RESEARCH ARTICLE OPEN ACCESS

A Comparative Study of Software Requirement, Elicitation, Prioritization and Decision Making.

Pradeep Kumar. G [1], Sandhiya .R [2], Prof. Manjula. R[3].
[1, 2, 3]*(Department of Computer Science, VIT University, Vellore- 632014)*

ABSTRACT
The failure of many software systems are mainly due to the lack of the requirement engineering. Where software requirement play a very vital role in the field of software engineering. The main task of the requirement engineering are eliciting the requirements from the customer and to prioritize those requirements to make decisions in the software design. Prioritization of the software requirement is very much useful in giving priority within the set of requirements. Requirement prioritization is very much important when there are strict constraints on schedule and the resources, then the software engineer must take some decisions on neglecting or to give prioritization to some of the requirements that are to be added to the project which makes it successful. This paper is the frame work of comparison of various techniques and to propose a most competent method among them.

Keywords: software requirement prioritization, dynamic, prioritization, stakeholders, AHP, Hundred-dollar test, Ranking, Numerical assignment.

I. INTRODUCTION

Requirements engineering is found to handle the activities that challenge to know the precise requirements of the clients in a software management system and to convert such needs into exact and definite statements, which are then utilized in the system growth.

In this paper we are using AHP method, which was established for performing elicitations, prioritization and decision making processes. It is importantly known for optimizing the decision making when one is facing with a mixture of quantitative, qualitative and occasionally conflicting factors that are in the concern. In general Elicitation involves defining the requirements of the stakeholders and understanding, finding, mining and /or determining requirements of the clients and the stakeholders [1]. Requirement elicitation process is one among the most significant knowledge intensive activities of the software development. A study that has been performed by [2] states that, 70% of the system errors are because of the insufficient specification of the system and 30% of the system errors are because of the design issues. Analyzing the software security system based on the requirements of the system elicited that are in the form of use and misuse cases. Use cases are established to be helpful in the elicitation of communication about, and the function requirements documentation. Using the elicitation process can assist in generating a steady and whole rest of the security requirements. The last but most important step is prioritizing the software, which mainly relies on the Requirements and the elicitation factors. In this paper we make a comparative study among few prioritization methods and conclude that which method can yield feasible solution.

II. RELATED WORKS

Software requirement prioritization is made by comparing AHP and ELECTRE I. In order to reflect the reality several stake holders and software professionals are directly involved in the case study. AHP method is difficult in performing software projects with a huge volume of requirements. Thus it is concluded that ELECTRE I, is more appropriate than AHP [1]. An Experimental pattern is made to compare three Software Requirements Prioritization Techniques that are AHP, Cumulative Voting, and Numerical Assignment. The main goal of this pattern is to analyse which technique is best suitable in performing, software prioritization in terms of time, accuracy, and portability. And finally it is concluded that cumulative voting is found to produce good prioritization results. A machine learning approach is carried out for performing software requirement prioritization. A method called Case-Based Ranking (CBRank) is used, that associates the preferences of project stakeholders along with the approximations of ordering that is computed with the help of machine learning approaches and brings out with promising advantages. In the first place, the human push to input preference data is decreased, while saving the exactness of the last positioning appraisals. Second, space information encoded as incomplete request relations characterized over the prerequisite

qualities can be misused, in this manner supporting a versatile elicitation process. Exact assessments of properties of CBRank are performed on reproduced information and contrasted and a best in class prioritization strategy, giving confirmation of the technique capacity to bolster the administration of the trade-off between elicitation exertion and positioning precision and to endeavor area learning. At long last, the CBRank technique has been situated regarding best in class approaches, with specific reference to the AHP strategy, which can likewise be viewed as an occasion of the case-based critical thinking worldview .Performing software requirement elicitation and prioritization using a process called analytical hierarchy. Success and the failure of a software system mainly rely on the quality of the requirements. The techniques that are hired during the requirements elicitation influence the requirements quality. Elicitation of Requirements is the most crucial part of the software development, because the defects at this initial stage is get transmitted through the development process and it is hard to repair them later. An algorithmic method is carried out to elicit the software requirements prioritization using Analytic Hierarchy Process (AHP), by which the requirements are ranked and implemented in a very easy way. Software requirement prioritization is carried out using fuzzy multi attribute decision making, which overcomes the defects like lack of consistency, time complexity, and implementation difficulties. This approach considers the indefinite nature of the quality attributes and requirements by designing the end as a fuzzy variable. This requirement prioritization problem is designed as a fuzzy multi attribute decision problem, by which predictable operator value is used in order to rank the alternatives that are listed in the formulation of a problem. Finally it has been stated that this approach is capable of breaking several features and criteria for providing a complete view of quality requirements. A new way for handling requirement prioritization using B-tree method is carried out. An organization must know about its customer perspective towards their products, as the customer expectations are not static, periodic prioritization is a must in order to increase the customer satisfaction. When considering a larger requirements case the prioritization technique requires more comparisons, thus B-tree method is used in order to overcome such problems. From this method it is found that the requirement comparison is drastically reduced if the branching factor is high, where its control is given to the prioritization group. These can be managed by simply altering the t value. Thus B-tree method is used in which the comparison required by the respective method is drastically reduced. Classifying several software requirement prioritization methods by highlighting their important features is carried out. This requirement prioritization process should be simple and firm producing us with valid results. The framework involved in developing the design to describe the proposals of the requirements and these prioritization methods are analyzed using a conceptual framework. The differences among these methods are highlighted by stressing their vital features which are useful in the elicitation process. Optimization of Requirement Prioritization using a new technique called FuzzyHCV is carried out. This FuzzyHCV is a hybridized method of Fuzzy Expert System and Hierarchical Cumulative Voting (HCV). Comparative analysis is made between the proposed technique and the currently available HCV techniques and found that the proposed FuzzyHCV technique is more consistent than any other existing techniques. Performing and analyzing an effective test case prioritization agenda, which make use of design diagrams, software requirements specification, test cases and source codes as input and delivers a prioritized order of test cases as an output. The requirement IDs are fragmented into words in order to calculate the requirements relativity and the calculation to find the interconnection of the activities, the design drawings were extracted in the XML format. Final weight is calculated by multiplying design interconnectivity, Requirements relativity and class dependencies with their assigned weights. The test cases are selected by mapping the test cases and customers' requirements using that weight. This proposed agenda is analyzed and finally stated that the collaborative information during the prioritization process is beneficial. A hybridized approach is provided for requirement engineering in the agile software development, with the assistance of JAD and prioritization method. Prioritization is performed with the help of viewpoint, by which increment selection is made very easy. Thus it is clear that the hybrid approach strongly focus on increased delivery and quality of the software.

III. ANALYTICAL HIERARCHY PROCESS (AHP)

The analytical hierarchy process which was developed by Thomas L. Saaty in the period of 1970s. It is a structured Techniques for organizing and analyzing complex decisions, from then many more studies and refinement were carried out. It is most widely used to make group decisions and in some of the decision situations. It is also very much used in other fields such as education, business, healthcare, industry and shipbuilding. AHP helps

the decision makers to find the decision that is best suitable, rather than choosing the correct decision for their goal. It also gives the perfect structure to the decision problem by providing a comprehensive and rational framework that helps in better representation and quantification of its elements and making relationship with the elements to overall goals. It is also useful in evaluation of alternative solutions.

The first step in the AHP is to decompose the decision problem into hierarchy of divided sub-problem and each of the sub-problem can be analyzed independently. After the hierarchy is built, the decision makers evaluate the various elements by comparing them to each other two at a time, in systematic approach. AHP converts theses evaluations into numeric values and that are processed and compared to the entire range of the problem. In the final step, for the each of the decision alternatives numerical priorities have been created.

IV. REQUIREMENTS ELICITATION TECHNIQUES:

The most important task in the software development is to extract the requirement from the user, here many techniques are developed for eliciting requirements from the customer. Development approach in the some of the elicitation techniques mainly concentrate on the security issues. On the other hand, which is traditional development to extract the requirements from the customer in general, this also support the security issues. In general software requirement elicitation is performed with the use of any elicitation methods or with help of few techniques. There are many elicitation techniques that helps in the extraction of requirement from the customer, few of them will be discussed for this study.

Elicitation of the requirement is a method in which the requirements of developing a software are collected very carefully which influences the development of the software and its reliability and feasibility. For this reason, requirement elicitation is given highest priority during the initial stages of the development of software. When this part is ignored or proper work is not put into it, it affects the overall work greatly. Any mistakes that are done at this stage or any constraints that are missed out will be infectious even in the later stages of the development, which may possibly negate a lot of other good work put into it. When considering the failures of software, 70% of them is because of the improper requirement elicitation.

There are lot many approaches which can be used for the requirement elicitation. Among them two techniques are very interesting one. They are

1. **Misuse Cases**
2. **Joint Application Development (JAD)**

Misuse Cases

This technique uses the uses cases to identify the defects or hidden constraints of the software development. It is a kind of backward engineering, where an undesirable scenario or constraint of the development is identified first, rather than setting up the constraints for the end user requirement. This technique excels in several environment, where there are several unknown aspects that can affect the system, which make them very useful in the security requirement elicitation. This technique creates a tree structure with the parent nodes reflecting the requirement and the child nodes corresponds to identifying the constraints. Whenever the child identifies an anomaly or a flaw, it propagates the details to the parent node for it to make necessary changes and consider them as a constraint. This makes the technique as a bottom up approach, eliciting the requirements from the details identified in the system.

JAD

Joint Application development is a very effective method for satisfying the end user and the shareholders. Here all the major parties involved in the project, including core developers, managers, end users and other interested parties. Here all the requirement elicitation and constraints are shared with the users and their opinions on the matter are gathered. Here no technical aspect of the development are used as a talking point. This will provide an end result that satisfies the consumer greatly. But it is highly mandatory to initially perform a few tasks ahead of the meeting like gathering the technical aspects, collecting initial requirements from it and features that can satisfy it.

These two techniques are comparatively very different that can be used together for different purposes. The Misuse Cases are arguably the best in terms of identifying the security threats while the JAD focuses on the satisfaction of the end user. Understanding the software and its usage is important in choosing the elicitation method.

Now that all the requirements are collected, whether by JAD or Misuse Cases, an impending task of prioritizing them arises. This is handled by the AHP. This is a decision making mechanism that identifies the right kind of requirement using pair wise comparison matrix that has to be focused. This technique greatly makes the software more desirable and reliable. The developers can make verifications or changes in the result obtained by the AHP, which makes them more effective.

V. PRIORITIZATION TECHNIQUES

Analytical Hierarchical Process (AHP):

The AHP method was discovered by Saaty [4] in the year 1980. The study stated by Regnell et al [12] declare that AHP a very hopeful method, but its unable to adapt to the prioritization environment that is found with several stakeholders, thus it should be designed in some other way to adapt the situation that is given. Still no research has been carried out to state what kind of modification could function properly.

In case of AHP method, requirements of the candidates are analyzed in a pair wise manner, and also to what degree one requirement is highly significant than other one. As this method describes the pair a wise analysis that is carried out on all the requirements of the candidate, the count of comparison raises polynomial. Pair wise analysis is made for a system that has n requirements which is represented as, n. (n - 1)/2.

S. NO	Level of Importance	Description
1	1	Equal Significance
2	3	Moderate Variaion in significance
3	5	Significant essential variation
4	7	Significant chief variation
5	9	Huge variation in importance

Cumulative Voting (CV):

Cumulative Voting (CV) was discovered by Widrig and Leffingwell , it is a linear and interesting voting system in which each and every stakeholder is provided with a constant quantity of invented units that a person could make use of it for voting purpose in goodwill of some significant issues [14]. By this path the quantity of money that has been allotted to the issue corresponding to the relative preference of respondent with other issues related to it. The statements are scattered according to the stakeholder's preference. Possibly the stakeholders can divide the amount equally to many issues.

CV method is also called as "proportional voting" method, it is because the quantity of units that is assigned to the issues provides the information about the virtual priority of some exact problems that is related to some other problems. The word "proportional" replicates the main truth that if the quantity of units that has been allotted to a specific issue is found separated by some constant units that is presented to each and every stakeholders, the final outcome turn out to be a zero and one proportion. Thus the ratings of the stakeholders for a certain layout of issues are thus taken as the mixture of person's perspective in case of any problems, where each and every issue takes a certain amount of choices into the person's preferences.

This process could result in some problems that are given with zero units, which states that the particular stakeholder takes these problems as an insignificant one. In particular type of data the value zero is generally a problem it is because they prepare the virtual needs or significant as entirely a meaningless one making the calculation of zeros as impossible one. In common the rule of CV method allows the stakeholders to distribute their entire quantity without any limitations.

Planning Game (PG):

In the case of extreme programming requirements are given on a story card by the customer, which differ based on several factors. Thereby the requirements are separated into three piles by the respective customer. Beck states that piles should have certain names that are nice to hold on [16]. Thereby the programmer calculates the time that is taken by each requirement for implementing and arranging the requirements into three various piles.

The final resulting of this sorting process is placing the sorted requirements on the ordinary scale. As the PG process consumes single requirement and concludes the pile to which the requirement has to be placed, thus time taken for prioritizing n number of requirements is considered as n comparisons. It clearly states that PG method is very supple and can be performed on several requirements with less time consumption.

Binary Search Tree (BST):

BST is an algorithm which is mainly to store the data and to retrieve back for the future reference. In the binary search tree, the tree which will be empty or it is engaged with the two child nodes. There will be three node, they are root node, right node and left node. In which right node that contains value which is highly important than the value of root node, where the value of left node which is less important when compared to root node. Each child node which acts as the root node for its child in the next level of hierarchy, where it is known as leaf. This method helps in the recursive search in this algorithm. The merits of using binary search tree algorithm is that, when the requirements are inserted in the order it takes only n log n number of comparisons for the prioritization of the software requirements. With help of this BST can prioritize in faster manner, where it can prioritize more number of requirements in the given time. It can be scaled up for more number of requirements.

Experimental work

Experimental work has been carried out by the comparison of various factors with the prioritization techniques such as ease of use, scalability, total time taken, total number of comparisons etc.

Ease of use

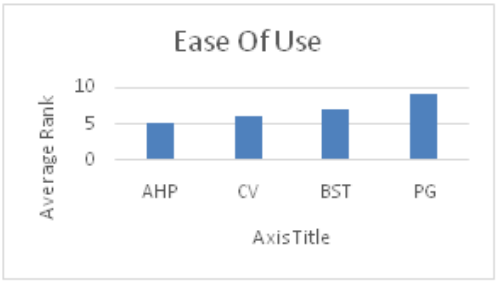

Fig: represents the parameter of ease of use, here comparisons between various prioritization have been carried out and the result is PG is more ease of use than any other techniques compared here.

Total time taken

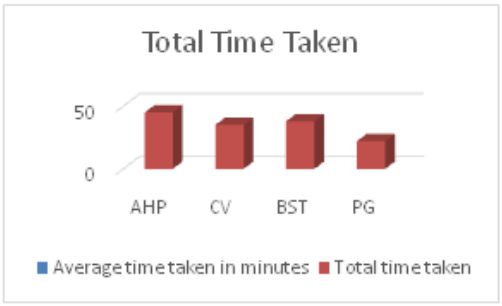

Fig: represents the parameter total time taken, here AHP is more time taken than any other methods compared here.

Accuracy

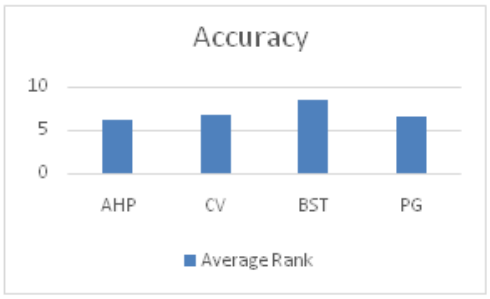

Fig: represents accuracy measures of the prioritization process. This graph represents that BST is more accurate than all other comparisons.

Scalability

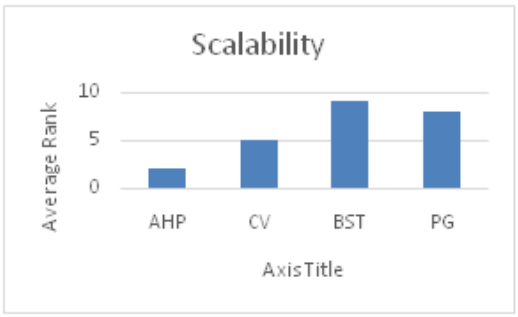

Fig: represents scalability measure of the techniques, BST is more scalable when compared with the above techniques.

Total number of comparisons

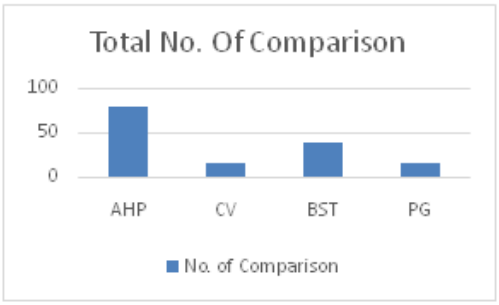

Fig: representation of total number of comparison needed to show the results.

Overall score

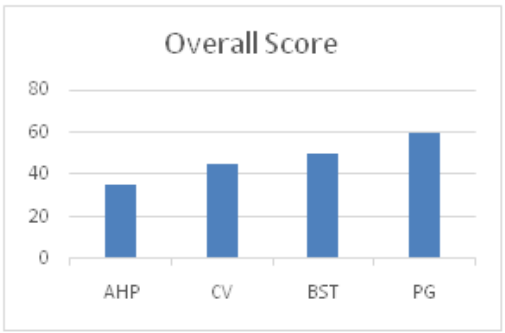

Fig: shows that PG is more suitable than any other techniques among the above comparison.

VI. DECISION MAKING

This section is all about analytic hierarchy process (AHP) which is one of the well-known methods of decision making for a multi- criteria decision problems. AHP is more suitable when there is a mixture of qualitative and quantitative aspects and when there are more clash among several requirements which are to be taken as deliberations. AHP is developed in such a way that, to enhance the role of decision making. AHP is more effective in the case of irreparable decisions. The decision-making process comprises the use of perception, intelligence and creativeness of the humans by satisfying their fundamental needs.

Estimating a decision involves various concerns like welfares are extracted for making the proper decision, risks, costs, and profit arising from the actions that are taken if the decision is made wrong. Methods of Decision making ranges from dependence on chance to the structured decision making tools. Considering all the features of the element, which makes decision making more effective. The merits and demerits of the object are to be taken into consideration that's helps in the upcoming achievements and to be in the business life or the life can be enhanced. In recent days of decision making has been more complex when many feature are to be taken into consideration. AHP was developed to overcome these issues like ranking, evaluation of decision and to prioritize the requirements.

In the year 1970s, AHP was developed by Thomas Saaty, who was working as a professor in the Wharton School of Business institute and as a consultant in an agency called Arms Control Disarmament. There he faced with some problematic factors like higher cost and a host of the considerations that are not specified easily. The AHP method deals with factors like resource, weapons tradeoffs, asset allocation, and decision making, where the judgements of the decision makers are made used in order to break down the difficulties into hierarchies. The complexity of the problem is denoted by the count of levels that are presented in the hierarchy that join with the model of the judgement maker's problem that has to be solved. Hierarchy is made used in order to extract the ratio values for project risks, the related measure which are against the goals of the organization (organizational effects, satisfaction of the customer, service/product, human resources and finance) and decision replacements. This AHP method make use of matrix algebra for arranging the factors in order to attain optimal solution and thus it is used in forming several trillion dollar decisions.

AHP decomposes the issues into hierarchies with the help of sentences of the decision makers. The number of stages in the hierarchy is the representation of the complication in the problem, the problem to be cleared here AHP combines with the decision maker's model. The levels of hierarchy which is used to eradicate the measures of ratio-scaled to which to make some alternative in the decisions and also the measure of the relative values by making the alternative decisions which affects the project risk and for the organizational goals. Mathematically the optimal solution is derived with help of matrix algebra in AHP. AHP which is said to be time-tested method that can also be used in multibillion dollar decisions.

AHP consist of 4 steps

1. Outlining of the problem and stating of goal or objective.
2. Expression of factors those are influencing the goals. And structure these factors into hierarchical levels.
3. Using of paired association of each of the factor with respect to each other that forms the comparison matrix.
4. Synthesize the alternative decisions rank until the final choice is made.

AHP scale for pair wise comparison	Numeric Rating	Decimal
Extreme Importance	9	1/9
Very strong to extremely	8	1/8
very strong importance	7	1/7
strongly to very importance	6	1/6
strong importance	5	1/5
moderate to strong	4	1/4
moderate importance	3	1/3
equally to moderate	2	1/2
equal importance	1	1

VII. CONCLUSION

Comparison has been made between the techniques that are used at the different stages of the software engineering. They are elicitation, prioritization and decision making of the software requirements. The conclusion from this paper work is that AHP is well suitable technique for the software requirement elicitation and for the decision making for the software requirements. But then for the process of requirement prioritization AHP is not suitable because of it takes more number of comparison i.e. n.(n-1)/2 number of comparisons. This time complexity need to be reduced and that makes AHP as an effective technique for the software requirement prioritization.

REFERENCE

[1]. Fernandez, J. M., Prozil Rodrigues, S., & Costa, L. A. (2015, June). Comparing AHP and ELECTRE I for prioritizing software requirements. In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on (pp. 1-8). IEEE.

[2]. Mustafa, B. A., & Zainuddin, A. (2014, August). An experimental design to compare software requirements prioritization techniques. In Computational Science and Technology (ICCST), 2014 International Conference on (pp. 1-5). IEEE.

[3]. Perini, A., Susi, A., &Avesani, P. (2013). A machine learning approach to software requirements prioritization. Software Engineering, IEEE Transactions on, 39(4), 445-461.

[4]. Sadiq, M., Ghafir, S., &Shahid, M. (2009, October). An approach for eliciting software requirements and its prioritization using analytic hierarchy process. In Advances in RecentTechnologies in Communication and Computing, 2009. ARTCom'09. International Conference on (pp. 790-795). IEEE.

[5]. Ejnioui, A., Otero, C. E., & Qureshi, A. A. (2012, October). Software requirement prioritization using fuzzy multi-attribute decision making. InOpen Systems (ICOS), 2012 IEEE Conference on (pp. 1-6). IEEE.

[6]. Beg, R., Abbas, Q., &Verma, R. P. (2008, July). An approach for requirement prioritization using b-tree. In Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on (pp. 1216-1221). IEEE.

[7]. MartínezCarod, N., &Cechich, A. (2005). Classifying software requirement prioritization approaches. In XI CongresoArgentino de Ciencias de la Computación.

[8]. Sharif, N., Zafar, K., &Zyad, W. (2014, April). Optimization of requirement prioritization using Computational Intelligence technique. In Robotics and Emerging Allied Technologies in Engineering (iCREATE), 2014 International Conference on (pp. 228-234). IEEE.

[9]. Siddik, M. S., &Sakib, K. (2014, December). RDCC: An effective test case prioritization framework using software requirements, design and source code collaboration. In Computer and Information Technology (ICCIT), 2014 17th International Conference on (pp. 75-80). IEEE.

[10]. Kumar, M., Shukla, M., & Agarwal, S. (2013, December). A Hybrid Approach of Requirement Engineering in Agile Software Development. In Machine Intelligence and Research Advancement (ICMIRA), 2013 International Conference on (pp. 515-519). IEEE.

[11]. Ann M. Hickey, Alan M. Davis, "Requirements Elicitation and Elicitation technique selection: A Model for Two knowledge-Intensive Software development Process", Proceedings of the 36th IEEE International Conference on System Sciences, 2002.

[12]. Beichter F et al, " SLAN-4-A Software Specification and Design Language", IEEE transaction on Software Engineering, SE- 10,2, 1994, pp 155- 162