**Requirement Prioritization Based on Non-Functional Requirement Classification Using Hierarchy AHP**

Thant Z. Win, Rozlina Mohamed and Jamaludin Sallim  
Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, 26300 Kuantan Pahang Malaysia  
E-mail: rozlina@ump.edu.my

**Abstract.** Requirement prioritization is a process in requirement engineering, which is a part of software development life cycle (SDLC). Requirement is prioritized due to constraints such as budget, time and resource allocation. Requirements of software is often classified as functional requirements (FR), and non-functional requirement (NFR). In order to produce a high-quality software, both requirement must be considered during requirement prioritization process. Various prioritization techniques have been invented, and Analytical Hierarchical Prioritization (AHP) is the most popular technique that has been cited. However, AHP does not support the NFR and unscalable. Meanwhile, Hierarchy-AHP has been introduced unto increase the scalability of AHP by using hierarchical requirements as input. Nevertheless, hierarchy-AHP does not meant for NFR and experimental result for increasing the scalability is not received significant attention. Thus, we intend to use NFR with large dataset on hierarchy-AHP. Aim of this paper is an exploration of hierarchy-AHP experimenting on RALIC dataset. Our major findings are: (i) NFR can be used hierarchy-AHP with minor process amendment, and (ii) hierarchy-AHP able to reduce pairwise comparison which is up to 97.33% for 403 number of requirements, compared to original AHP.

**Keyword.** Requirements prioritization, Non-functional requirement, Analytic Hierarchy Process (AHP), Hierarchy AHP.

1. **Introduction**

Requirement engineering is one of the most important and fundamental stage in SDLC in any software process model. Requirement engineering is the process to collect the software requirement from user, analyses and document are known as requirement engineering. Requirements prioritization is an activity for identifying the most important requirements for a system [1]. Requirement prioritization is in either software validation or software analysis process which is part of software requirement engineering [2]. Determining the accurate requirement which is selected from massive amount requirements is a crucial step. Requirement is prioritized based on many aspects such as significance, cost, time, risk and volatility. Various scale can be used in prioritization activity such as a ratio scale or ordinal scale [3].
2. Related work

The related works in this research is divided into 4 sub-sections.

2.1. Analytic Hierarchy Process

Analytic Hierarchy Process (AHP) is initially developed by Prof. Thomas L. Saaty. AHP is widely known in software prioritization as a structured technique for organizing and analyzing requirements to be prioritized. Prioritization decision is made based on mathematics and psychology [5]. In this technique, all requirements will be compared in pairwise. Each comparison is compare based on a ratio scale [6]. Then, the ratio scale of the pairwise comparison is put on a matrix. The matrix is used to compute the relation significance of each requirement.

Thus, total number of pairwise comparison in AHP is up to \( n \times (n-1)/2 \), where \( n \) is number of requirements. Therefore, the entire pairwise comparison will dramatically increase proportion to the increment number of requirements [7]. As a result, AHP is *unscaleable* due to inappropriate for a large number of requirements. This is because, number of requirements stipulate the number of pairwise comparisons that need to be made [8].

2.2. Hierarchy AHP

Normally, requirements for a large scale software project are structured. Thus, structured requirements can be broken-down into hierarchy of interrelated requirements. The hierarchically structured requirements are arranged where the most generalized requirement is placed on top of the hierarchy, while the more specific requirements are place on the lower level of the hierarchy. This hierarchically structured requirements are becoming the input unto AHP in stages, based on the hierarchy level. In contrast with input unto original AHP, requirements are in flat structure. By using hierarchy AHP, not all requirements are compared in pairwise. The pairwise comparison is only occurred between requirements in the same hierarchy level.

Using a hierarchical AHP reduces the required number of pairwise comparison. As a result, hierarchy AHP is more scaleable than original AHP.

2.3. Hybrid Assessment Method

Hybrid Assessment Method (HAM) is used to support a prioritization ranking for NFR. It is used to aid software engineers to decide which qualities are more important to be satisfied and, in case of conflict, which should be prioritized over others and HAM recommend the best architecture by ranking architectural alternatives considering the stakeholder’s views [11]. HAM is an efficient method combines one single pairwise comparison matrix for a decision making to determine the weights of criteria with one typical weighted decision matrix to prioritize the alternatives [11]. The consistency problems associated to scale and the prioritization method, geometric scale is used in HAM. The geometric scale is used for evaluating the criteria and the geometric mean for defining the alternative rankings.

Advantages of HAM are: (1) consistent, (2) strong and (3) flexible. HAM allows trade-offs between criteria importance, for example; non-functional requirement resulting in the automatic determination of their relative importance (weights). Disadvantages of HAM are: (1) does not support group decision making, and (2) requires technical knowledge about the context domain for providing judgments for relative importance of criteria, in addition to provide appraisals for the alternatives [11].
2.4. Non-Functional Requirements

Non-functional requirements are the system properties and limitations for example reliability, performance usability, efficiency and space requirements. Non-functional requirement is expressed to increase the accuracy of FR. Three parts of NFR are: product requirements, organization requirements and external requirements [12]. NFR may affects the overall architecture of a system rather than the individual components. NFR is typically identify or limit characteristics of the system as an entire such as performance, security, or usability. Type of NFR are: Efficiency Requirement, Security Requirement, Dependability Requirements, Performance Requirements, Space Requirements, Portability Requirements, Modifiability Requirements, Reliability Requirements, and Usability Requirements [12]. Table 1 represented comparative for some requirement prioritization.

Table 1. Comparative of requirement prioritization techniques

| Requirement Prioritization Techniques | Advantages | Disadvantages | References |
|--------------------------------------|------------|---------------|------------|
| AHP                                  | It is use for complex decision making and usage for pairwise comparison to compute the relation importance of each requirement. By decreasing complex decisions pairwise comparison AHP supports decision makers to get best decision. | Time consuming for pairwise comparison when use the large amount of requirement. Thus this technique is unscalable. | [5, 7, 13, 14] |
| Hierarchy AHP                        | Reduce number of pairwise comparison | Requires initial works to disseminate requirements into hierarchical structured No experimental results reported. | [20, 21] |
| HAM                                  | Combining one single pairwise comparison decision matrix. Thus, it is simple, efficient, consistent, strong and flexible method. | Does not support group decision making. Requires technical knowledge about the context domain to provide the importance judgments for criteria compared. | [11] |

3. Methodology

Methodology in this research is consists of four main process, study the existing requirement prioritization techniques, pairwise comparison of non-functional requirement for hierarchy (a), and pairwise comparison at low level of requirement hierarchy (a), prioritized non-functional requirement for each functional requirement.
3.1. RALIC dataset

Dataset is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix. The RALIC project was a software development originated to substitute the existing access control systems at UCL and consolidate the new system with library access and borrowing. RALIC stands for Replacement Access, Library and ID Card. The RALIC project datasets were collected by Soo Ling Lim at the University College London (UCL) [4, 19].

3.2. Pairwise comparison

Analytic Hierarchy Process technique uses a pair-wise comparison matrix to compute the relative significance of each requirement. The total number of comparisons to make with AHP are \( n \times (n-1)/2 \) (where \( n \) is the number of requirements) at each hierarchy level. After making pairwise comparison for hierarchy \( (a) \) lowest to highest \( (a.1 \text{ to } a.3) \), calculate the average the crossing for every row of requirement and got the weights result for each requirement row prioritized list which was highest weight was first prioritized. Figure 1 represented the research methodology flow for this paper.

![Figure 1. Research methodology](image)

3.3. Consistency Ratio Check

The consistency of comparisons is assessed by using the Eigenvalue (Max) \( (\lambda) \) after all pairwise comparisons have been finished. A consistency index (CI) is then calculated with the eigenvalue, via the following equation 1: After getting pairwise comparison weights value for requirement, verify the consistency ratio check accuracy for pairwise comparison using Equation 1 of consistency index check for consistency ratio (CR). Where \( (\hat{\lambda}) \) is Eigenvalue (Max) and \( n \) is number of requirements.
\[ CI = \frac{(\lambda - n)}{(n - 1)} \]  
\[ CR = \frac{(CI)}{(RI)} \]

where
- CI is a Consistency index
- \(\lambda\) is an Eigenvalue (Max)
- n is a Number of requirements

4. Implementation

The implementation includes two parts, which are: comparison matrix for using RALIC dataset hierarchy (a) non-functional requirement, and part B is determine the non-functional requirement for hierarchy (a) for this study.

4.1. Pairwise Comparison Using Ralic Dataset Hierarchy (a) Non-functional Requirement

AHP pairwise comparison using RALIC dataset requirement hierarchy (a). Table 3 describes the ratio scale for AHP technique.

Step 1, compare the hierarchy (a) requirement number 45 requirements need to compare for pairwise comparison. Table 6 illustrates the comparison of RALIC dataset hierarchy (a) for non-functional requirement.

Step 2, when assign the actual value into upper triangular matrix, reciprocal value for lower triangular matrix automatically appeared in pairwise comparison table. Pairwise comparison for RALIC dataset hierarchy (a) illustrated in Table 6.

Step 3 comparison hierarchy (a), divide each element of hierarchy (a) the matrix with the sum of its column and total sum of each column is 1 and the normalized the row by row of the matrix by averaging across the rows, got the weights result.
Step 4 calculate the consistency measure for consistency index, and calculate the consistency ratio (CR).

**Table 3.** Ratio scale of analytic hierarchy process.

| Scale | Definition     |
|-------|----------------|
| 1     | Equal          |
| 3     | Moderate       |
| 5     | Strong         |
| 7     | Very strong    |
| 9     | Extreme        |
| 2, 4, 6, 8 | Intermediate values |

**Table 4.** Pairwise comparison for sub-hierarchy (a.1).

| a.1  | a.1.1 | a.1.2 | a.1.3 | a.1.4 |
|------|-------|-------|-------|-------|
| a.1.1| 1     | 3     | 5     | 3     |
| a.1.2| 0.33  | 1     | 4     | 7     |
| a.1.3| 0.20  | 0.25  | 1     | 9     |
| a.1.4| 0.33  | 0.14  | 0.11  | 1     |
| Sum  | 1.87  | 4.39  | 10.11 | 20    |

| a.1  | a.1.1 | a.1.2 | a.1.3 | a.1.4 | Weights |
|------|-------|-------|-------|-------|---------|
| a.1.1| 0.54  | 0.68  | 0.49  | 0.15  | 0.47    |
| a.1.2| 0.18  | 0.23  | 0.40  | 0.35  | 0.29    |
| a.1.3| 0.11  | 0.06  | 0.10  | 0.45  | 0.18    |
| a.1.4| 0.18  | 0.03  | 0.01  | 0.05  | 0.07    |
| Sum  | 1     | 1     | 1     | 1     | 1       |
### Table 5. Pairwise comparison for sub-hierarchy (a.3).

|       | a.3.1 | a.3.2 | a.3.3 | a.3.4 | a.3.5 | a.3.6 |
|-------|-------|-------|-------|-------|-------|-------|
| a.3.1 | 1     | 6     | 5     | 7     | 3     | 9     |
| a.3.2 | 0.167 | 1     | 3     | 4     | 3     | 6     |
| a.3.3 | 0.20  | 0.33  | 1     | 3     | 3     | 2     |
| a.3.4 | 0.14  | 0.25  | 0.33  | 1     | 5     | 7     |
| a.3.5 | 0.33  | 0.33  | 0.2   | 1     | 9     |
| a.3.6 | 0.11  | 0.17  | 0.50  | 0.14  | 0.11  | 1     |
| Sum   | 1.95  | 8.08  | 10.17 | 15.34 | 15.11 | 34    |

### Table 6. Pairwise comparison for hierarchy (a).

|       | a | a.1 | a.2 | a.3 |
|-------|---|-----|-----|-----|
| a.1   | 1 | 5   | 3   |
| a.2   | 0.2 | 1   | 7   |
| a.3   | 0.33 | 0.14 | 1   |
| Sum   | 1.53 | 6.14 | 11  |

|       | a | a.1 | a.2 | a.3 |
|-------|---|-----|-----|-----|
| a.1   | 0.65 | 0.81 | 0.27 | 0.58 |
| a.2   | 0.13 | 0.16 | 0.64 | 0.31 |
| a.3   | 0.22 | 0.02 | 0.09 | 0.11 |
| Sum   | 1   | 1   | 1   | 1   |
4.2. Determined the Hierarchy (a) For Non-Functional Requirement

In this part determined the hierarchy (a) for non-functional requirement for RALIC dataset. According to the Table 4, pairwise comparison for sub-hierarchy (a.1) and Table 5 shows, pairwise comparison for sub-hierarchy (a.3). Table 6, described the pairwise comparison for hierarchy (a).

Table 7. Determined the NFR for hierarchy (a).

| Hierarchy (a) | Weights | Type of NFR |
|---------------|---------|-------------|
| a.1           | 0.58    | Efficiency  |
| a.2           | 0.31    | Security    |
| a.3           | 0.11    | Usability   |

5. Result

The results of requirement prioritization based on non-functional requirement classification using hierarchy analytic hierarchy process (Hierarchy AHP). The result of hierarchy (a) for non-functional requirements is efficiency (60%) which is highest prioritized in hierarchy (a). The outcome weights result and percentage of pairwise comparison using RALIC dataset hierarchy (a) for non-functional requirement list described in Table 7.

Figure 2 described the percentage of NFR for hierarchy (a) dataset and Figure 3 indicated the percentage of hierarchy (a to j) for non-functional requirements. For hierarchy (a), efficiency requirement is the highest percentage of non-functional requirement. And then, for the all hierarchy (a to j) of the highest percentage of non-functional requirement was Portability requirement which reached (92%) in hierarchy (e) RALIC dataset.

Figure 2. Percentage of NFR for hierarchy (a).
6. Conclusions

A Prioritization process can be completed based on several features such as significance, budget, time, risk and instability. Requirement prioritization techniques are important and help to right decisions making in order to develop software products. The study focus on hierarchy analytic hierarchy process pairwise comparison matrix calculation using existing RALIC dataset requirement and classified the hierarchy (a) for non-functional requirement. The outcome of hierarchy (a) consistency ratio accepted for accuracy. As well as, the percentage of efficiency requirement is the highest percentage for non-functional requirement in hierarchy (a). When making the pairwise comparison for hierarchy (a to j) requirements input is (15, 051), in contrast, hierarchically pairwise comparison for (a to j) requirements input is (403). Therefore, hierarchically comparison can reduce the number of requirements (97.33 %) by using hierarchy AHP. Secondly, AHP technique had weakness when validate consistency ratio check for the accuracy calculation accepted for 10 number of requirements only because Random Index (RI) has value for 10 number of requirements only. Consistency Ratio = Consistency Index / Random Index (CR = CI / RI) in that formula when calculate the accuracy formula able to accept for 10 requirements and then not allow to more than 10 number of requirements. Therefore, this paper segregated and prioritized the non-functional requirements using RALIC dataset hierarchy (a) requirements.
Figure 3. Percentage of NFR for hierarchy (a to j).
Acknowledgment

The work presented in this paper is supported by Universiti Malaysia Pahang Grant: RDU182207-2.

References

[1] Sommerville, I., Introduction to software Engineering. Addison-Wesley, 2007.
[2] Vestola, M., A Comparison of Nine Basic Techniques for Requirements Prioritization. Helsinki University of Technology, 2010.
[3] Lehtola, L., Providing value by prioritizing requirements throughout software product development. Licentiate thesis, Helsinki University of Technology, 2006.
[4] Lim, S.L. and A. Finkelstein, StakeRare: using social networks and collaborative filtering for large-scale requirements elicitation. IEEE Transactions on Software Engineering, 2012. 38(3): p. 707-735.
[5] Saaty, T., The Analytic Hierarchy Process, McGraw-Hill, New York, 1980. There is no corresponding record for this reference.
[6] Karlsson, J. and K. Ryan, A cost-value approach for prioritizing requirements. IEEE software, 1997. 14(5): p. 67-74.
[7] Berander, P. and A. Andrews, Requirements prioritization, in Engineering and managing software requirements. 2005, Springer. p. 69-94.
[8] StepH Fam. A List of Requirements Prioritization Techniques You Should Know About. 2016 August 18, 2016 [cited 2017 2/1/2017].
[9] Berander, P. and P. Jönsson, Hierarchical cumulative voting (hcv)—prioritization of requirements in hierarchies. International Journal of Software Engineering and Knowledge Engineering, 2006. 16(06): p. 819-849.
[10] Voola, P. and A.V. Babu, Comparison of requirements prioritization techniques employing different scales of measurement. ACM SIGSOFT Software Engineering Notes, 2013. 38(4): p. 1-10.
[11] Ribeiro, R.A., et al., Hybrid assessment method for software engineering decisions. Decision Support Systems, 2011. 51(1): p. 208-219.
[12] Sommerville, I., Software engineering. 2010: Pearson.
[13] Karlsson, J., C. Wohlin, and B. Regnell, An evaluation of methods for prioritizing software requirements. Information and Software Technology, 1998. 39(14): p. 939-947.
[14] Khan, K.A., A systematic review of software requirements prioritization. 2006, Blekinge Institute of Technology.
[15] Berander, P., Evolving prioritization for software product management. 2007.
[16] Leffingwell, D. and D. Widrig, Managing software requirements. A use case aproach. 2003, Addison-Wesley.
[17] Brackett, J.W., Software requirements. 1990, DTIC Document.
[18] Mustafa, B.A. and A. Zainuddin. An experimental design to compare software requirements prioritization techniques. in Computational Science and Technology (ICCST), 2014 International Conference on. 2014. IEEE.
[19] Lim, S.L., Social Networks and Collaborative Filtering in Large-Scale Requirements Elicitation. 2010
[20] Khan, J.A, et al., An Evaluation of Requirement Prioritization Techniques with ANP. International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, pp. 419-429, 2016
[21] J. Karlsson, C. Wohlin and B. Regnell. “An Evaluation of Methods for Prioritizing Software Requirements”, Information and Software Technology, Vol. 39, No. 14-15, pp. 939-947, 1997-98