Pak. j. eng. technol. sci.
Volume 6, No 1, 2016, pg 42- 63
ISSN: 2222-9930 print
ISSN: 2224-2333 online

Requirements Prioritization Techniques for Global Software Engineering

Samina Saghir
University of Agriculture Faisalabad.

Tasleem Mustafa
University of Agriculture Faisalabad.

ABSTRACT

Increase in globalization of the industry of software requires an exploration of requirements engineering (RE) in software development institutes at multiple locations. Requirements engineering task is very complicated when it is performed at single site, but it becomes too much complex when stakeholder groups define well-designed requirements under language, time zone and cultural limits. Requirements prioritization (RP) is considered as an imperative part of software requirements engineering in which requirements are ranked to develop best-quality software. In this research, a comparative study of the requirements prioritization techniques was done to overcome the challenges initiated by the corporal distribution of stakeholders within the organization at multiple locations. The objective of this study was to make a comparison between five techniques for prioritizing software requirements and to discuss the results for global software engineering. The selected techniques were Analytic Hierarchy Process (AHP), Cumulative Voting (CV), Value Oriented Prioritization (VOP), Binary Search Tree (BST), and Numerical Assignment Technique (NAT). At the end of the research a framework for Global Software Engineering (GSE) was proposed to prioritize the requirements for stakeholders at distributed locations.

KEYWORDS: Requirements Engineering, Requirements Prioritization, Global Software Engineering, Stakeholders, Prioritization Techniques.

1. INTRODUCTION

Prioritization of the requirements is an action through which all the requirements must be identified that is extremely important for the organization. Only a few requirements can be put into practice in first release; however it is necessary for a product to be according to the demands of the clients and to be provided to the customers at the specified time (Siddiqi and Shekaran 1996). In software engineering, prioritization of the requirements is most imperative activity of the process of requirement engineering which takes part in formulating proper judgments for the systems of software. The concept of globalization for business is becoming popular from almost twenty years. The management, development, and maintenance of software product have dispersed from single development location to several locations throughout the world, most of the times recognized as Global Software Engineering (GSE). Software development at various locations is a development that is geographically dispersed and permits all stakeholders including developers, managers and others to stay at different distant and distributed places throughout the software development process (Herbsleb and Moitra, 2001).

This research explores that how the process of prioritization and choice of requirements is controlled throughout the development of software in globally distributed software organizations working in the software business and which are the functional issues faced during global software engineering. Furthermore, the appropriateness of prioritization techniques from the requirements engineering writings to handle these issues in distributed system development is discussed. Scope of this research is to evaluate some of the techniques of prioritization and to introduce a new framework for requirements selection and prioritization by considering the importance value for sites and stakeholders for global software engineering.

Global Software Engineering (GSE) permits colleagues to be situated in different remote/various separation destinations throughout the software lifecycle. Global Software Engineering (GSE) includes the development of use software through connections of individuals, associations, and innovation crosswise over countries with distinctive foundations, and working styles. In these methodologies the stakeholders as scattered crosswise over diverse nations, and time zones to accomplish the basic goals. GSE has a few favorable circumstances over the customary or non-conveyed software development process. These points of interest are managing gifted works, low-work, day and night development, cost reserve funds in creating nations and nearness to the customer. GSE is

one of the greatest difficulties in circulated development environment from the software development perspective (Minhas and Majeed, 2012).

Research Problem

The research problem of this thesis is declared as follows:

"What is the condition of practice in the prioritization and determination of requirements in dispersed organizations working in the software product development, and how can the diverse difficulties be handled with existing requirements prioritization strategies for global software engineering?"

The goal of this research is to make a comparison between five techniques for prioritizing software requirements and to apply the result for global software engineering. The chosen techniques are Analytic Hierarchy Process (AHP), Cumulative Voting (CV), Value Oriented Prioritization (VOP), Binary Search Tree (BST) and Numerical Assignment Technique (NAT). Another goal is to propose a new framework to prioritize requirements for global software engineering.

2. RELATED WORK

Ahmad *et al.* (2010) stated that the choice of the right requirements for product launch can make a big difference in the success of the product. The choice of the requirements is done by various techniques used for prioritization of requirements. These techniques are based on systematic and pre-defined steps to compute the comparative weight. Prioritization is complicated by the development of new environments, from development cooperation with the traditional seat of geographically dispersed development. Stakeholders associated with a project, are distributed worldwide. This geographical distribution of the stakeholders makes it difficult to demand that every stakeholder has to prioritize his own expectations and perceptions of the software project requirements.

Ramzan (2010) described that the process of prioritizing requirements helps the stakeholders to formulate a choice about the selection of the system requirements and organize them in the best requirements groups. Depending on the most advantageous groups the project manager may decide for the releases of the artifact. In its standard run; organization's main requirements will be applied in the opening release and depending on their precedence the subsequent set of requirements will be applied in the consecutive releases.

Smite *et al.* (2010) stated that many organizations started global software engineering (GSE) to get benefit from economical, quicker and improved development of software systems. Although, empirical studies show that accomplishment of these benefits is not a simple task and the GSE field is still undeveloped. The quantity of empirical studies is comparatively small. The most of the studies generate problem-oriented reports by focusing on diverse aspects of GSE organization in terms of valuable practices or techniques. Organizations are still motivated by cost reduction tactics, and at that time, the frequently discussed suggestions specify a need of investments in travelling and socialization.

Minhas and Majeed (2012) stated that prioritization of requirements is a process to find out the importance and rank of requirements in which requirements should be developed and tested during the lifecycle of development. The necessity and importance of requirement prioritization has enlarged due to global software development style to attain low-cost software. The importance is more provoked when software development is dispersed between different development locations. As scattered stakeholders have their own impact on development through their attitude, language, culture, time zone, technical ability.

Parviainen (2012) mentioned that the gradually more complex and spirited market condition places extreme demands on organizations, involving them to give response to demands of clients, and to provide appropriate practical ability and best class software. Organizations want to utilize the current resources efficiently and accurately, and there is also a requirement to use resources on a global coverage from various locations inside the organizations and from associated companies all over the world.

Richardson *et al.* (2012) mentioned that in today's global economy, it is expected that an increasing number of software engineers work in a scattered locations. In the dispersed sites, physical distance establishes the corporal division between the management and team members and limited the chances to have a contact directly. There is also need for cooperation and coordination, and cultural difference has a negative impact on the understanding level and assessing the behavior and hard work of the team members due to a long distance. The difference of a native language generates new obstacles and hurdles in the process of communicate. These differences can result in the total space, which is usually faced in the Global Software Engineering (GSE) settings. In the result of these differences GSE has to face complicacies as compared to those practiced in locally developed setups for software.

Seyff *et al.* (2015) described that the role of stakeholders, such as the potential end users of the system mechanism is a key achievement feature generally for software engineering

(SE) and particularly for the Requirements Engineering (RE). A number of process models are available to explain requirement engineering activities. The major activities are the eliciting requirements, prioritizing and negotiating. Eliciting requirements is the process of finding, capturing and consolidating requirements through available sources. The gathered requirements must be given priorities. The priority of a requirement demonstrates its significance compared with other; it also helps us to choose the requirements that must be applied in a development. Furthermore, prioritization of the requirements helps fulfilling demands that target on resolving inconsistencies by finding a contract that suits to all groups of stakeholders that are involved usually.

3. RESEARCH METHODOLOGY

The Research Approach

Quantitative research is characterized by the author in his study as: to clarify some phenomena by gathering numerical information that are examined utilizing scientific based routines (Creswell, 1994). As per different author, quantitative research is a research that spotlights on observational research (Cohen and Manion, 1980). Observational research intends to try to focus the extent to which a particular program or approach exactly satisfies or does not satisfy a specific standard or standard. A few cases of quantitative strategies which are very much acknowledged by human researchers in their research include: Surveys, experiments and numerical methods whereas qualitative research is characterized as to study things in their normal settings and endeavoring to decipher phenomena from individuals' perspective (Pearson *et al.*, 2005). Another author clarified it as: the qualitative strategies are dominatingly convenient for uncovering the rich typical world that underlies needs, wishes, implications and decision (Kozinets, 2002). Samples of qualitative techniques include: Literature Survey, case study, ethnography and documentary analysis.

At the point when requirements are evoked, it frequently yields a greater number of requirements than can be actualized immediately. The requirements should be organized so that the most critical ones are met by the soonest product releases (Siddiqi and Shikaran, 1996). During a task, decision makers in software development need to settle on a wide range of choices with respect to the release arrangement. Issues, for example, accessible assets, points of reference, clashing stakeholder views, accessible business sector opportunity, risks, product procedures, and costs should be looked into when arranging future releases. Unluckily, there is an absence of straightforward and powerful

methods for requirement's prioritization, which could be utilized for release planning (Karlsson and Ryan, 1997).

Prioritization Stages

Prioritizing methods provide the guidance to decision makers in their task of analyzing the requirements for assigning the symbols or numbers expressing their significance. A prioritizing procedure may include three consecutive stages (Karlsson *et al.*,1998)

1. The Preparation Stage

It is the first stage where an individual structures the requirements as per the standard of the organizing strategies to be utilized. A group and a group pioneer for the session is chosen and gave all vital data.

2. The Execution Stage

It is the second stage where the leaders do the real prioritization of the requirements utilizing the information they were furnished with in the previous stage. The assessment criteria must be settled upon by the group before the execution stage is started.

3. The Presentation Stage

It is the third stage where the results of the execution are displayed for those included. Some prioritization techniques involve various types of computations that must be done before the outcomes can be presented.

Prioritization Techniques

This section enlightens the techniques for requirements prioritization considered in this thesis.

1. Analytical Hierarchical Process (AHP)

The Analytic Hierarchy Process (AHP) was initially developed and illustrated by Saaty (Saaty, 1980). Despite the fact that this is a promising procedure, the system itself is not adjusted to dispersed prioritization with various stakeholders; consequently it must be changed in somehow. However, at present time there have not been distributed any research how that sort of adjustment would work (Regnell *et al., 2001)*. In AHP the applicant requirements are compared in form of pairs, and to which degree one of the requirements are more imperative than the other requirement. The value of importance ought to be as indicated by Table 1. Since this method recommends pair-wise correlations

of all applicant requirements, the required number of comparisons develops polynomial. For a software framework with n candidate requirements, n. (n - 1)/2 no. of pair wise comparisons are required (Saaty, 1980).

2. Cumulative Voting (CV)

The Cumulative Voting (CV) or 100-Point Method or Hundred-Dollar ($100) test, portrayed by Leffingwell and Widrig, is a basic, clear and instinctively engaging voting plan where every stakeholder is given a consistent sum (e.g. 100, 1000 or 10000) of nonexistent units (for instance fiscal) that he or she can use for voting for the most vital issues (Leffingwell and Widrig, 2003). Along these lines, the measure of cash relegated to an issue speaks to the respondent's relative inclination (and in this way prioritization) in connection to alternate issues. The focuses can be circulated in any capacity that the stakeholder wishes. Every stakeholder is allowed to put the entire sum given to him or her on one and only issue of commanding importance.

3. Value Oriented Prioritization (VOP)

VOP utilizes a system that gives requirement engineers an establishment for prioritizing and settling on choice about requirements. It gives visibility to all stakeholders during choice making, wiping out extensive dialogs and contentions over individual requirements by accentuating the center business values. The initial phase in setting up a quality situated prioritization procedure is to build up a system for recognizing the business's core values and the relative connections among those values. VOP utilizes the connections that exist between core business values to evaluate and organize requirements and guarantee their traceability. The VOP system sets up an instrument for measuring and requesting requirements for an application increment, a model, or a software requirements determination. Organization officials recognize the core business values and utilize a straightforward ordinal scale to weight them as indicated by their importance to the organization (Azar *et al.*, 2007).

4. Binary Search Tree (BST)

BST is a computer algorithm with the reason to store information, which then could be recovered or looked for after. The BST as a rule is either empty, or has one or two child nodes. The child nodes to the right side R2 have more prominent quality/importance than the root node R1 and the child node on left side R3 have less esteem/importance than the root node R1. Each child node is in itself a root node to its child nodes. In the event that a node does not have any child node, it is known as a leaf. This makes it possible to inquiry in the BST recursively. The advantage for utilizing BST, when organizing requirements, is that with n requirements, it takes just n log n comparisons until every one of the

requirements have been embedded all together. That makes BST a quick applicant, which could be great if there is a considerable measure of requirement to organize among, i.e. BST could undoubtedly scale up to a great many requirements, and still be a quick hopeful. There is one imperative thing to think about the BST calculation, which is that a tree should be adjusted to have the most brief insertion time (Standish, 1997).

5. Numerical Assignment Technique (NAT)

The numerical assignment technique is in light of the rule that every requirement is appointed an image speaking to the requirement's apparent importance. This methodology is regular in Quality Function Deployment (QFD) where organizing of competitor requirements is needed (Sullivan, 1986). A few variations taking into account the numerical assignment procedure exist. A clear way to deal with the procedure is displayed by, who propose that requirements ought to be named compulsory, alluring, or inessential. A methodology utilizing better granularity is to appoint every requirement a number on a scale running from 1 to 5, where the numbers show (Brackett, 1990):

1. Does not matter.
2. Not important (the customer would accept its absence).
3. Rather important (the customer would appreciate it).
4. Very important (the customer doesn't want to be without it).
5. Mandatory (the customer cannot do without it).

Experiment Framework

The purpose of the experiment is to compare the five prioritizing techniques to identify which one of them is declared as better among all, i.e. which technique is the simplest to utilize, takes minimum time, scalable when including more requirements, most accurate and takes less number of comparisons. This is tested by permitting the participant's answer how they encounter and trust that each technique would have the capacity to satisfy every basis. With the inspiration of picking up a superior comprehension of requirements prioritization procedures, the author performed a single project study with the point of describing and assessing the five prioritizing methods from the viewpoint of clients. The test was populated with ten stakeholders. They were requested to prioritize thirteen quality requirements utilizing the prioritization methods under consideration. The requirements were prioritized by the participants freely, and to the best of their insight. The quality requirements were prioritized without considering the cost of accomplishing the requirements. That is, just the importance for the clients was considered. Besides, the requirements were considered orthogonally, i.e. the importance of one requirement is not related to another.

Keeping in mind the purpose to minimize the risk that the members recall how they did the last prioritization, the author spread the test over small duration of time with settled interims. Only one technique was used in a day. Consistently, 30 minutes were allocated for exhibiting the strategy which was under perception on that day and after receiving the confirmation from each participant who clearly understood the prioritization technique, 90 minutes were allocated for completing the experiment of that day. Every participant was supplied with essential papers and time taken by every participant to finish the trial was recorded independently.

Analysis of Collected Data

The testing starts with the first question of each method; trailed by the second and third etc. For every question, participants ranked every technique and finally mean value was calculated. Those questions that the participants were asked after every method were the following:

Question#1

The first question that was asked from the participants was about accuracy and they were asked to position the techniques according to their judgment in the perspective of accuracy of the outcome created by each method. The response of the question is depicted in Figure 3.1.

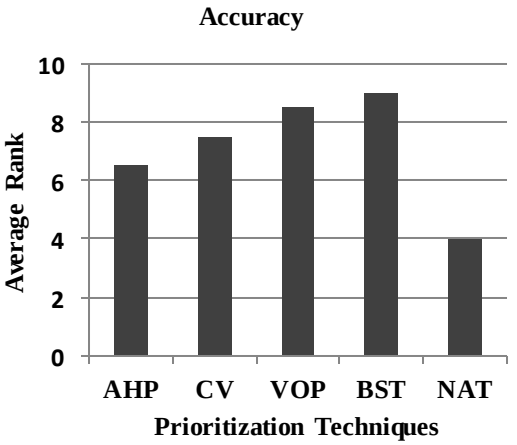


Figure 3.1 Comparison of the techniques for the criteria "Accuracy"

Question#2

The second question that was asked from the participants was about the ease of use and they were asked to identify that how easy the prioritization technique was to apply. The response of the question is depicted in Figure 3.2.

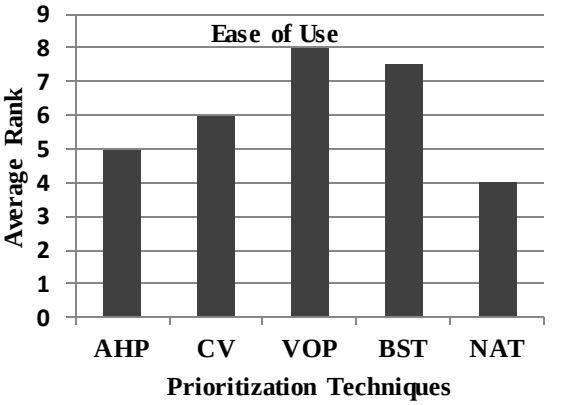


Figure3.2 Comparison of the techniques for the criteria "Ease of use"

Question#3

The third question that was asked from the participants was about scalability of the techniques and they were asked to position the techniques according to their judgment in the perspective of the use of more than thirteen requirements for each technique. The response of the question is depicted in Figure 3.3.

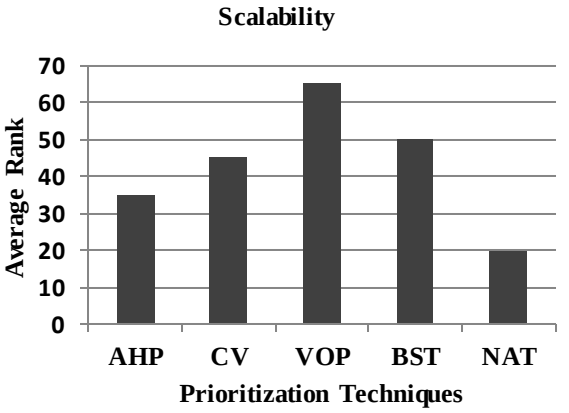


Figure 3.3 Comparison of the techniques for the criteria "Scalability"

Question#4

The fourth question that was asked from the participants was about how much time they took to execute the prioritization with the help of techniques under observation. The response of the question is depicted in Figure 3.4.

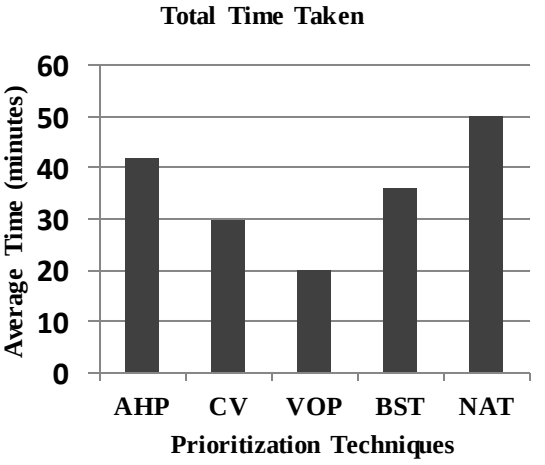


Figure 3.4 Comparison of the techniques for the criteria "Total Time Taken to Prioritize"

Question#5

Final question that was asked from the participants was to identify that how many comparisons were needed for each and every technique. The response is depicted in Figure 3.5.

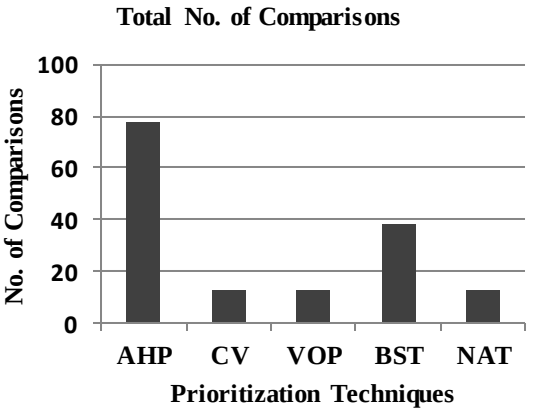


Figure 3.5 Comparison of the techniques for the criteria "Total Number of Comparisons"

Selecting the Best Technique for Prioritization

After gathering data based on above mentioned criteria, the researcher allocated weight for each criterion and then applied the formulae given in eq. 1 and eq. 2 to determine the overall best technique for the process of requirements prioritization. Each of the above criteria was allocated weight according to Table 3.1.

Criteria	Weight
Accuracy	8.5
Ease of Use	9
Scalability	8
Total Time Taken	7
Total No. of Comparisons	8

Table 3.1 Weight Table for Each Criterion

Then following formulae were used to compute the overall score by each of the prioritization techniques under consideration.

$$S_{ij} = W(C_i)*((N+1) - R_i(T_j)) \quad (1)$$

$$OS(T_j) = \frac{\sum_{i=1}^{NC} Ci,j}{NC} \quad (2)$$

Where,
N = Number of techniques used
S_{ij} = Score of technique j in criteria i
W (C_i) = Weight of criteria i
NC = Number of criteria's
Ri (T_j) = Ranking of technique j in criteria i
OS(T_j) = Overall score of technique j

The result after calculation is depicted in Figure 3.6.

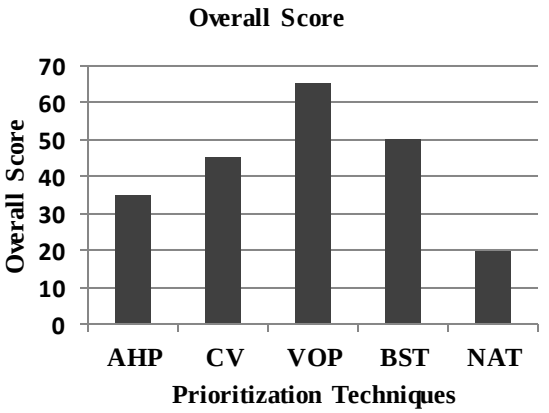


Figure 3.6 Comparison of the techniques on the basis of weighted value of criteria's

The order of the requirement prioritization procedures got from this experiment, however, is not a worldwide one as rankings can be reordered if model weights are doled out in an unexpected way. Rather, the procedure and formulae utilized here to analyze among distinctive prioritization methods can be utilized as a part of any situation with proper standard weights suitable for that situation.

STEP-II

Global Software Engineering (GSE) permits stakeholders to be present at different remote locations throughout the software lifecycle. GSE includes the development of application software through interaction of individuals, organizations, and technology

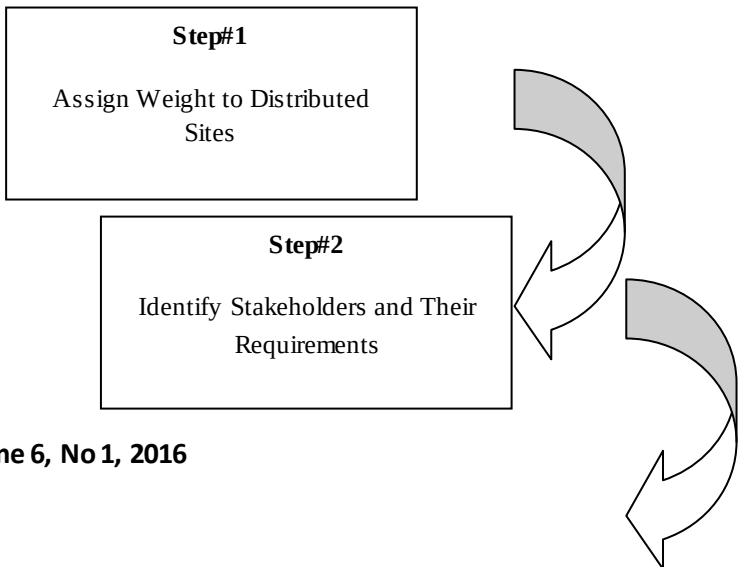
crosswise over countries with diverse foundations, languages, and working styles. In these methodologies the stakeholders as scattered crosswise over diverse nations, continents and time zones to accomplish the normal goals. GSE has a few preferences over the conventional or non-dispersed software development process. These points of interest are managing skilled labours, low-work, around the clock development, cost savings in developing nations and closeness to the customer. GSE is one of the greatest issues in disseminated development environment from the software development perspective. The temporal and geographic distance between stakeholders created the problem in the RE process development. GSE main factors that have impact on software development are distance, communication, time zone, culture etc.

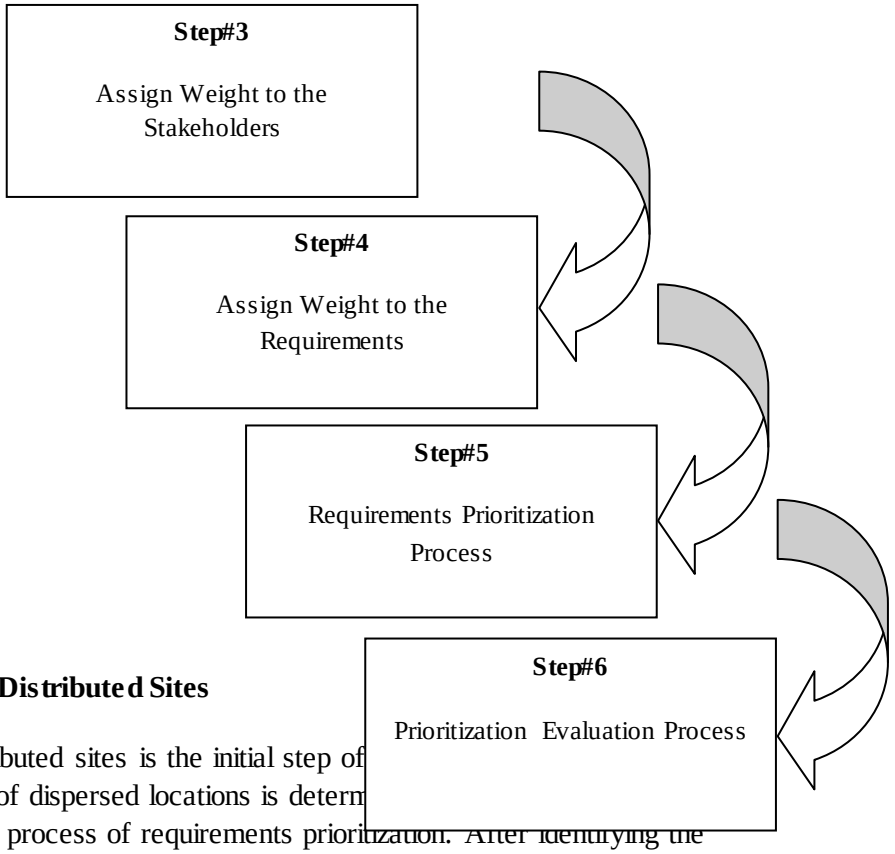
A lot of techniques have been developed for prioritization of the requirements according to the demands of the clients. Most of them are created to fulfill the demands of the customers located at same location. It may be noted that we are moving towards the global software engineering and have a broader scope for the development of software products. We have a number of distributed sites, stakeholders and numerous requirements in case of GSE.

The Proposed Framework

In this section the author proposed a new framework for requirement prioritization with reference to Global/Distributed Software Engineering. The proposed system starts after requirement elicitation and takes some starting inputs from elicitation process. The author supposed in this research that requirement elicitation procedure obtain the requirements after taking some valuable inputs like client's needs, existing system information, domain information, standards and regulations and so forth. The proposed framework comprises of six important steps that are as follows

Figure 3.7 The Proposed Framework





1. Assign Weight to Distributed Sites

Identification of distributed sites is the initial step of [illegible] this step the number of dispersed locations is determ[illegible] of that location in the process of requirements prioritization. After identifying the distributed sites the next step is to assign the weight to the dispersed locations according to their importance

2. Identify Stakeholders and Their Requirements

In this step all the stakeholders are identified at distributed places among various countries having different culture, time zone, language, technical ability and knowledge ability. In this step profiles are collected of all identified stakeholders. To identify requirements of the stakeholder is a very important step of the proposed framework, in this step the requirements of all the stakeholders are found and these requirements are used contribute in the process of requirement prioritization.

3. Assign Weight to the Stakeholders

In this step the knowledge base system assigns weight to identified stakeholders. As each stakeholder have some interest and influence on software during the software development. Each Stakeholder may have different worth, post, experience, knowledge, technical ability, nationality, culture, government policies and trustworthiness which are the basic GSE impact factors. This research study shows that requirement prioritization activity is affect by the distribution of the stakeholders with reference to GSE. Some stakeholders may have more importance as compared to other stakeholders. Therefore, these factors have high influence on software success and can play vital rule during the selection of right requirements for releases in requirement prioritization activity.

4. Assign Weight to the Requirements

In this step of the proposed framework the requirements are assigned weight according to their importance with respect to the distributed sites and stakeholder's weightage and GSE impact factors.

5. Requirements Prioritization Process

In this step of our proposed framework a process of prioritization is applied on the basis of weighted requirements and used Value Oriented Prioritization (VOP) technique selected from the previous experiment for requirements with respect to cover GSE factors.

6. Prioritization Evaluation Process

In this step of the proposed framework a process of evaluation of the prioritization is applied, if a problem occurred in the process of requirement prioritization the process has to be repeated.

4. RESULTS AND DISCUSSION

The aim of the research was to make the comparison between the five prioritization techniques to assess which one of them is by all accounts the better, i.e. which system is the most easy to utilize, takes less time, scalable for more requirements, accuracy and takes less number of comparisons. This is tried by letting the members' answer how they encounter and trust that every method

would have the capacity to satisfy every rule. This experiment was conducted by the exploratory research.

The research was led in two steps. First step of this research spoke the truth to assess five techniques to prioritize the software requirements. The chosen strategies were Analytic Hierarchy Process (AHP), Cumulative Voting (CV), Value Oriented Prioritization (VOP), Binary Search Tree (BST) and Numerical Assignment Technique (NAT). To assess these methods, the author connected all procedures efficiently for the process of prioritization of an arrangement of thirteen requirements. After that the methods were ordered from a client's perspective considering the five elements, for example, accuracy, easy to use, scalability, total number of comparisons needed to set aside a time taken by the procedure.

Result of the experiment of the first step showed that VOP was the best system for organizing software requirements. It is a simple system, it gives a standout amongst the most precise results, and it is somewhat agreeable to handle regardless of the possibility that there are numerous more requirements. In many questions BST was situated in the center, neither the best nor the most exceedingly bad systems. The most exceedingly awful applicant as indicated by result is NAT. The purposes behind most exceedingly awful execution of NAT are deciding the supreme data is troublesome than relative data, members' subjective assessments with respect to a number vary generally, it is not viable when quantities of requirements are low, less precise and educational, it takes greatest time to organize. However, this request of the requirement prioritization procedures got from this examination, however, is not a global one as rankings can be reordered if model weights are doled out in an unexpected way. By the by, the method and formulae utilized here to analyze among diverse prioritization procedures can be utilized as a part of any situation with proper model weights suitable for that situation.

In this research, it is found that the existing techniques cannot cover requirement prioritization process for global software engineering (GSE). A couple of frameworks are created for GSE, however they can have various short accompanies appreciation to GSE on prioritization. Thus, there is a requirement for more research on requirement prioritization process in term of GSE. In this study the essential target is to break down the five existing techniques used for requirements prioritization and after that to research most prominent components that impact the requirement prioritization process when stakeholders are scattered

among the different nations. After that another framework has been proposed by using the best system from the examination.

In this step the author proposed another structure for requirement prioritization with reference to Global/Distributed Software Engineering. The proposed framework began after requirement elicitation and took some beginning inputs from elicitation process. The author assumed in this research that requirement elicitation technique get the requirements in the wake of taking some profitable inputs like customer's necessities, existing framework data, space data, norms and regulations etc. The proposed structure includes six essential steps that are;

Weight of disseminated destinations is the beginning step of the proposed system, in this step the quantity of scattered areas is resolved to judge the importance of that area during the time spent requirements prioritization. After that every one of the stakeholders are recognized at dispersed spots among different nations having distinctive society, time zone, language, specialized capacity and information capacity. In this step profiles are gathered of every single distinguished stakeholder. To recognize requirements of the stakeholder is a critical step of the proposed system, in this step the requirements of the considerable number of stakeholders are discovered and these requirements are utilized contribute as a part of the procedure of requirement prioritization. After that the learning base framework appoints weight to distinguished stakeholders. This research study demonstrates that requirement prioritization movement is influence by the dispersion of the stakeholders with reference to GSE. A few stakeholders may have more importance when contrasted with different stakeholders. Consequently, these variables have high impact on software achievement and can play imperative standard during the choice of right requirements for releases in requirement prioritization action. In this progression of the proposed structure the requirements are doled out weight as per their importance regarding the disseminated locales and stakeholder's weightage and GSE effect variables. In the proposed structure a procedure of prioritization is connected on the premise of weighted requirements and utilized Value Oriented Prioritization (VOP) method chose from the past test for requirements regarding spread GSE elements. In the proposed system a procedure of assessment of the prioritization is connected, if an issue happened during the time spent requirement prioritization the procedure must be rehashed.

5. CONCLUSION

The generalization of this research is restricted because of the little specimen and the particular connection. A genuine task has requirement's interdependencies, and time and spending plan weight to consider, which cause the choice making to be much more troublesome. However, we trust that VOP is legitimate as prioritization strategy. The fundamental weakness of the test being the trouble to sum up to mechanical activities, it is significant to attempt the examination out for a situation study. The taking an interest association would then get information about prioritization and maybe discover a system that suits their needs.

A few procedures are produced for requirement prioritization in light of positioning, dole out qualities to the requirements or clients. A few strategies comprise of voting framework for requirement prioritization and a few methods embraced pair to combine examination process. In this study it is completely contemplated that every one of these methods are not secured the GSD/DSD aspects in productive conduct. A few strategies are produced for global software development perspective yet there is numerous streams and time deferral elements. The development organization's principle goal is to created software or product inside of time spending plan to win benefit. The greater part of these current systems are manual, and these cannot be actualized without adjustments. The methodology for requirement prioritization is in light of worth/weight of both stakeholders and requirements. In this research the author diminished the collaboration of human. At first, human allocate default qualities to stakeholders on the premise of their stereotyping. Requirement's quality/weight relies on upon variables like importance, desperation, expense, time, advantage, danger, market and so on and also on stakeholder's worth components.

Another structure has been proposed utilizing VOP method by considering the evoked requirements from distinctive stakeholders conveyed at different areas. As indicated by this system the requirements are evoked first and after that values are relegated to the diverse locales, stakeholders and their requirements. The procedure is led in six defined steps.

REFERENCES

Ahmad, A., M. Goransson and A. Shahzad. 2010. Limitations of the Analytic Hierarchy Process Technique with Respect to Geographically Distributed Stakeholders. In *Proceedings of World Academy of Science, Engineering and Technology,* 4(9):97-102.

Azar, J., R. K. Smith and D. Cordes. 2007. Value-Oriented Requirements Prioritization in a Small Development Organization. *Software, IEEE,* 24(1):32-37.

Brackett, J. W. 1990. Software Requirements Technical Report SEI-CM-19-1.2. *Software Engineering Institute, Camegie Mellon University, USA,* pp:1-27.

Cohen, L. and L. Manion. 1980. Research Methods in Education. *Groom Helm Ltd. London,* pp:1-12.

Creswell, J. W. 1994. Research Design: Qualitative and Quantitative Approaches. *SAGE Publications London,* pp:1-22.

Herbsleb, J. D. and D. Moitra. 2001. Global Software Development. *IEEE Software,* 18(2):16-20.

Karlsson, J. and K. Ryan. 1997. Prioritizing Requirements Using a Cost-Value Approach. *IEEE Software,* 14(5):67-74.

Karlsson, J., C. Wohlin and B. Regnell. 1998. An Evaluation of Methods for Prioritizing Software Requirements. *Information and Software Technology*, *39*(14):939-947.

Kozinets, R. V. 2002. The Field Behind The Screen: Using Netnography for Marketing Research in Online Communities. *Journal of Marketing Research,* pp:61-72.

Leffingwell, D. and D. Widrig. 2003. Managing Software Requirements: A Use Case Approach. 2nd ed. *Addison-Wesley, Boston, USA,* pp:101-135.

Minhas, N. M. and A. Majeed. 2012. An Integrated Approach to Requirement Prioritization with Reference to Distributed Software Development. In *1st IEEE International Conference on Engineering Management,* pp:1-11.

Parviainen, P. 2012. Global Software Engineering: Challenges and Solutions Framework. *Doctoral dissertation, University of Oulu, Linnanma,* pp:1-234.

Pearson, A., R. Wiechula and C. Lockwood. 2005. The JBI Model of Evidence Based Healthcare. *Internationa Journal of Evidence Based Healthcare,* 3(8):207-215.

Ramzan, M. 2010. Intelligent Requirement Prioritization using Fuzzy Logic. *Doctoral dissertation, National University of Computer and Emerging Sciences*, pp:1-169.

Regnell, B., M. J. Host, P. Beremark and T. Hjelm. 2001. An Industrial Case Study on Distributed Prioritization in Market-Driven Requirements Engineering for Packaged Software. *Requirements Engineering,* 6: 51-62.

Richardson, I., V. Casey, F. McCaffery, J. Burton and S. Beecham. 2012. A Process Framework for Global Software Engineering Teams. *Information and Software Technology*, 54(11):1175-1191.

Saaty T. L. 1980. The Analytic Hierarchy Process: Planning, Priority Setting, Resources, Allocation. *McGraw-Hill, Inc*., pp:1-27.

Seyff, N., I. Todoran, K. Caluser, L. Singer and M. Glinz. 2015. Using Popular Social Network Sites to Support Requirements Elicitation, Prioritization and Negotiation. *Journal of Internet Services and Applications*, 6(1):1-16.

Siddiqi, J. and M. C. Shekaran. 1996. Requirements Engineering: The Emerging Wisdom. *IEEE Software*, 13 (2):15-19.

Smite, D., C. Wohlin, T. Gorschek and R. Feldt. 2010. Empirical Evidence in Global Software Engineering: A Systematic Review. *Empirical Software Engineering*, 15(1):91-118.

Standish, T. 1997. Data Structures in Java. *Addison-Wesley Longman Publishing Co., Inc.,* pp:110-136.

Sullivan, L. 1986. Quality Function Deployment: A System to Assure that Customer Needs Derive the Product Design and Production Process. *Quality Progress*, pp: 39-50.

Zowghi, D. and C. Coulin. 2005. Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In *Engineering and Managing Software Requirements*, pp:19-4