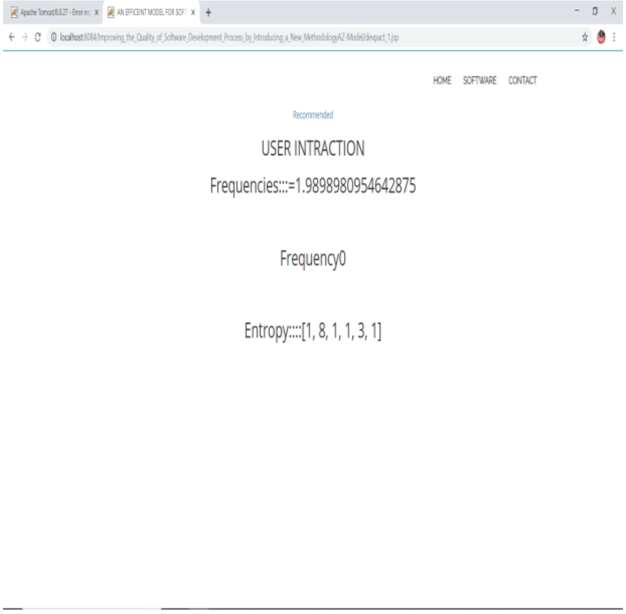
Job Name:::cto

Organization Name:::zxcxzc

Frequencies:::=0.6998428398862386

Frequency0

Entropy:::[1, 13, 1]

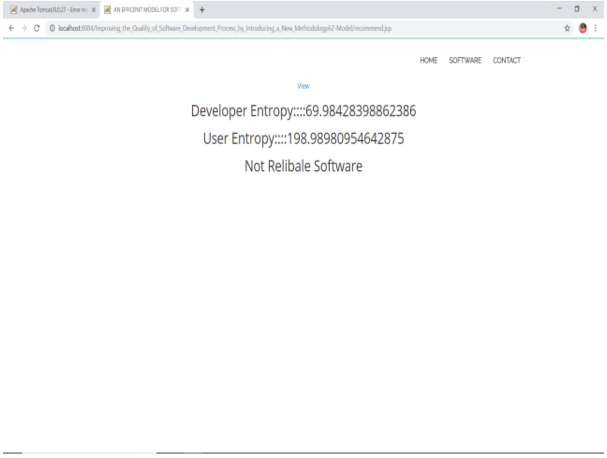

Apache Tomcat/8.0.27 - Error re
AN EFFICEINT MODEL FOR SOFT
localhost:8084/Improving_the_Quality_of_Software_Development_Process_by_Introducing_a_New_MethodologyAZ-Model/devpact_1.jsp
HOME
SOFTWARE
CONTACT
Recommended

USER INTRACTION

Frequencies:::=1.9898980954642875

Frequency0

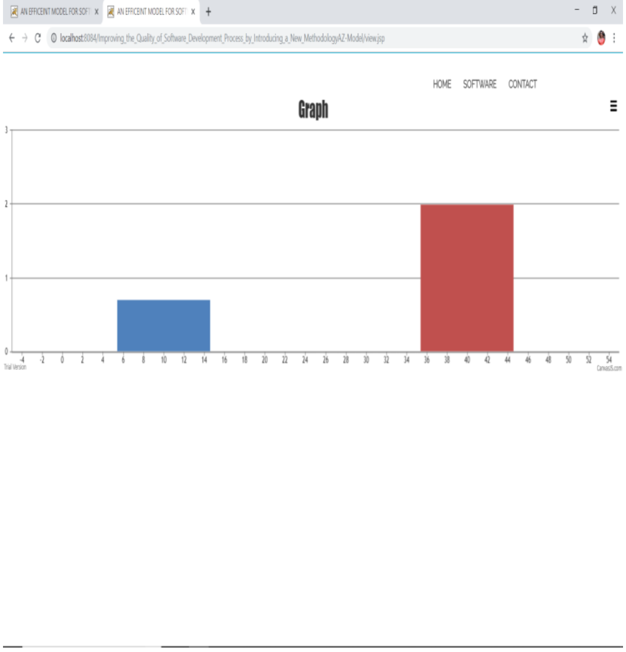
Entropy::::[1, 8, 1, 1, 3, 1]


Apache Tomcat/8.0.27 - Error re
AN EFFICEINT MODEL FOR SOFT
localhost:8084/Improving_the_Quality_of_Software_Development_Process_by_Introducing_a_New_MethodologyAZ-Model/recommend.jsp
HOME
SOFTWARE
CONTACT
View

Developer Entropy::::69.98428398862386

User Entropy::::198.98980954642875

Not Relibale Software


AN EFFICEINT MODEL FOR SOFT
AN EFFICEINT MODEL FOR SOFT
localhost:8084/Improving_the_Quality_of_Software_Development_Process_by_Introducing_a_New_MethodologyAZ-Model/view.jsp
HOME
SOFTWARE
CONTACT
Graph
3
2
1
0
-4
-2
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
Trial Version
CanvasJS.com

VI. 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 softwarequality 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.

REFERENCES

1. Ahad, A., Yalavarthi, S. B., & Ali Hussain, M. (2016). A new approach for integrating social data into groups of interest doi:10.1007/978-81-322-2755-7_24 Retrieved from www.scopus.com
2. Chandra Prakash, V., Sastry, J. K. R., Kantharao, V., Sriharshini, V., Sriram, G., & Ganesh, C. H. V. S. (2017). An expert system to assess memory power of a student for selection of a suitable career. Journal of Advanced Research in Dynamical and Control Systems, 9(Special Issue 6), 309-321. Retrieved from www.scopus.com
3. Chaitanya Krishna, B., Sai Mounish, P., Vinay Kumar, U., &Divya Mallika, A. (2018). Online business site quality assessment. International Journal of Engineering and Technology(UAE), 7, 838-840. Retrieved from www.scopus.com
4. Sharma, N., &Yalla, P. (2016). Software engineering and natural language processing- how can they be together? International Journal of Software Engineering and its Applications, 10(12), 389-396. doi:10.14257/ijseia.2016.10.12.32
5. Preetham, C. S., Siva Ganga Prasad, M., Pratap Reddy, N., &SashankaTeja, L. (2016). Outage probability analysis of amplifyandforward and decodeandforward dual hop relaying with hardware defects. Journal of Theoretical and Applied Information Technology, 86(2), 290-298. Retrieved from www.scopus.com
6. Naga Malleswari, D., Suresh Babu, S., Moparthi, N. R., Mandhala, V. N., & Bhattacharyya, D. (2017). Hash based indexing in running high dimensional software systems. Journal of Advanced Research in Dynamical and Control Systems, 9(Special Issue 6), 34-43. Retrieved from www.scopus.com
7. Chandraprakash, V., Sastry, J. K. R., Sravani, G., Manasa, U. S. L., Khyathi, A., & Harini, A. (2017). Testing software through genetic algorithms – a survey. Journal of Advanced Research in Dynamical and Control Systems, 9(Special Issue 2), 1607-1633. Retrieved from www.scopus.com
8. Shahabuddin, S. M., &Yalla, P. (2017). Impact of lean software development into agile process model with integration testing prior to unit testing. Journal of Theoretical and Applied Information Technology, 95(22), 6163-6175. Retrieved from www.scopus.com
9. Naga Malleswari, D., Kumar, M. P., sathvika, D., & Kumar, B. A. (2018). A study on SDLC for water fall and agile. International Journal of Engineering and Technology(UAE), 7(2), 10-13. doi:10.14419/ijet.v7i2.32.13516
10. NagaMalleswari, D., & Subrahmanyam, K. (2018). SIS framework for risk assessment through quantitative analysis. International Journal of Engineering and Technology(UAE), 7(2.32 Special Issue 32), 367-370. Retrieved from www.scopus.com
11. Lakhmi Prasanna, P., Rajeswara Rao, D., Meghana, Y., Maithri, K., &Dhinesh, T. (2018). Analysis of supervised classification techniques. International Journal of Engineering and Technology(UAE), 7(1.1), 283-285. Retrieved from www.scopus.com
12. Lakshmi, K. N., & Rao, K. S. S. (2018). A study on role of emotional intelligence on employee performance. International Journal of Civil Engineering and Technology, 9(3), 440-448. Retrieved from www.scopus.com
13. J. Cahill, J. M. Hogan, and R. Thomas, "Predicting Fault-Prone Software Modules with Rank Sum Classification," pp. 211-219, 2013.
14. R. C. and andF.Kemerer, "A Metrics Suite for Object Oriented Design " IEEE Transactions on Software Enginneering vol. 20, 1994.
15. Rüdiger L, Jonas L, and W. L, "Comparing Software Metrics Tools," ACM, July 2008.
Monday, August 01, 2016). Analyst4j Find Using Metrics Available: www.codeswat.com
Syntegrico. Analyst4j Find Using Metric. [24] (Friday October 30, 2015). CCCC Software metrics Available: http://www.sourceforge.net/projects/cccc/
16. Chidamber and Kemerer Java Metrics Available: http://www.spinellis.gr/sw/ckjm/ [26] (Friday, October 30, 2015). Eclipse Metrics Plug-in 1.3.6. Available: http://sourceforge.net/projects/metrics/
Eclipse Metrics Plug-in 3.4 Available: : http://eclipse-metrics.sourceforge.net/
17. Jarallah S, Raimi A, and Rufai and Sohel M, "OOMeter: A Software Quality Assurance Tool " Proceedings of the Ninth European Conference on Software Maintenance and Reengineering 2005
SciTools Source Code Analysis and Metrics, Understand for Java. Available: www.scitools.com
18. D. a. Tom, Controlling Software Projects. New York: Yourdon Press, 1986.
19. Campbell , Luke, and B. K, "Software Metrics: Adding Engineering Rigor to a Currently Ephemeral Process " 1995.
20. 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
21. Malkit S and D. S, "Software Defect Prediction Tool based on Neural Network " International Journal of Computer Applications, vol. 70 2013.
22. H. R. Bhatti, "Automatic Measurement of Source Code Complexity," MASTER'S THESIS Computer Science and Engineering Luleå University of Technology.
23. 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.
24. 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.
25. C. Z. a. Q. SHICHAOZHANG, "Data Preparation for Data Mining " Applied Artificial Intelligence vol. 17, pp. 375–381 2003.
26. S. Shivaji, "Efficient Bug Prediction and Fix Suggestions " PhD A dissertation UNIVERSITY OF CALIFORNIA 2013.
27. 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.
28. D. Kafura, "A survey of software metrics " presented at the ACM annual conference on The range of computing New York 1985
29. 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
30. 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.
31. Kunal Chopra and M. Sachdeva, "EVALUATION OF SOFTWARE METRICS FOR SOFTWARE PROJECTS " INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY vol. 14, April 2015
32. 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.
33. C. J, "Strengths and Weaknesses of Software Metrics " Software Productivity Research LLC, 2006

34. A. H. and and T. J, "Structured Testing: A Testing Methodology using the Cyclomatic Complexity Metric " National Institute of Standards and Technology Gaithersburg, September 1996 [18] H. B. KLASKY, "A Study of Software Metrics " The State University of New Jersey New Brunswick2003

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, Vaddeswarm, 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. Divya Vadlamudi is currently working as an Assistant Professor(CSE) in KLEF, Vaddeswarm, 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.