Available online at www.elixirpublishers.com (Elixir International Journal)

Electrical Engineering

Elixir Elec. Engg. 144 (2020) 54571-54573

ISSN: 2229-712X

A Comparison of Requirement Prioritization Methodologies

K.Jeyaganesh kumar[1] and V. Meena[2]

[1]Department of Computer Science and Engineering, Anna University, Chennai, India.

[2]Department of Computer Science and Engineering, AVS Engineering College, Salem, India.

ARTICLE INFO

Article history:

Received: 11 June 2020;

Received in revised form: 10 July 2020;

Accepted: 20 July 2020;

Keywords

Software Engineering, Requirements Engineering(RE), Requirement Prioritization.

ABSTRACT

Software Engineering upholds the productivity of quality software products that meets the customer requirements. Requirements engineering ensures the customer satisfaction through the selection of requirements through elicitation processes. Requirement elicitation promotes prioritizing a set of requirements that enables the stakeholders for conformance to the requirements to be built. Requirement prioritization process solves critical decision making problems towards the selection of requirements among group of inappropriate requirements. The main objective of the comparison of the prioritization techniques concludes the differences between them in terms of cost and time. This paper compares and elaborates the prioritization technique that provides a deep study of their constraints, objectives and selection basis of the requirements.

Introduction

In software development, it is often a challenge for people to pick the 'right' requirement among several or many options if it's not obvious which requirement is desirable. Requirements prioritization helps people to discover the most desirable requirements. It seems that most requirements prioritization techniques work well on alittle number of requirements, but many of them have constraints on medium to large numbers of requirements. Requirement prioritization process is employed to work out which candidate requirement of a software project should be included during a certain release, for this purpose different techniques are used. These techniques use different approaches that consider various factors for prioritization e.g. cost, value, risk, benefit etc.

Requirements prioritization is a crucial activity in software development. Usually, the amount of requirements from the purchasers exceeds the amount of features which will be implemented within the given time and available resources. For that reason, a number of the requested features won't be fulfilled or they're moved to later releases.

Therefore, the customer and the development teams must decide what's the foremost essential functionality which should be implemented as early as possible. In other words, the stakeholders should prioritize the wants. There are several different techniques presented within the literature the way to prioritize requirements. It would be difficult to select the foremost suitable method due to the massive number of them. Some methods are longer consuming than others but provide more accurate results. Some methods scale well to be used with larger number of requirements but provide very coarse results. In other words, none of the techniques can really be considered the best one but a practitioner must pick a way that's the foremost suitable for his situation, for instance, in terms of scalability, accuracy and time consumption.

Features of Requirements Prioritization

Stakeholders can prioritize requirements to seek out which requirement is most vital to them. However, the word "importance" are often a multifaceted concept which can have different meanings to different people. For example, importance could mean high market price, top quality of the merchandise, or urgency of implementation among other things. It is essential to specify the meaning of "importance" first to scale back the likelihood of confusion when letting the stakeholders prioritize the needs.

Time is often the time spent on successfully implementing the candidate requirement. Time can also be influenced by other factors like degree of parallelism in development, or staff training time.

Cost is often the cash spent on successfully implementing the candidate requirement. Cost is often directly influenced by staff hours. Cost also can be influenced by other factors like the additional resources needed so as to implement the needs.

Penalty is what proportion must be paid if a requirement isn't fulfilled. Penalty is a crucial aspect that must be evaluated. Sometimes requirements may have low values, but failing to satisfy these requirements may cause a high penalty.

Change in one aspect may end in a change in another aspect. Since each aspect may have an influence on the degree of successful of the ultimate product, it is essential to think about multiple aspects so as to extend the degree of success of the ultimate product. Several aspects are often considered when prioritizing requirements, but generally it is not practical to think about all the aspects. Which aspects should be considered depends on the important situation.

Techniques of Requirements Prioritization

I. AHP

AHP is developed by Saaty and it's designed for complex deciding. The thought of AHP is that it compares all possible pairs of hierarchical requirements to work out the priority. When using AHP, the user first identifies the attributes and alternatives for every requirement and uses them to create a hierarchy. Then the user specifies his/her preference to every pair of the attributes by assigning a preference scale which is usually 1to9, where 1 indicates equal value and 9 indicates extreme value. After that AHP converts the user's evaluations to numerical values and a numerical priority is derived for each element of the hierarchy. Note that a redundancy might exist when using the AHP method to prioritize requirements, therefore a consistency ratio should be calculated after using the AHP method to judge if the prioritization is valid. If n requirements need to be prioritized, n*(n-1)/2 pair-wise comparisons are required when using the AHP method. Therefore the complexity of AHP is O (n^2).

II. Hierarchy AHP

Hierarchy AHP, which is introduced by Karlsson et al. uses the AHP method to prioritize requirements only at an equivalent level of hierarchy. This method can reduce the amount of selections compared with the AHP method, since not all the wants are compared pair-wise. this will reduce the amount of redundant comparisons, but the trade-off is that the power to spot inconsistent judgments is additionally reduced.

III. Minimal Spanning Tree

Minimal spanning tree is another prioritization method which is introduced by Karlsson et al. the thought of minimal spanning tree method is that if the choices are made perfectly consistent, the redundancy won't exist, and during this case the amount of comparisons will reduce to only n-1 comparisons (n is that the number of requirements). A minimal spanning tree constructs unique pairs of requirements. it's a directed graph which is minimally connected. Minimal spanning tree can reduce the amount of pair wise comparisons dramatically compared with AHP.

IV. Bubble Sort

The idea of the bubble sort method for sorting requirements is that the users compare two requirements at a time and swap 18 them if the 2 requirements are within the wrong order. The comparisons continue until no more swaps are needed. The results of bubble sort may be a list of ranked requirements. the typical and worst case complexity for bubble sort is O(n^2).

V, Binary search tree

The idea of the binary search tree method for ranking requirements is that each node represents a requirement, all requirements placed in the left subtree of a node are of lower priority than the node priority, and all requirements placed in the right subtree of a node are of higher priority than that node priority. When performing the binary search tree method, first choose one requirement to be the top node. Then, select one unsorted requirement to compare with the top node. If that requirement is of lower priority than the top node, it searches the left subtree, but if that requirement is of higher priority than the top node, it searches the right subtree. The process is repeated until no further node needs to be compared and at that time the requirement can be inserted into the right position. The average complexity for binary search tree is O (n log n).

Techniques Evaluation

AHP can provide the foremost reliable results of the six methods, but it requires the most important number of selections and therefore the longest time consumption. Minimum spanning tree involves the littlest number of selections and therefore the shortest amount of your time consumption, but it provides the smallest amount reliable result and therefore the lowest fault tolerance. Bubble sort is that the simplest way to use and it can provide relatively reliable results and comparatively good fault tolerance, but it involves the most important number of selections (same as AHP).

Hierarchy AHP and binary search tree reside within the middle. They produce less reliable results than AHP and bubble sort, but also take fewer decisions and fewer time to perform than AHP and bubble sort. it's seen that nobody prioritization method is ideal among these six methods. Minimum spanning tree requires less effort and time to perform the prioritization process, but it contains a high risk of misdirecting project resources and time since it provides low reliable results. AHP and bubble sort methods can provide reliable results, but they have large amounts of effort and time to perform. When handling a little number of requirements, the quantity of effort and time spent are often relatively small. But since the complexity of the AHP and bubble sort methods is high (both are O (n^2)), when handling large numbers of requirements the quantity of effort and time spent may become unmanageable. Karlsson et al. also admit that AHP and bubble sort both contain a proportion problem. Among the six prioritization methods, it seems that no method is ideal for giant numbers of requirements (AHP and bubble sort contain a proportion problem, and other methods contain a particular degree of accuracy problems).

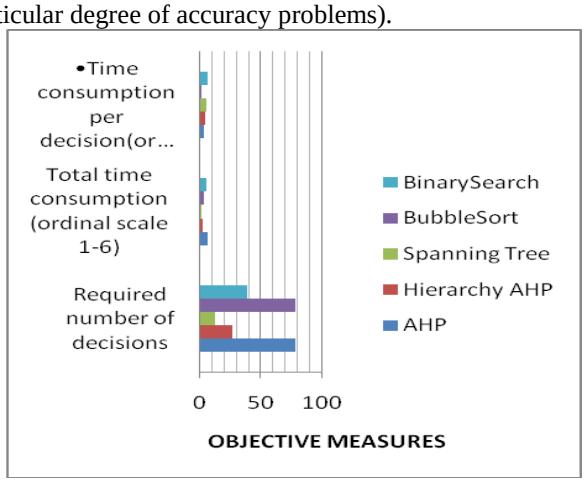

Figure 1. Objective Measures (Karlsson Et Al.,)

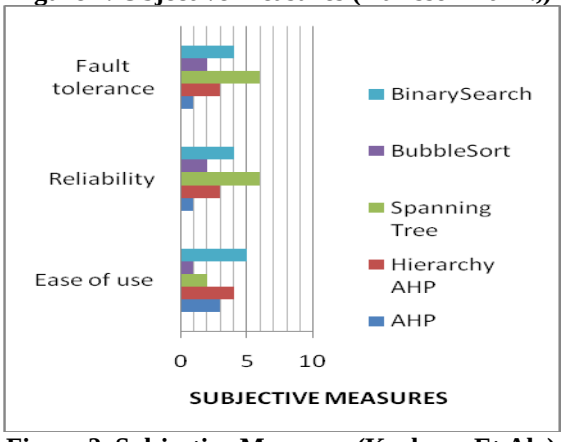

Figure 2. Subjective Measures (Karlsson Et Al.,)

Conclusion

It is almost impossible to implement all the wants in one release. Some kind of prioritization process is required to implement the foremost important requirements within the first release and leave the smaller ones for the longer term releases. Requirements prioritization helps requirement engineers during this complex and crucial decision making situation. This paper, we introduced five basic requirements prioritization techniques: Bubble sort, Binary search tree, AHP, Hierarchy-AHP, Minimal spanning tree.

References

[1]. L. Lehtola, M. Kauppinen, and S. Kujala, "Requirements prioritization challenges in practice," in Proc. of 5th Intl Conf. On Product Focused Software Process Improvement (PROFES), 2004, pp. 497–508.

[2]. P. Berander, K. Khan, and L. Lehtola, "Towards a research framework on requirements prioritization," in SERPS06: Sixth Conference on Software Engineering Research and Practice in Sweden, 2006, pp. 39–48.

[3]. A. Perini, F. Ricca, A. Susi, and C. Bazzanella, "An empirical study to compare the accuracy of ahp and cbranking techniques for requirements prioritization," in CERE '07: Proceedings of the 2007 Fifth International Workshop on Comparative Evaluation in Requirements Engineering. Washington, DC, USA: IEEE Computer Society, 2007, pp. 23–35.

[4]. P. Berander and P. J¨onsson, "Hierarchical cumulative voting (hcv) prioritization of requirements in hierarchies," International Journal of Software Engineering & Knowledge Engineering, vol. 16, pp. 819–849, 2006.

[5]. P. Berander and M. Svahnberg, "Evaluating two ways of calculating priorities in requirements hierarchies - an experiment on hierarchical cumulative voting," J. Syst. Softw., vol. 82, no. 5, pp. 836–850, 2009.

[6]. J. Karlsson, "Software requirements prioritizing," in Requirements Engineering, 1996., Proceedings of the Second International Conference on, 15-18 1996, pp. 110 –116.

[7]. A. S. Danesh and R. Ahmad, "Study of prioritization techniques using students as subjects," in ICIME '09: Proceedings of the 2009 International Conference on Information Management and Engineering. Washington, DC, USA: IEEE Computer Society, 2009, pp. 390–394.

[8]. S. Hatton, "Early prioritisation of goals book series," in Advances in Conceptual Modeling Foundations and Applications, vol. 4802, 235- 244 2007, pp. 517 –526.

[9]. Leoucopoulos P. and Karakostas V. "System Requirements Engineering", Mc Graw-Hill, 1995

[10]. Martín A., Martínez C., Martínez Carod N., Aranda G., and Cechich A. "Classifying Groupware Tools to Improve Communication in Geographically Distributed Elicitation". IX Congreso Argentino en Ciencias de la Computación, CACIC 2003, La Plata, 6-10 Octubre 2003, (942-953).