



PHandler: An expert system for a scalable software requirements prioritization process



Muhammad Imran Babar, Masitah Ghazali^{*}, Dayang N.A. Jawawi, Siti Maryam Shamsuddin, Noraini Ibrahim

Faculty of Computing, UTM, Johor 81310, Malaysia

ARTICLE INFO

Article history:

Received 29 March 2014
Received in revised form 13 February 2015
Accepted 9 April 2015
Available online 20 April 2015

Keywords:

Software requirement prioritization
Artificial neural networks
Expert systems
Hybrid model
Decision support

ABSTRACT

Software requirements engineering is a critical discipline in the software development life cycle. The major problem in software development is the selection and prioritization of the requirements in order to develop a system of high quality. This research analyzes the issues associated with existing software requirement prioritization techniques. One of the major issues in software requirement prioritization is that the existing techniques handle only toy projects or software projects with very few requirements. The current techniques are not suitable for the prioritization of a large number of requirements in projects where requirements may grow to the hundreds or even thousands. The research paper proposes an expert system, called the Priority Handler (PHandler), for requirement prioritization. PHandler is based on the value-based intelligent requirement prioritization technique, neural network and analytical hierarchical process in order to make the requirement prioritization process scalable. The back-propagation neural network is used to predict the value of a requirement in order to reduce the extent of expert biases and make the PHandler efficient. Moreover, the analytical hierarchy process is applied on prioritized groups of requirements in order to enhance the scalability of the requirement prioritization process.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Software requirement prioritization is a process in which requirement engineers find the most important stakeholders' requirements in order to develop a system [65], particularly an innovative system. These core requirements must be implemented within the defined constraints of time, resources, cost and quality, in order to satisfy the customers [2,34,64,71,73]. The software requirement prioritization process helps to identify conflicts among various requirements and ultimately helps in the resolution of these conflicts; it also provides a future road map [72]. Nonetheless, the selection of the right requirements is the main challenge in order to satisfy all the key needs of the stakeholders and maximize the business value of the product [60]. The wrong set of requirements for an innovative system leads to an increased cost in the modification of the system. The wrong requirements also have an adverse effect on the quality of the system. The value

of the requirements is calculated by using a suitable requirement prioritization technique. For decision making, the prioritization of requirements is also considered as highly significant [7]. The requirement prioritization process is also a very complex decision-making process [21,33,47,52]. In order to apply requirement prioritization techniques, the experts must have sound knowledge of the domain and possess the relevant professional skill set [40].

The satisfaction of the stakeholders is taken into account when a quality software system fulfills all the requirements of the stakeholders or users [18,49]. The consideration of all requirements in the development of a software system is not possible because of the constraints such as the time to market, budget, and other resources. Thus, only important requirements are taken into account in a single release [35]. Software requirements have different features such as risk [7,52,64,71], importance [47], volatility [52], cost, penalty and time [71] and dependencies with other requirements based on cost, technical importance or value, customers and changes in the requirements [24,61]. On the basis of the above-mentioned features, the unimportant requirements are not given consideration; rather they are totally rejected. Only value-added requirements are considered in the requirements set in order to develop a valuable system of high quality. The value of the system may be in terms of profit, efficiency, the services,

^{*} Corresponding author. Tel.: +60 7 553 8801 (office), +60 19 669 3103 (mobile); fax: +60 7 556 5044/557 4908.

E-mail addresses: ibmuhammad2@live.utm.my (M.I. Babar), masitah@utm.my (M. Ghazali), dayang@utm.my (D.N.A. Jawawi), sitimariyams@gmail.com (S.M. Shamsuddin), noraini_ib@utm.my (N. Ibrahim).

good performance, correct data, fulfillment of the right user needs and acceptance by a wide community [12].

Different software requirement prioritization techniques are used in order to prioritize the most important requirements, and the requirement prioritization process is still immature [7]. Software engineers use multiple techniques on the basis of factors such as cost, time, and relevant importance, but such an approach is the cause of certain conflicts. These conflicts are initiated due to the effect of one aspect on another aspect. For instance, cost has an immense effect on the requirement prioritization process. If the cost of an important requirement is high, then there is a possibility that the stakeholders may change their mind about the requirement. Such a change in the mind of the customer also results in the change of the priority of that very requirement [43]. In the software requirement prioritization process, the requirements' attributes or aspects are directly affected by the role of the customers or stakeholders involved in the development process of an innovative system. In addition, it is pointed out that different stakeholders perceive the term "prioritizing requirement" differently [27].

This paper comprises eight major sections. This paper first describes the research background in Section 2. Section 3 is about the value-based intelligent requirement prioritization (VIRP) technique. Section 4 describes the proposed expert decision support system PHandler followed by a description of the experimental setup in Section 5. Section 6 discusses the experiments, while Section 7 presents the experimental results. Section 8 presents an overview of the related works and discussion. Lastly, Section 9 concludes the paper.

2. Research background

Different software requirement prioritization techniques are used to select the user requirements in order to develop systems of high quality. Some of the most prominent software requirement prioritization techniques are the analytical hierarchical process (AHP), binary search tree, ranking, the numerical assignment, top ten requirements, cumulative voting or hundred dollar test, and many different hybrid techniques are presented. Different methods are used to prioritize the requirements. In one study, the researchers applied different software requirement prioritization techniques "to prioritize 13 well-defined quality requirements on a small telephony system" [38]. The results obtained from the different prioritization techniques demonstrated that the bubble sort technique and AHP are more reliable than the other techniques. It is also noted that in the case of bubble sort and AHP, the time consumption is very high due to the rising number of decisions. The implementation of the AHP is complex, and in it the pairwise comparisons consume too much time for larger datasets. Karlsson et al. [38] conclude that the AHP and bubble sort are problematic for larger projects, and these techniques face the issue of scalability. In a research study, it is concluded that the AHP is not suitable for resolving the problem of scalability due to the exponential increase in the number of comparisons [44]. Laurent et al. [44] stated that the existing techniques are suitable for smaller or medium sized projects, and they are not suitable for larger projects with a large number of requirements. The AHP is only suitable for a small number of requirements [46]. Some of the existing software requirement prioritization techniques are described as follows:

2.1. Analytical hierarchy process

Due to the high reliability of the results, the AHP is used in this research. The AHP is a statistical assessment technique proposed by Saaty [62]. The AHP is used to determine the best alternative,

to set priorities and to allocate the resources [62]. Saaty [62] did not present the AHP for requirement prioritization; rather it was presented to solve the problems in the domains of social science, economics and management sciences. In this research, the AHP is applied for requirement prioritization. For n number of software requirements, the following formula is used to calculate the number of comparisons:

$$n \times (n - 1)/2$$

If there will be 25 requirements in a software project, then the total number of comparisons will be 300. In this research, the divide and conquer approach is applied on the requirements dataset based on some key aspects. The smaller clusters of the requirements are gathered which comprise competing requirements. In order to resolve such a situation, the AHP is chosen. The AHP is then applied on these clusters in order to provide a scalable solution for projects with a large number of requirements. The technique is suitable for software requirements with different dimensions or goals; for example, time, significance, associated risks, accuracy and benefits. The AHP works in an efficient and reliable way to solve the problem of requirement prioritization for projects with a small number of requirements. A complete description of the steps of the AHP is given in Section 5.2. Table 1 describes the pairwise comparison method of the AHP.

2.2. Hierarchy AHP

The Hierarchy AHP (HAHP) is introduced in order to solve the scalability issue of the AHP [38]. As the number of requirements increases in the AHP then the number of comparisons also increases. An increase in the number of comparisons reduces the efficiency of the AHP. Moreover, the AHP is only suitable for projects with few requirements and is not suitable for projects with large-scale requirements. Hence, in order to overcome this deficiency of the AHP the new approach HAHP is introduced. However, the HAHP is difficult to apply and is less reliable as compared to the AHP.

2.3. Minimal spanning tree

The Minimal Spanning Tree (MST) is presented in order to reduce the pairwise comparisons among requirements [38]. The redundant comparisons are eliminated, and the number of comparisons is reduced to $n - 1$ as compared to the AHP. The MST claims that the reduced number of comparisons is sufficient to measure the relative importance of the requirements.

2.4. Cost-value approach

The cost-value approach is presented in order to prioritize the requirements [37]. This approach focuses on the relationship between requirement value and implementation cost. The term

Table 1
AHP comparison method.

Importance	Description
1	Equal importance
3	Moderate difference in importance
5	Essential difference in importance
7	Major difference in importance
9	Extreme difference in importance
2, 4, 6, 8	Intermediate values between
Reciprocals	If requirement i has one of the above numbers assigned to it when compared with requirement j , then j has the reciprocal value when compared with i

Download English Version:

<https://daneshyari.com/en/article/402290>

Download Persian Version:

<https://daneshyari.com/article/402290>

[Daneshyari.com](https://daneshyari.com)