

An Authoritative Method Using Fuzzy Logic to Evaluate Maintainability Index and Utilizability of Software

*Yenduri Gokul, **Madhwaraj Kango Gopal

*Research scholar, Department of Information Technology, VFSTR Deemed to be University,
Vadlamudi, Guntur-AP, India (yenduri.gokul@gmail.com)

**Professor, Department of Information Technology, VFSTR Deemed to be University,
Vadlamudi, Guntur-AP, India (drkgm_it@vignanuniversity.org)

Abstract

Maintainability and Usability are developing software quality properties, which assume a key part in deciding the nature of a product framework. Directly, little research exertion has been committed to estimation of Maintainability and ease of use of protest arranged software framework either subjectively or quantitatively by utilizing fuzzy parts of both the factor. All the more essentially, neither any fuzzy model exists for measuring above quality components nor does a particular model presents noteworthy rules in such manner. This proposed work makes a commitment to the field of software quality and its maintenance by showing object-arranged measurements for measuring practicality and ease of use and the time factor reduction and the maintenance cost. The protest arranged measurements have an immediate association with quality elements and they can be utilized as indicators of the practicality and convenience of question situated software frameworks productively. The proposed work formally characterizes question arranged measurements remembering that Maintainability and ease of use for the most part affect on software quality. It additionally proposes model and interesting estimation techniques to assess Maintainability and ease of use of protest arranged framework. The measure of exertion expected to keep up a product framework is identified with the specialized nature of the source code of that framework. The ISO 9126 model for programming item quality perceives viability as one of the 6 principle attributes of programming item quality, with versatility, variability, dependability, and testability as sub characteristics of viability. Astoundingly, ISO 9126 does not give a consensual arrangement of measures for assessing practicality based on a framework's

source code. Then again, the Maintainability Index has been proposed to compute a solitary number that communicates the practicality of a framework. To expand nature of a product, to oversee programming more effective and to diminish cost of the product, practicality, viability estimation and practicality assessment models have been proposed. In any case, the down to earth utilization of these models in programming building devices and practice stayed minimal because of their impediments or dangers to legitimacy. In the proposed technique We have utilized fuzzy rationale for measuring the product Maintainability. Protest situated measurements have been utilized as info factors in a fuzzy surmising motor. Fuzzy approach in blend with file layered strategy is utilized as a part of the estimation of ease of use of question situated software framework. Utilizing instrument robotization, the estimations of question situated measurements were acquired. A question arranged quality model additionally has been proposed.

Key words

Software, Maintainability, Utilizability, Fuzzy logic.

1. Introduction

Interest for effective software is expanding step by step and question situated outline procedure can satisfy this request since protest arranged software improvement has moved toward becoming defect standard of software advancement associations. Question situated outline method can help in decreasing the cost as well as aides in the improvement of superb software frameworks. Software engineers constantly required fitting measurements to create quality situated software framework. Along these lines, in this perspective protest arranged measurements can assume imperative part because of their significance in the improvement of productive and effective software frameworks.

In software building environment, building up a product item with the end goal of meeting its practical necessities is vital. Decreasing the cost and exertion that are expected to finish a product item likewise possesses a similar centrality. Along these lines, software Maintainability is a testing undertaking that each association experiences. The upkeep period of any product framework is imperative since it has been appeared to be the biggest and costliest.

One of the measurements used to create the MI is the normal Cyclomatic Complexity. We feel this is a generally imperfect number. Especially for frameworks fabricated utilizing object oriented innovation, the many-sided quality per module will take after a power law conveyance. The normal many-sided quality will perpetually be low, while recounted confirm proposes that

the support issues will happen in the couple of exceptions that have a many-sided quality of more than 100.

Ease of use is likewise a standout amongst the most imperative quality factors in the fields of software building and a vital issue in the improvement of fruitful software applications. Ease of use enhances profitability, lessened preparing and documentation for software framework. There are additionally web based business thinks about that demonstrate that a 5% change in ease of use could expand incomes by 10-35%. To create proficient software frameworks, software designers require fitting measurements. Question arranged measurements can assume vital part in the advancement of fruitful and usable software applications since protest situated outline methods are a standout amongst the most effective instruments to satisfy the request of intuitive and proficient software frameworks.

At present, little research endeavors have been devoted to considering how the question arranged measurements may affect on Maintainability index of software framework and in addition how the ease of use of protest situated framework is to be measure by taking thought of fuzzy parts of both the variables. Subsequently, this proposition formally characterizes protest arranged measurements remembering that practicality and ease of use for the most part affect on software quality. It additionally proposes model and one of some kind estimation techniques to assess Maintainability and in addition convenience of question arranged framework.

In light of the constraints of measurements, for example, the MI, we have framed a comprehension of the insignificant prerequisites that must be satisfied by a commonsense model of viability that is grounded in source code investigation. With these necessities at the top of the priority list we have begun to define and apply an elective practicality show. In this elective model, a set of well-picked source-code measures are mapped onto the sub-qualities of viability as indicated by ISO 9126, following commonsense mapping and positioning rules.

The work revealed in this paper is preparatory in the sense that our practicality demonstrate is as yet advancing. Specifically, modifications and refinements are made to the model on a case by case premise. In any case, the pragmatic estimation of the model has just been shown in our training, and we anticipate advance enhancements of the model to just bring an expanded level of detail and exactness.

2. Literature Review

To enhance the nature of software framework, the protest situated outline is exceptionally mainstream approach. The quality components like multifaceted nature, ease of use, reusability, testability, practicality, understandability and so forth have much effect on the nature of software

framework. These variables are particularly related with question arranged highlights and can be likewise used to build the effectiveness of protest situated frameworks. Protest situated outline indicates the way toward arranging an arrangement of communicating objects with the end goal of taking care of a product issue.

There are different perspectives about question arranged plan ideas. AL-Badareen. et.al [2] proposed that protest situated plan must show four highlights: legacy, information deliberation, dynamic authoritative, and data stowing away. Lionel C et al. [4] portrayed that question situated outline measurements has likewise been appoint the abnormal state plan quality characteristics for the protest arranged software with the assistance of various leveled display. S. R. Chidamber et al.[5] have given discernment that nature of software additionally assumes an imperative part as far as wellbeing angles and money related perspectives. Question arranged outline is more useful in software improvement condition and protest situated plan measurements are basic highlights to gauge the product quality. Question arranged outline comprise every one of the properties and nature of software that is required to grow little or expansive venture. In this way, question situated plan strategies are valuable for improvement of extensive and complex software frameworks.

Don M et.al [6] built up the protest arranged approach called Responsibility-Driven outline. Their approach recommends that distinctive obligations are characterized for each class and to satisfy the duties of the classes, they have to build up relationship with different classes. Protest arranged investigation and plan technique was proposed by P. F. tiako [9], which is a well ordered strategy for creating object-situated models. The characterized steps are: discovering class and questions, characterizing subjects, recognizing structures, characterizing properties and characterizing administrations.

James W et al. [8] depicted an Object Modeling Technique (OMT) approach that enables framework creators to conceptualize the general framework design. OMT depicts a technique for framework examination, framework configuration, protest outline and usage of the framework and suggests utilization of these stages iteratively. The OMT approach, utilizing three models: protest, dynamic and useful, is one of the most grounded ways to deal with investigation and outline of a product framework. Examination and plan technique, created by Don Coleman et al. [10], underscores on the pretended by objects in the framework. This part is subject to the necessities of the framework instead of the properties of the protest. Along these lines a solitary question may perform distinctive parts at various phases of the framework. A pyramid display was proposed by Jacobson's Object Oriented Software Engineering (OOSE) strategy for the way

toward creating object-arranged plan, in which instruments offer help for the exercises in three classifications: design, technique and process.

Ping et al. [12] created Hierarchical Object Oriented Design (HOOD) strategy. The principle procedure in HOOD is the essential outline step. It depends on the distinguishing proof of articles by methods for protest arranged outline systems. The primary point of HOOD is to build up the outline as an arrangement of articles, which together gives usefulness to the program. The Booch strategy depicts the investigation and configuration periods of a protest situated framework usage. A way is offered by this technique from prerequisites to usage by utilizing object-arranged investigation and outline. It underscores on the refinement between consistent view and physical perspective of a framework. The philosophy works through a two-phase improvement, one full scale and the other small scale.

Sepasmoghaddam et.al [13] was the principal individual to give the significance of ease of use building and the relativity of its idea. He gave the most vital idea of item acknowledgment in view of ease of use. According to him ease of use can be characterized as: "the ease of use of a framework is the ability in human useful terms to be utilized effortlessly and adequately by the predetermined scope of, given indicated preparing and client bolster, to satisfy the predefined scope of errands, inside the predefined scope of situations". Sepasmoghaddam did not give any assessment strategies or strategy for ease of use estimation however considers that ease of use has two subscales: clients and framework's goal measures.

Abdellatief et.al [15] was another master who contemplated and perceived ease of use as an imperative trademark to impact the acknowledgment of an item. He comprehensively isolated worthiness into functional and social acknowledgment. The global association of institutionalization has been in charge of giving different measures for ease of use in the course of recent years. The fundamental need of these models is to give consistency to the segments of UI, for example, symbols, tabs, appearance and some more. Therefore, its model comprises of three fundamental traits to be specific: adequacy, productivity and fulfillment. The ISO 9241-11 set out the gauges and rules for the estimation of ease of use yet it doesn't give the methodology and activities to be attempted.

2. Maintainability Index

The Maintainability Index comprising of various effectively figured measurements, with a specific end goal to have the capacity to rapidly and effortlessly anticipate software Maintainability. It comprises of polynomial articulation and outcomes in a single number demonstrating the general framework Maintainability. The MI is assigned in view of relationship

with subjective assessments by software maintainers. The MI subsists in 2 variations, where only one component is varied.

$$MI = 182 - 6.4 * \ln(aveHV) - 1.15 * aveHV(h') - 14.5 * \ln(aveLOC)$$

in which

aveHV = average Halstead Volume V per module

aveHV(h') = average extended cyclomatic complexity per module

aveLOC = average count of line of code per module

The segments are figured at the component intensity, and afterward arrived at the midpoint of. The term 'module' utilized means littlest unit of usefulness. Contingent upon the software dialect, this is a capacity, technique, strategy, subroutine or segment.

The MI is much of the time been utilized as an examination demonstrate in other Maintainability inquire about. Other than utilizing MI at framework stage, as depicted over, a further application which is identified in some cases. In this regard, one can recognize the components with the most minimal MI esteem, which are believed to be the sections with high necessity is forwarded.

3.1 MI parts

3.1.1 Outline of Code

Software or units estimate is regularly calculated utilizing the Lines of Code (LOC) metric. It most likely was the primary software, utilized as the reason for measuring software profitability and exertion. LOC still is an extremely prevalent metric, and not at all due to its effortlessness. A few descriptions check all lines, while others tally lines in the code which ends with delimiter. The distinction among these different descriptions is as huge as 100%, contingent upon the software dialect. The MI computation calls for checking physical lines of code.

3.1.2 Metrics Used to Evaluate the Maintainability

To assess the practicality picked factors are connected with CK measurements. The CK metric suite comprises of six outline many-sided quality measurements - WMC, DIT, NOC, CBO, RFC and LCOM. The factor many-sided quality is connected with metric WMC, class is connected with RFC, coupling is connected with CBO, legacy is connected with DIT and number of kids is connected with NOC. Aside from LCOM, every one of these measurements can be

utilized as Maintainability indicators as LCOM is uncorrelated with the practicality of the product (Turk, 2009). Along these lines, LCOM is disposed of in the exploration.

3.2 Fuzzy Logic

Significant favorable position of fuzzy rationale approach is that the fuzzy rationale models can be developed with no information or with little information (Ryder 1998; Mac Donell et al. 1999). Likewise, fuzzy rationale models can adjust to new condition when information wind up plainly accessible (Sailu et al., 2004). Fuzzy rationale depends on if then guidelines, which are planned by considering the feeling of specialists from the space. Master based estimation was observed to be superior to anything all relapse based models (Musilek et al., 2000). As indicated by Lather et al. (2000), fuzzy model is the best decision for overseeing questionable, far fetched, repudiating and dissimilar feelings and it is the distance a superior decision when many-sided quality and non-linearity are high. Lin et al. (2007) introduced another appraisal technique to get the coordinated software quality for assessing client fulfillment by utilizing the fuzzy set hypothesis in view of the ISO/IEC 9126-1 (2001) quality model with a solitary evaluator. Gupta et al. (2011) has given a contextual analysis of various software quality estimation methods to fabricate a product quality model. Their paper recommends that the fuzzy and manage based framework procedures are better to design and assessing a product quality model.

3.3 Working of Fuzzy Model

The working of fuzzy model includes following advances:

i. Fuzzification: To figure how much the info information coordinate the state of the fuzzy principles.

ii. Deduction: Fuzzy derivation framework is the way toward figuring the mapping from an offered contribution to a yield utilizing fuzzy rationale. It ascertains the administer's decision in view of its coordinating degree.

iii. Piece: To consolidate the conclusion derived by every fuzzy run into a last conclusion.

iv. Defuzzification: To change over a fuzzy determination into a fresh one. The contribution for the defuzzification procedure is a fuzzy set and the yield is a solitary number.

There are add up to 11 enrollment capacities accessible in MatLab. For fuzzification we considered Triangular Membership Functions (TMF), due to its effortlessness and overwhelming use by scientists for forecast models. It is a three-point work, characterized by least α , most extreme β and modular esteem m i.e. $TMF(\alpha, m, \beta)$, where $(\alpha \leq m \leq \beta)$.

4. Maintainability Evaluation

Following sub-sections describe the fuzzy maintainability model and also describe the working methodology and validation of this model.

4.1 Proposed Fuzzy Maintainability Model

There are different techniques for Maintainability estimation (Dubey et al., 2011b), yet none of them was correct approach. Hence, we proposed a fuzzy model approach for Maintainability estimation of question situated framework.

The advantage of utilizing fuzzy rationale is that the fuzzy rationale models can be constructed even with practically no information. The proposed fuzzy Maintainability demonstrate is spoken to in Figure 4.1.

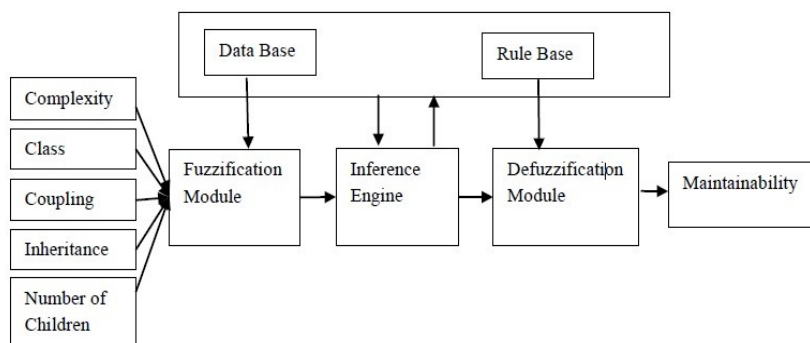


Fig 4.1. Proposed Fuzzy Maintainability Model

4.2 Working of the Proposed Model

In this model we have taken five contributions as intricacy, class, coupling, legacy and number of kids to give a fresh estimation of Maintainability utilizing standard base. Fuzzy Inference System (FIS) utilizes fuzzy rationale to outline contribution to yield. Mamdani fuzzy surmising technique is utilized. After the fuzzification procedure is finished, we take the fuzzy sets for yield variable that requires defuzzification. For defuzzification the information will be a fuzzy set and yield will be a singleton esteem. We utilized centroid strategy for this reason, which gives focus of zone under bend. There are many sorts of participation works yet for straightforwardness we have utilized triangular enrollment work.

4.3 Membership Functions for Inputs and Output

For measuring practicality of a question situated framework we have considered five sources of info - multifaceted nature, class, coupling, legacy and number of youngsters. The

fuzzifications of info factors are appeared in Figure 3.2, 3.3, 3.4, 3.5, and 3.6 individually. We have taken three enrollment capacities - Low, Medium and High for each info. Since factors are mapped with CK measurements and edge for these measurements are between 0 – 100 (selvarani et al., 2009; Shatwani, 2006). In this way, these data sources are taken in interim of [0,100]. For the yield (Maintainability) we have taken five participation capacities – low, low, medium, high and high . The range for this is likewise taken in interim of [0,100].

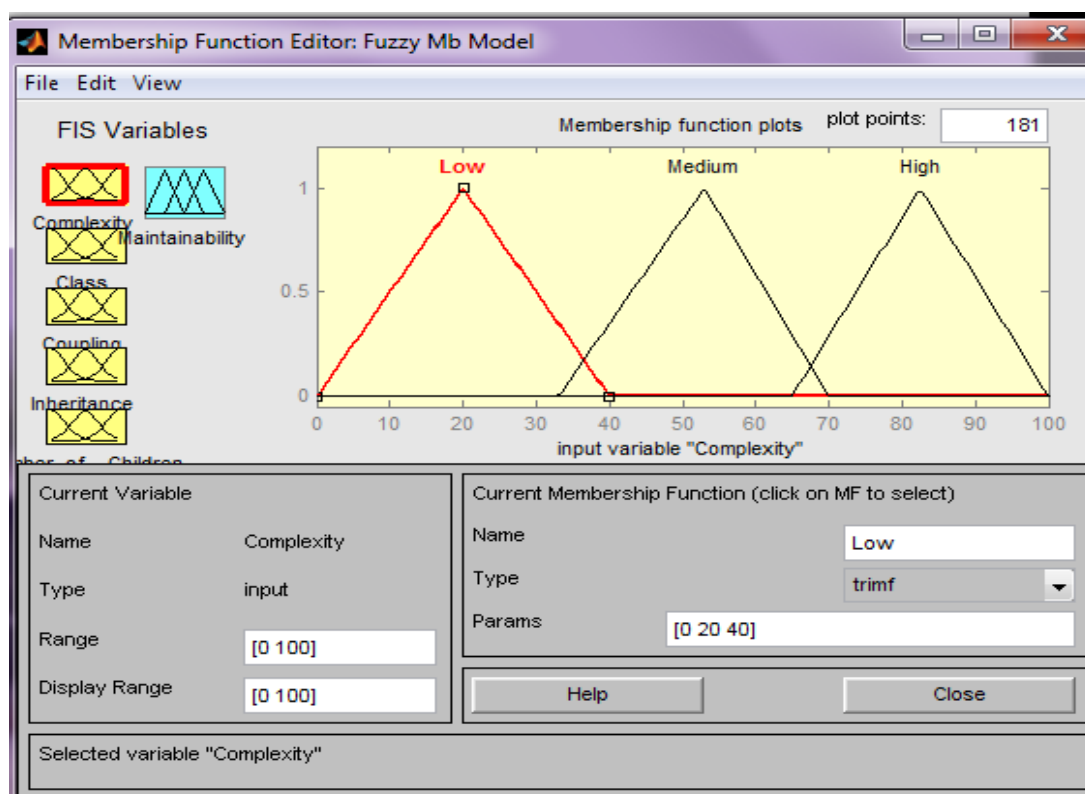


Fig 4.2. Membership Function for Complexity

4.4 Usability Model Chosen for Research

The global association of institutionalization has been in charge of giving different guidelines for ease of use in the course of recent years. The essential need of these benchmarks is to give consistency to the parts of UI, for example, symbols, tabs, appearance and some more. The convenience demonstrates decided for look into is ISO 9241-11 (1998) standard. This model is appeared in Figure 4.3. This Figure is received from the paper by Liang et al. (2009).

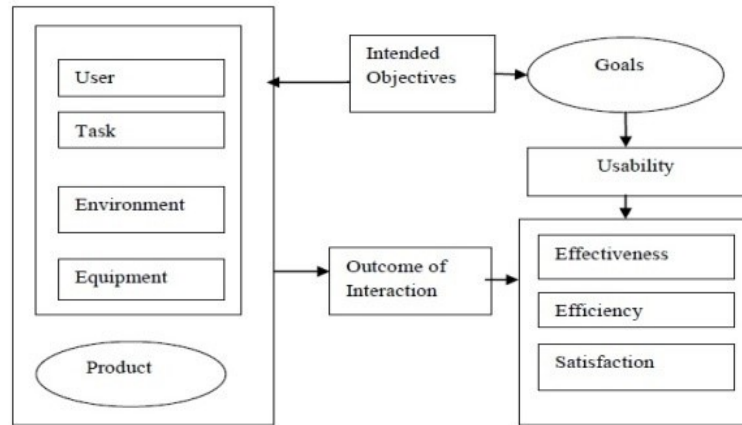


Fig 4.3. ISO 9241-11 Usability Model

In ISO 9241-11 standard, convenience is a three-dimensional nature of software item, characterized as "the degree to which an item can be utilized by indicated clients to accomplish determined objectives with adequacy, productivity and fulfillment in a predefined setting of utilization". Subsequently, this model comprises of three sub-factors as:

i. Adequacy: It can be portrayed as an execution measure of the framework to finish a predefined errand or objective effectively inside time

ii. Productivity: It can be depicted as the degree of accomplishment of an assignment by a framework. Its idea is identified with precision and fulfillment of the predetermined objectives.

iii. Fulfillment: It can be depicted as the client's adequacy of the framework, in the predetermined setting of utilization.

Bevan and Macleod (1994) likewise proposed that ease of use measure is better reflected into Maintainability, proficiency and fulfillment. Fulfillment dependably depends on subjective judgment, however on some degree, the estimation of adequacy and productivity can be equitably figured in some ways (Liang et al., 2009).

4.5 Proposed Usability Model

Ease of use comprises of different sub-factors. In this way, to finish the principle sub-factors on which convenience depends, we checked on ease of use factors by altogether investigation of measures and models (Dubey and Rana, 2010b). In survey, the four sub-factors specifically adequacy, effectiveness, fulfillment and learnability were found as the fundamental benefactors in the appraisal of convenience of the product framework.

Out of these four sub-factors, the three sub-factors viz. adequacy, productivity and fulfillment are additionally principle measurements of convenience in ISO/IEC 9241-11 standard.

The outcomes from audit demonstrates that learnability factor alongside Maintainability, productivity and fulfillment has more noteworthy effect among the various sub-factors on the UI (ease of use) of the product framework (Dubey and Rana, 2010d). As indicated by Chang and Dillon (2006), "when one is doing the advancement of the UI using a specific plan strategy, it is hard to assess convenience that the UI infers on the clients".

4.6 Measurements Used for Evaluating Usability

The idea of protest arranged software especially relates the outline and execution period of software framework, which is straightforwardly identified with the ease of use of framework. To assess the ease of use of the protest arranged software, we have to break down its outline and execution utilizing proper measurements. There is no single metric that measure every one of the highlights of a question situated software item. In view of this reality, appropriate measurements are expected to evaluate the convenience of software framework since ease of use builds dependability of software and furthermore portrays the consequence of an improvement exertion.

5. Results

Figure 5.1 indicates different anomalies or potentially extremes. The estimations for these frameworks were checked however no mistakes could be found. In this manner, these information focuses are viewed as substantial. The frameworks being referred to were not expelled from the informational collection.

Table 1. Descriptive statistics of results

language	type	SMM								MI					
		maint	ana	chg	tst	dip	comp	usise	vol	MI3	MI4	aveV	aveV(g')	aveLOC	perCM
C	min	-2.000	-2.000	-2.000	-2.000	-2.000	-2.000	-2.000	-2.000	49	81	565.4	3.479	21.89	0.08
	max	0.611	0.667	1.000	0.500	2.000	2.000	-1.000	2.000	89	124	3359.1	12.491	163.88	0.44
	median	-1.278	-0.667	-1.500	-2.000	-1.000	-2.000	-2.000	2.000	76	103	862.9	5.301	40.95	0.23
C++	min	-1.889	-1.667	-2.000	-2.000	-2.000	-2.000	-2.000	-1.000	58	97	361.9	2.368	16.99	0.09
	max	0.000	0.000	0.500	-0.500	1.000	1.000	-2.000	2.000	97	119	1621.4	6.304	76.89	0.36
	median	-1.056	-1.000	-1.000	-2.000	-1.000	-2.000	-2.000	0.000	85	115	636.9	3.319	26.68	0.18
Java	min	-1.889	-1.667	-2.000	-2.000	-2.000	-2.000	-2.000	-1.000	81	109	133.4	1.288	7.37	0.03
	max	0.889	1.000	1.500	1.000	2.000	2.000	0.000	2.000	113	150	713.6	3.405	29.56	0.42
	median	-0.611	0.000	-0.500	-1.500	0.000	-1.000	-2.000	2.000	100	136	243.0	1.985	14.05	0.28
OO	min	-1.889	-1.667	-2.000	-2.000	-2.000	-2.000	-2.000	-1.000	58	97	133.4	1.288	7.37	0.03
	max	0.889	1.000	1.500	1.000	2.000	2.000	0.000	2.000	113	150	1621.4	6.304	76.89	0.42
	median	-0.750	-0.333	-0.500	-1.500	0.000	-1.000	-2.000	1.000	98	132	277.3	2.228	14.90	0.27
all	min	-2.000	-2.000	-2.000	-2.000	-2.000	-2.000	-2.000	-2.000	49	81	133.4	1.288	7.37	0.03
	max	0.889	1.000	1.500	1.000	2.000	2.000	0.000	2.000	113	150	3359.1	12.491	163.88	0.44
	median	-0.944	-0.333	-1.000	-1.500	-1.000	-1.000	-2.000	1.000	93	124	395.2	2.776	18.21	0.24

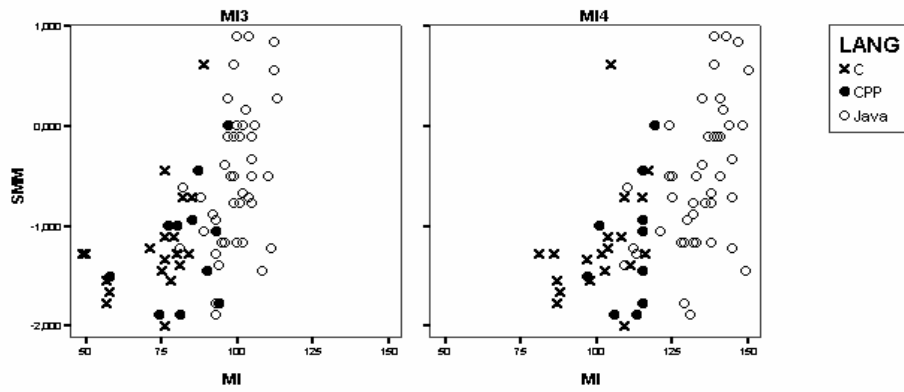


Fig 5.1. Scatter plots of MI3 and MI4 against SMM

Table 2. Rank correlation of SMM versus MI3 and MI4

<i>language</i>	<i>SMM - MI3</i>			<i>SMM - MI4</i>		
	<i>corr.</i>	<i>sign.</i>	<i>n</i>	<i>corr.</i>	<i>sign.</i>	<i>n</i>
C	0.494	0.016	19	0.476	0.020	19
C++	0.365	0.135	11	0.423	0.098	11
Java	0.459	0.001	43	0.500	0.000	43
OO	0.554	0.000	54	0.569	0.000	54
all	0.621	0.000	73	0.617	0.000	73

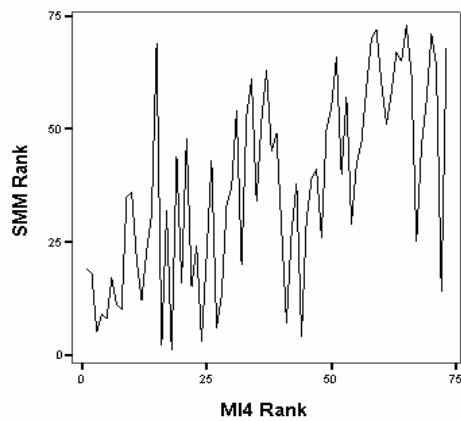
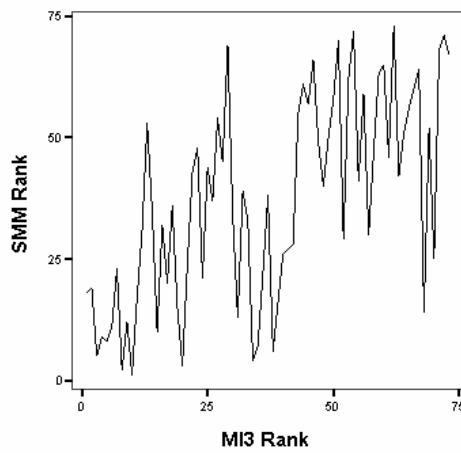


Fig 5.2. MI3 and MI4 ranking against SMM ranking

6. Conclusion

The part initially distinguishes the most imperative factors that effect on practicality of question arranged framework. These variables are then identified with Chadamber and Kemerer measurements suit. The work proposed a fuzzy rationale based way to deal with assess the Maintainability for protest arranged software framework. Add up to 243 standards are intended for the model in view of master's conclusion. These principles are spoken to by part works and separated into three classifications, Low, Medium and High. Yield variable i.e. Maintainability is isolated into five classifications: Very Low, Low, Medium, High and Very High. Practicality of proposed demonstrate is assessed utilizing focus of gravity defuzzification strategy for online question situated ventures. Fuzzy model robotizes assessment of the Maintainability. The model was approved by Analytical Hierarchy Process. The outcome demonstrates that introduced show can evaluate the Maintainability of protest arranged framework as a solitary score.

It proposed a model for ease of use and the utilizations fuzzy file strategy to subjectively assess it for protest situated software framework. Record framework strategy utilizes base to top approach for the product components. The recognized ease of use sub-factors for this model are Maintainability, effectiveness, fulfillment and learnability. These Usability sub-factors are chosen by completely investigation of gauges and models. The assessment is contrasted and ISO 9241-11 convenience show. A relationship is set up between the variables of ease of use and the plan metric framework that prompts the assessment of software framework convenience. The outcome demonstrates that subjective nature of proposed question situated ease of use show is superior to the current models. Practicality display is assessed by utilizing fuzzy rationale system. Nonetheless, other rising methodologies in software designing like hereditary calculation, neuro-fuzzy strategy and so forth may likewise be utilized to assess. Proposed ease of use demonstrate is assessed utilizing fuzzy record technique. This model might be likewise assessed by utilizing manufactured neural system approach. The proposed model might be thought about by other ease of use models moreover. In both the models we have taken just CK measurements suits as info measurements. So in future, other question arranged measurements may likewise incorporate into look into as information sources measurements. Proposed protest arranged quality mode is assessed and approved just on classroom based task. Endeavors are being made to assess and approve it for the genuine modern applications.

References

1. Pigoski, T. Practical Software Maintenance. Wiley computer publishing, 1997.

2. AL-Badareen, Anas Bassam, et al. The impact of software quality on maintenance process. 2010, ACM International Journal of Computers, vol. 2.
3. Ian Somerville, Software Engineering Chapter 21, Evolution Process 7th edition published by Dorling Kindersley(India) pvt. ltd. copyright © 2004.
4. Lionel C. Briand, Christian Bunse, and John W. Daly. A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs, 2001. IEEE Trans. Software Eng., vol. 27, no. 6, pp. 513–530.
5. S. R. Chidamber, C. F. Kemerer. A metrics suite for object oriented design, 1994, IEEE Trans. Softw. Eng., vol. 20, no. 6, pp. 476–493.
6. Don M. Coleman, Dan Ash, Bruce Lowther, Paul W. Oman. Using metrics to evaluate software system maintainability, 1994. IEEE Computer, vol. 27, no. 8, pp. 44–49.
7. Ward Cunningham. The WYcash portfolio management system. In OOPSLA '92: Addendum to the proceedings on Object-oriented software systems, languages, and applications (Addendum), 1992, pp. 29–30, New York, NY, USA. ACM.
8. Alain April, Alain Abran. Software maintenance management: evaluation and continuous improvement, 2012. Vol. 67. John Wiley & Sons.
9. James W. Paulson, Member, IEEE, Giancarlo Succi, Member, IEEE Computer Society, and Armin Eberlein, Member, IEEE Computer Society, An Empirical Study of Open-Source and Closed-Source Software Products, 2004, IEEE transactions on software engineering, vol. 30, no. 4.
10. P. F. Tiako, maintenance in joint software development, in computer software and applications conference, 2002. COMPSAC 2002. proceedings. 26th annual international, pp. 1077-1080.
11. Don Coleman, Dan Ash, Hewlett-Packard Bruce Lowther, Paul Oman, Using Metrics to Evaluate Software System Maintainability, NIX-9162194/\$4.00 © IEEE
12. Lincke Rüdiger, Jonas Lundberg, Welf Löwe. Comparing software metrics tools, 2008. Proceedings of the 2008 international symposium on Software testing and analysis. ACM.
13. Liang Ping. A Quantitative Approach to Software Maintainability Prediction. In Information Technology and Applications (IFITA), 2010 International Forum on, vol. 1, pp. 105-108. IEEE.
14. Sepasmoghaddam, Alireza, and Hassan Rashidi. A novel method to measure comprehensive complexity of software based on the metrics statistical model. Computer Modeling and Simulation (EMS), 2010 Fourth UKSim European Symposium on. IEEE.

15. T. E. Tutt, et al. Risk-informed Preventive Maintenance optimization. Reliability and Maintainability Symposium (RAMS), 2012 Proceedings-Annual. IEEE.
16. Majdi Abdellatif, Abu Bakar Md Sultan, A. A. Ghani, Marzanah A. Jabar. Multidimensional size measure for design of component-based software system. Software, IET 6, no. 4 (2012): 350-357.
17. Sneed, Harry M., Katalin Erdos. Measuring and Evaluating a DotNet Application System to Better Predict Maintenance Effort. Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2012 Joint Conference of the 22nd International Workshop on. IEEE.
18. Martin Kunz, Reiner R. Dumke, Andreas Schmietendorf. How to measure agile software development, 2008, pp. 95–101.
19. Mika V. Mäntylä, Casper Lassenius. Drivers for software refactoring decisions. In ISESE '06: Proceedings of the 2006 ACM/IEEE International symposium on Empirical software engineering, pp. 297–306, New York, NY, USA. ACM.
20. Mika V. Mäntylä, Casper Lassenius. Subjective evaluation of software evolvability using code smells: An empirical study, 2006. Empirical Softw. Engg., vol. 11, pp. 3, pp. 395–431.
21. Naouel Moha, Yann-Gael Gueheneuc, Laurence Duchien, AnneFrancoise Le Meur. Decor: A method for the specification and detection of code and design smells, 2010, IEEE Trans. Softw. Eng., vol. 36, no. 1, pp. 20–36.
22. http://blogs.msdn.com/b/zainnab/archive/2011/05/26/code-metrics-maintainabilityindex.aspx?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+zainnab+%28Visual+Studio+Tips+and+Tricks%29.