International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Software Requirement Parameter Optimization - A Review

Anika Bisht, Madan Kushwaha
Computer Science and Engineering
Bansal institute of engineering and technology, Lucknow, UP
anika.bisht95@gmail.com

Abstract: Software Engineering (SE) is one of the youngest engineering domains emerging and developing within past four decades or so. Still, a massive amount of research work has gone into shaping it the way we see it functioning today. As a result, we have an impressive knowledge repository to work with in the form of software development models, software engineering theories and practices etc. The aim of SE is to create software products, services or their artifacts in order to meet the requirements posed by stakeholders while meeting quality constraints imposed on them. In order to meet both these objectives, any software development derives its purpose and meaning from the requirements posed by various stakeholders. Requirement Prioritization and Classification is a very critical but often neglected area of requirement engineering. Experience has shown that without proper prioritization and classification of requirements presented by various stakeholders, the end product usually fails to meet its objectives optimally.

Keywords: Fuzzy Logic, Requirement Prioritization, Requirements engineering, RCF

1. Introduction:

Programming Engineering (SE) goes for making programming items or their relics in a manner that these meet the prerequisites postured by partners while satisfying quality requirements forced on them. Keeping in mind the end goal to meet both these destinations, any product advancement infers its motivation and significance from the necessities postured by different partners. Prerequisite Engineering is a set up area of information inside of programming designing which sets up practices and standards for powerful necessity elicitation, displaying, detail, documentation and so forth. One imperative however regularly ignored routine of programming necessity building is prerequisite prioritization. A few necessity prioritization procedures have been introduced by creators. These systems are both quantitative and subjective in their tendency. Some understood prerequisite prioritization procedures incorporate Analytical Hierarchy Process (AHP), Cumulative Voting, Numerical Assignment, Ranking, Theory W, Requirement Triage, Wieger's Method and so forth. What's more, there are a few different methods which we should examine in this paper. Necessity prioritization empowers us to comprehend the importance of prerequisites opposite the framework to be created and among prerequisites also. With necessity prioritization, we can distinguish the center zones which require the vast majority of our consideration so as to build up an item which ideally meets the prerequisites of the partners. In the vast majority of the circumstances, because of spending plan and time limitations, it gets to be difficult to actualize every one of the prerequisites postured by partners. Likewise the way of numerous undertakings is such that necessities are actualized in an organized situation. In both of these situations, we require prerequisite prioritization [1]. We can organize prerequisite to acknowledge which necessities can be deferred or changed so that other earnest necessities can be actualized and to what degree. We can likewise utilize necessity prioritization to figure out which prerequisites to be actualized in before stages or later stages. We have been working with a few supported undertakings amid our examination. These ventures are confronted with both of the aforementioned circumstances. We have discovered it critical to organize necessities in their actual sense keeping in mind the end goal to build up a significant and fruitful item.

2. Related Work:

In order to develop cost effective quality software, it is really necessary to select the right requirements from the set of all requirements. It would be more beneficial if they are grouped release wise. In this way, we can decide upon which requirements need to be focused in any particular release. One way to categorize the requirements is to prioritize them based on some parameters. But achieving this goal is not usually easy because there are issues that should be addressed properly. These issues include the selection of stakeholders whose suggestion should be used to give priority (value). Similarly, the aspect or criteria, which shall be used as parameters to assign priorities needs to be formalized. Issues such as organization's setup, market value as well as stakeholder's personalities and agendas need to be catered as well when performing requirement prioritization. Working in such a situation requires several tradeoffs to be reached between the different stakeholders in conflicting environments. Only then can we expect to reach to a proposed and agreed system which caters to the needs of stakeholders. The major objective of prioritization is to help on decision making process about every aspect of system development. This decision making process is important in all manifestations of daily life from very complex such as economics, military, politics, administration to as simple as running your daily life's chores. In software engineering, the principal significance of prioritization is in the fact that it is used in almost every phase its software project life cycle i.e., selecting a project, finalizing the requirements, designing and developing of modules, testing, implementation, and even in post implementation activities. However, the major limitation in prioritization process is its gradual difficulty as the subject expands its base criteria. In very simple form, a project needs no prioritization but with very simple variations, it becomes difficult to make a decision based on more than one criteria. When the number of alternatives reaches hundreds or even thousands, the decision making and prioritization becomes next to impossible. This gives rise to the fact that manual and human based prioritization is not a solution for modern day complex systems. A number of prioritization techniques are available in order to deal with different level of complexity. These techniques are in used in different domains for supporting decision making.

2.1 Benefits of Prioritization

Followings are some key factors that advocate including prioritization process in requirement engineering:

2.1.1 Constraint Driven strategy:

Time and budget constraints usually do not allow entertaining all requirements. Therefore, project manager can use prioritization as a tool to help him in selection of those requirements that can be implanted in a certain time and budget constraints.

2.1.2 Planning the Releases:

Prioritization helps the stakeholders decide the type of requirements of the system and categorize them in optimal requirements sets. Based on these optimal sets a project manager can plan the releases of the product. In its normal flow; system's core requirements will be implemented in first release and based on their priority values the next group of requirements will be implemented in successive releases.

2.1.3 Balancing benefits against cost:

Before starting implementation, key stakeholders should identify related requirements and their corresponding benefits. Similarly, each of these requirements shall be evaluated for implementation cost. Prioritization based on the benefit and related cost; may help the business to balance their benefits of each requirement against the cost of implementing it.

2.1.4 Negotiating conflicts:

Conflicts between stakeholders arise because their viewpoints for most of the requirements are different. Requirements prioritization often involves negotiation process to handle contradictory requirements. Therefore it also helps in resolving conflicts and disagreement between stakeholders.

2.1.5 Better Understanding:

Requirements prioritization is an expensive activity to some extent, because it requires time, stakeholders' involvement, experts' opinions meeting, etc. On the other hand it helps all the stakeholders to understand all the requirements more closely. At the prioritization stage, each requirement is evaluated thoroughly and their implementation feasibility is checked. If certain requirements are found not-feasible then alternatives can be discovered.

2.2 Prioritization Parameters

Requirements should be prioritized objectively; i.e., there must be some parameters that shall be used to assign values to each requirement. Following are some important parameters for prioritization:

2.2.1 Time:

Time is an important factor in any project management process. Therefore, it must be considered when planning the releases of the software. Mostly, time constraint helps in identifying those requirements that could be entertain in short duration and long duration.

2.2.2 Cost:

The estimation of cost of a requirement requires past experience. Usually the developing organization estimates cost which is often expressed in terms of man-hours. Prioritizing requirements with respect to cost helps the project to be within budget.

2.2.3 Penalty:

Penalty can be viewed as negative points that can be introduced in case a requirement is not fulfilled. In its normal way, core requirements have higher penalty value than those that are in second level to core requirements.

2.2.4 Risk:

Risk management is used to cope with both internal and external risks, therefore it should also be consider when planning requirements. Based on the risk impact for each requirement, risk level of the project should also be calculated.

2.2.5 Functionality

Functionality in RE relates to the work/s that the proposed system has been assigned to perform. Any system to be developed embeds within itself several functionalities which could be visualized as programmed representation of various requirements posed by stakeholders. Just as priority of requirements varies, so does the priority of functionalities of the system. That is one of the major reason why both in linear or iterative approaches, our focus is always to implement high priority functionalities first and lower ranked ones later. There are other parameters that can be considered when prioritizing requirements like financial benefit, strategic benefit, competitors, competence/resources, release theme, ability to sell, etc.

2.3 Requirement Prioritization Techniques: an Overview

As mentioned in the literature review, there are various requirement prioritization techniques. However, no evaluation of these techniques has been made so far so that their utility and relevance can be determined. We have extensively analyzed existing mechanisms as mentioned earlier [12, 13, 19]. In this section, we give a comprehensive overview of various requirement prioritization techniques.

2.3.1 Analytical Hierarchy Process (AHP)

AHP is a relative evaluation based measurable system to organize prerequisites for programming items. In the event that we have n number of necessities, AHP makes n x (n-1)/2 correlations at every pecking order level. All things considered, we are normally working with prerequisites which have numerous goals. AHP fills in as an effective method in these sorts of circumstances by making pair savvy correlation to ascertain relative esteem and cost of every necessity against the other one. This altogether expansive number of examinations makes the strategy less compelling as increment in number of correlations dependably happens at the rate of O(n2). AHP is viewed as a five stage strategy.

1. Establish completeness of requirements.
2. Apply the pair-wise comparison method to assess the relative value.
3. Apply the pair-wise comparison method to assess the relative cost
4. Calculate each candidate requirement's relative value and implementations cost, and plot each on a cost-value diagram.
5. Use the cost-value diagram as a map for analyzing the candidate requirement

Countless have been made in later past to decide the adequacy of AHP for prerequisites prioritization. Karlsson [13] has made various studies which have demonstrated the viability of this method in modern settings. In the meantime, some different studies [15] have discussed AHP as being troublesome, less effective and tedious. AHP can be considered as a profoundly refined and complex procedure which can build up prioritization at the level of individual necessities. Endeavors have been made to diminish the quantity of examinations. In any case, this has constantly improved the room for give and take. As we would like to think, this tradeoff is essential since a few correlations might very be required.

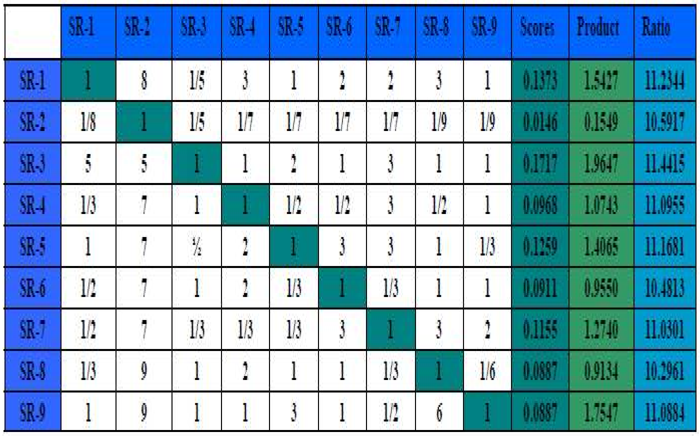
2.3.2 Cumulative Voting (CV)

Also referred to as 100 $ test or 100 point method sometimes, resembles very much with voting mechanism of brainstorming sessions. Each stakeholder is given 100 points that he or she can distribute among the requirements as they seem fit. It seems like very straightforward mechanism but it becomes complex as the number of requirements increases or the stakeholders involved become too many. This scheme also has several drawbacks associated with it. Firstly this scheme treats all the requirements as equal opportunity candidates. Secondly, the element of bias can never be over ruled. It has been observed that in second or subsequent voting, stakeholder assign more votes to their favorite requirements in order to move them up. Many researchers [24] have pointed out shortcomings in cumulative voting mechanism. Cumulative voting technique can also be considered as one which is complex in its nature but attempts to prioritize requirements at their individual level.

2.3.3 Numerical Assignment (NA)

It is presumably the most well-known prioritization method which is likewise simple to utilize. In the initial step, necessities are ordered into diverse gatherings. These necessities are given to every partner. Every necessity inside of these gatherings is doled out a number on a size of 1 to 5 by individual partners. The last calculating so as to position is dictated normal of all the positioning given to every prerequisite by each partner. This procedure due to its convenience has likewise been proposed by IEEE Std. 830-1998. Since the necessities are initially ordered into gatherings and afterward organized so we can say that this method does not organize prerequisites at the level of singularity. Rather one level of deliberation is presented. In spite of its wide materialness, this strategy additionally represents a few issues. Clear definition of the gatherings is one noteworthy disadvantage. Second issue is that even with clear definitions, partners will tend to put the vast majority of their prerequisites into basic gatherings due to their inclination (which can not be overruled). Another truth that we must be careful about is that inside of every gathering, every one of the necessities are at first at the same need level. The majority of these downsides in numerical task procedure have been very much recorded in [24].

Table 1: Sample Cost- Value Diagram for AHP



	SR-1	SR-2	SR-3	SR-4	SR-5	SR-6	SR-7	SR-8	SR-9	Scores	Product	Ratio
SR-1	1	8	1/5	3	1	2	2	3	1	0.1373	1.5427	11.2344
SR-2	1/8	1	1/5	1/7	1/7	1/7	1/7	1/9	1/9	0.0146	0.1549	10.5917
SR-3	5	5	1	1	2	1	3	1	1	0.1717	1.9647	11.4415
SR-4	1/3	7	1	1	1/2	1/2	3	1/2	1	0.0968	1.0743	11.0955
SR-5	1	7	½	2	1	3	3	1	1/3	0.1259	1.4065	11.1681
SR-6	1/2	7	1	2	1/3	1	1/3	1	1	0.0911	0.9550	10.4813
SR-7	1/2	7	1/3	1/3	1/3	3	1	3	2	0.1155	1.2740	11.0301
SR-8	1/3	9	1	2	1	1	1/3	1	1/6	0.0887	0.9134	10.2961
SR-9	1	9	1	1	3	1	1/2	6	1	0.0887	1.7547	11.0884

2.3.4 Ranking

This method is more suitable in nature where a solitary partner is included. On the off chance that there are n number of prerequisites, these necessities are positioned from 1(most huge) to n (slightest noteworthy). This positioning is select in its temperament on the grounds that prerequisites are not positioned with respect to different necessities similar to the instance of AHP or total voting. Different methods like air pocket sort, brisk sort or twofold inquiry procedures can be utilized to accomplish this positioning. There are two noteworthy downsides connected with this procedure. To start with significant issue is that it can bring about a greater number of contentions than assentions when connected in a domain of numerous partners. The second downside is that necessities are seen and positioned in segregation. The effect of one necessity over the other doesn't assume any part in general prioritization. Since necessities can have different measurements to them so scientists have conceived an instrument of joining these measurements and ascertaining a mean need for every prerequisite [25]. This adjustment has its own particular confinements and additionally has been appeared in [25].

2.3.5 Top-Ten requirements

This technique prioritizes only the most important requirements into a set of top-ten from a larger set of requirements. Selection of the most important requirements is subjective to the project environment and so it can be erroneous if based on human judgment. Since we create only a set of top-ten requirements, no prioritization within this set takes place. This can be termed as a shortcoming in many situations. The technique can be applied in conjunction with other techniques to achieve better results. According to Lausen, it is mostly helpful in situations where there are multiple stakeholders with uniform or similar significance.

2.3.6 Theory

The fundamental defender of this hypothesis is Dr. Barry Boehm who presented this idea [13] in 1989. Prominently known as Win-Win demonstrate, this procedure depends vigorously on transaction to determine any distinctions of conclusion among different partners. The transactions are led in a manner that every partner is in a "Win" circumstance. The standards of this system are advancement taking into account predefined arrangement, hazard appraisal and danger taking care of. In this system, clients are requested that rank their necessities before genuine transactions begin. Clients are asked to painstakingly consider which prerequisites they are willing to arrange and which they are definitely not. Hypothesis W has been a dynamic region of exploration among researchers which has been connected in necessity designing as well as in different spaces of programming building. Hypothesis W is a noteworthy constituent of Value Based Software Engineering (VBSE) plan and standard too.

2.3.7 Planning Game (PG)

This specific requirement prioritization technique is very suitable to extreme programming. In this specific technique, requirements are prioritized in consultation with customers. This is a variation of numerical assignment technique as discussed. However it offers more flexibility than numerical assignment where users are asked to essentially divide the requirements into three groups. Some other new and innovative techniques to emerge recently include Requirement Triage (RT) [13] and Wieger's Method (WG) [14]. In requirement Triage, each requirement is prioritized relative to the resources that are necessary to meet that specific requirement. In this way, a subset of requirements is selected which can optimize the probability of success of product while using the allocated resources efficiently. In Wieger's method, the priority of each requirement is set by determining the utility of that requirement to the customer as well as penalty incurred by the customer if that requirement remains unfulfilled. In this section, we have presented a brief overview of existing requirements prioritization techniques. In this next section, we shall present a theoretical evaluation of these techniques as well as present the brief idea of our proposed and implemented approach.

2.4 Literature Review:

Necessity Engineering (RE) is one of the soonest and extremely basic periods of programming designing. RE as a learning stream is fundamentally gone for securing, displaying and documentation of prerequisites for the product item. Necessity Engineering is an one of a kind order as in it consolidates the ideas of building as well as of human and sociologies. At times, alluded to as necessities investigation, RE is dealt with as a sub control of framework designing and programming building. Necessity designing means to characterize decisively the prerequisites that should be met. This is not a standard undertaking. As indicated by Fred Brooks, choosing what should be manufactured is the most troublesome piece of programming improvement. We can envision one programming prerequisite as one archived need that product item ought to fulfill. Typically necessities are delegated either as procedure based and item based or useful and non-practical prerequisites. Programming prerequisite can best be characterized as the portrayal of framework usefulness alongside its quality concerns. Prerequisite prioritization is the following coherent errand once necessities have been elicitated and legitimately dissected. By and large, it is truly hard to meet every one of the prerequisites that have been given by different partners. The vast majority of the times, elicitated necessities are dubious, clashing or out rightly false. Over timeframe, as our comprehension of the framework turns out to be more clear, the prerequisites begin accomplishing their real or particular shape. Likewise, by and large, necessities are executed in an amazed manner. In such circumstances, it gets to be critical to organize the prerequisites in an organized request to add to the framework in more practical way. This assignment turns out to be much more troublesome when performed right on time in the lifecycle. One of the best issues of programming specialists is advancement of such an item which doesn't fulfill the needs and desires of partners. In this way, numerous different analysts underlined upon the hugeness of necessity prioritization. Prioritization of necessities is a critical issue where expressed that prioritization of prerequisites was one noteworthy subject of dialog amid the overview that they attempted.

There are different systems for organizing necessities. Some significant strategies are Analytical Hierarchy Process (AHP), Binary Search Trees, 100 focuses strategy, arranging diversion, numerical task method and hypothesis W 20 and so forth. The accord of these studies is that the venture's prosperity or disappointment is emphatically dictated by how successfully, we can organize the necessities.

Computational Intelligence and delicate registering are built up methods which have determined numerous true issues. These systems incorporate Artificial Neural Networks, Fuzzy Logic, and Evolutionary Computing and so forth. Fluffy rationale is a system fixated on fluffy set hypothesis. Therefore it is considered as an augmentation of traditional set hypothesis. The idea of fluffy sets as presented by Lotfi Zadeh can be considered as speculation of the traditional sets which are fresh in their temperament. The motivation behind fluffy rationale is to both lessen the many-sided quality of existing arrangements and in addition expanding the availability of control hypothesis. Computational insight based procedures including fluffy rationale have been broadly utilized as of late to handle numerous genuine issues. Fluffy rationale has additionally discovered its way in programming designing where it has most as of late been utilized as a part of exertion estimation, programming undertaking closeness, programming advancement, venture assessment, programming development and so forth. Fluffy rationale has been utilized as a part of necessity designing also for different assignments. Chengdong Li et al. displayed a novel methodology for utilizing former learning and tested information in fluffy frameworks. Some other late advancements in the space of fluffy rationale have likewise exhibited new vistas of examination in programming designing. What is apparent subsequent to concentrate broadly is the way that scientists in programming designing need to apply counterfeit consciousness in different spaces of learning of SE to propose such methods which are evolvable and can insightfully create proficient results. In this paper, we have proposed another use of fluffy rationale in programming building. We have proposed to utilize fluffy rationale in the space of necessity building. We propose acquainting fluffy rationale with decide the need of necessities. The following area is given to elaboration of this proposed procedure.

3. Conclusion:

Requirement prioritization is one important activity of requirement engineering phase in software development. There are various requirement prioritization techniques in literature and practice. However, no significant comparative evaluation of these techniques has been made so far. In this paper, we conclude another application of fuzzy logic in the domain of requirement engineering. We suggest introducing fuzzy logic to determine the priority of requirements with the integration of Bayesian network.

References:

[1] F. Brooks, No silver bullet: Essence and accidents of software engineering, IEEE Computer, vol.20, no.4, pp.10-19, 1987.

[2] J. Karlsson and K. Ryan, Supporting the selection of software requirements, Proc. of the 8th International Workshop on Software Speci_cation and Design, 1996.

[3] J. Karlsson and K. Ryan, A cost-value approach for prioritizing requirements, IEEE Software, vol.14, no.5, pp.67-75, 1997.

[4] X. Liu, C. C. Veera, Y. Sun, K. Noguchi and Y. Kyoya, Priority assessment of software requirements from multiple perspectives, Computer Software and Applications Conference, vol.1, pp.410-415, 2004.

[5] L. Fellows and I. Hooks, A case for priority classifying requirements, The 3rd International Conference on Requirements Engineering, pp.62-65, 1998.

[6] E. Yourdon, Death March Projects, Prentice Hall, 1997.

[7] M. Lubars, C. Potts and C. Richter, A review of the state of the practice in requirements modeling, Proc. of the IEEE Intemational Symposium of Requirements Engineering, pp.2-14, 1993.

[8] V. Ahl, An Experimental Comparison of Five Prioritization Methods, Master Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden, 2005.

[9] T. Saaty, The Analytic Hierarchy Process: Planning, Priority Setting, Resource, Allocation McGraw-Hill, New York, 1980.

[10] T. L. Saaty and G. Hu, Ranking by eigenvector versus other methods in the analytic hierarchy process, Applied Mathematical Letter, vol.11, no.4, pp.121-125, 1998.

[11] F. Hartwich, Weighting of agricultural research results: Strength and limitations of the analytic hierarchy process, Research in Development Economic and Policy, Discussion Paper, Grauer Verlag, Stuttgart, no.9, 1999.

[12] F. Hivert, J. Novelli and J. Thibon, The algebra of binary search tree, Theoretical Computer Science, vol.339, no.1, pp.3-10, 2005.

[13] L. Xiang, K. Ushijiam, T. Zhao, T. Zhang and C. Tang, O(1) time algorithm on BSR for constructing a random binary search tree, Proc. of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies, pp.599-602, 2003.

[14] I. Al-furaih, S. Aluru, S. Goil and S. Ranka, Parallel construction of multidimension binary search tree, IEEE Trans. on Parallel and Distributed Systems, vol.11, no.2, pp.136-148, 2000.

[15] C. Lee, L. Hung, M. Chang, C. Shen and C. Tang, An improved algorithm for the maximum agreement subtree problem, Information Processing Letters, vol.94, no.5, pp.211-216, 2005.