Non-functional Requirement Prioritization Approach

Varun Gupta[*], Shivam Lohia, Deniz Çetinkaya and Hye-jin Kim

Assistant Professor-III, ASET/CSE, Amity University, Noida.
Student, B. Tech, ASET/CSE, Amity University, Noida.
Assistant Professor, Department of Software Engineering, ATILIM University, Turkey.
Business Administration Research Institute,
Sungshin W. University, Seoul, Korea
Vgupta7@amity.edu , lohiashivam@yahoo.com, deniz.cetinkaya@atilim.edu.tr, hyejinaa@daum.net

Abstract

This paper proposes a non-functional requirement prioritization technique where both functional and non-functional requirements are prioritized and that is based on AHP (Analytic Hierarchy Process) technique employing suitable aspects.

The non-functional requirements are further re-prioritized based on their dependency with high priority functional requirements and usage counts. The usage of the proposed method is illustrated with a hypothetical example. The technique is also executed on a case study of the evolving software and the technique resulted in selection of 16 requirements out of 30, yielding considerable reduction in the number of requirements to be implemented. Although the use of the AHP technique may limit the optimization, the use of the numerical assignment technique based on the usage count and dependency count do considerably reduce the number of requirements.

The proposed technique prioritizes non-functional requirements independently from functional requirements but they are further selected in terms of their dependencies and usage counts with respect to functional requirements.

***Keywords**: Non-functional requirements, functional requirements, prioritization*

1. Introduction

Non-functional requirements together with functional requirements comprise the complete software system. Functional requirements are demanded by the customer and intended users since they satisfy the needs or expectations of the users. Users generally not directly ask for non-functional requirements but these requirements determine the level of acceptability of the software system. For example, a user may need many functions to be provided by the ATM software, but the level to which it can be used depends not only on functional requirements but also on non-functional ones like security, reliability, usability, performance, *etc*. In any incremental software development process, the development team has to perform decision making about the requirements to be implemented under resource and time constraints. Number of requirements is numerous and increases significantly with every increment, which makes it difficult to select the set of functional and non-functional requirements.

There exist many techniques for prioritization of non-functional requirements like those proposed in (Dabbagh and Lee, 2013; Dabbagh *et. al.*, 2014; Thakurta, 2013). These techniques may be based on various approaches like separate prioritization of functional requirements from non-functional ones, prioritization together or hybrid

[*] Corresponding Author

approach as proposed in (Chopra *et. al.*, 2016). It had been reported in literature that the prioritization of non-functional requirements is mostly ad-hoc and non-functional requirements are mostly ignored or neglected (Mairiza *et. al.*, 2010a, b; Thakurta, 2013). However, it had been reported that the non-functional requirements are crucial for the product to be successful (Svensson *et. al.*, 2011).

This paper presents a non-functional requirement prioritization approach which is based on findings of (Chopra *et. al.*, 2016) that non-functional requirements must be prioritized separately but in accordance with the functionality of the proposed system. The paper is structured as follows: Section 2 explains the proposed algorithm and Section 3 illustrates the use of the proposed technique with a hypothetical example. Section 4 executes the technique on a case study of the tool "automatic analysis and comparison of different release planning methods" and highlights the optimization achieved. Finally, the paper is concluded and directions for future work is given in Section 5.

2. Proposed Requirement Prioritization Approach

The proposed algorithm for the prioritization of non-functional requirements does the prioritization of functional and non-functional requirements separately using AHP (Analytic Hierarchy Process) technique considering the aspects like business values, cost, time or any other as deemed necessary by the organization. The non-functional requirements are further re-prioritized and selected in accordance with their impact on the highest priority functional requirements, dependencies with them and historical usage count. The idea is to first prioritize the requirements separately, then to consider all non-functional requirements with only highest priority value (threshold may vary from company to company) on the basis of their dependencies with these functional requirements and usage count as revealed by the historical values of same or similar projects. So, the non-functional requirements that are ranked high by the developer and have dependency on these functional requirements and are used maximum times, are considered to be of highest priority.

Usage count of non-functional requirements is to be managed through establishing relationships with the usage count of functional requirements. The algorithm is given as follows:

```
ALGORITHM: NFR_Prioritization (F, NFR, USAGE, DEP).

Where F is set of functional requirement with cardinality N,

NFR is the set of non-functional requirements with cardinality M,

USAGE is the set that stores the usage count for every element of the set NFR,

DEP is the N * M matrix that stores the dependency count between one element of set
F and another element of set NFR. The value of DEP lies in the range of 0 to 9 where
0 means no dependency and 9 means highest dependency (subjective judgment).

Step 1. Prioritize both F and NFR separately using AHP technique.

Step 2. Select the functional requirement from the set F that has threshold
        value of priority above the organization dependent value of Q. Call
        this set as $F^1$.
```

Step 3. Populate the DEP array by putting the elements of set F^1 as **columns** of the array and all NFR as **the rows**. The array is to be filled by the entries lying between 0 to 9. This is subjective judgment of the developer.

Step 4. Populate the USAGE set.

USAGE(i) = U(F(j)) + U(F(k)) + …….. + U(F(l))

Here U is the function that returns the usage count of a functional requirement F(i). Here a NFR say i is dependent on functional requirements j, k and l.

Step 5. Prepare three buckets namely High priority, Medium priority and Low priority. Put the NFR in particular bucket according to the value of established priority and maximum value of DEP array. In other words, first put NFR in particular bucket according to the priority value. Then reshuffle those of medium and low priority bucket to higher level (low to medium and medium to high) with highest value of dependency with functional requirements belonging to the set F^1.

Step 6. Further shuffle the requirement from low to medium and medium to high according to their usage count values. The bucket with High priority is to be implemented during current release.

The algorithm employs the numerical assignment technique in step 5 and 6. The ever increasing requirements may make the proposal bit less scalable due to the use of AHP technique but large number of requirements can be handled as numerical assignment technique is also employed.

3. Hypothetical Example

Let there are 4 functional (A, B, C and D) and 4 non-functional requirements (E, F, G and H) and are already prioritized using AHP resulting in the ordering B, D, A, C and H, G, F, E. The usage count is given as E(5), F(2), G(1) and H(1). The dependency count of each NFR with other functional requirements is given in Table 1.

Table 1. Dependency Count of Each NFR on all Functional Requirements

	A	**B**	**C**	**D**
E	2	1	2	2
F	4	1	2	8
G	5	2	1	1
H	2	2	2	4

The H and G are in High priority bucket, F is in medium and E is in low priority bucket. Let’s suppose that B and D are above threshold and considered as high priority. On the basis of Usage count, E will get shifted to medium priority bucket. On the basis of usage count, only the requirement E will get shifted to High priority bucket. Thus three NFR's will be implemented namely H, G and E.

4. Proposed Method Evaluation

The proposed technique is evaluated on a software project of tool for automatic analysis and comparison of different release planning methods, developed in the Computer Programming laboratory of Jawaharlal Nehru Govt. Engineering College, Sundernagar, India.

The reason for using this tool was that this project was implemented in the year 2012 and the authors had the evolution details of the software. This software evolved six times in form of six increments. The total functional requirements of sixth increment were 24 and non-functional were 6.

The proposed technique was executed in the following steps:

1. The 24 functional requirements and 6 non-functional requirements were independently compared using pair wise comparison using AHP technique. The total number of pair wise comparisons were 276 for functional and 15 for non-functional requirements. Let's denote non-functional requirements by A, B, C, D, E and F.
2. The AHP established the priority of all requirements and as per the analysis of historical changes in requirement priorities, it had been observed to keep the threshold value at the 50% of the number of requirements on the basis of its priority.
3. Total 13 functional requirements were finally selected for later stage processing.
4. All six prioritized non-functional requirements were placed in either of the three buckets i.e. High priority, Medium priority and Low priority. Requirement B and F were added to High bucket, A in medium bucket and C, D and E were added to Low priority bucket.
5. The requirements were added in six increments and the usage count was maintained for all requirements. To maintain it, the counter is associated with each requirement and as the software is getting executed in the laboratory, the usage count value changes.
6. The six non-functional requirements were organized as rows and 13 functional requirements as columns of the dependency matrix. The dependency values were filled. The B and F remained in High bucket, A remained in medium bucket and C moved to medium bucket with others remaining in Low bucket.
7. Based on above dependency values, the usage count of all non-functional requirements was calculated. The requirement A moved to High bucket with others remaining in old buckets.
8. High bucket thus has requirement A, B and F, medium has C and Low has D and E.
9. Software team must select all 13 functional requirements and 3 non-functional requirements from High bucket and may go ahead with medium bucket if resources allows.
10. This way, there were total 16 requirements out of 30 requirements.

.

5. Conclusion & Future Work

The proposed technique prioritizes NFR separately from functional requirements but considers their dependencies with functional requirements and the usage count. The use of AHP will make the proposal susceptible to scalability problem but not to that amount due to use of numerical assignment technique. Further, subjective judgments may prove to be threat to validity. The validation of the proposed work on large data set is planned as the future work.

References

[1] Dabbagh M. and Lee S., "A Consistent Approach for Prioritizing System Quality Attributes", 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, **(2013)**, pp. 317-322.

[2] Dabbagh M., Lee S. and Parizi R., "Application of Hybrid Assessment Method for Priority Assessment of Functional and Non-Functional Requirements", International Conference on Information Science and Applications (ICISA), **(2014)**.

[3] D. Mairiza, D. Zowghi and N. Nurmuliani, (2010a), "An investigation into the notion of non-functional requirements", ACM symposium on applied computing—SAC, pp. 311–317.

[4] D. Mairiza, D. Zowghi and N. Nurmuliani, (2010b), "An investigation into the notion of non-functional requirements", In Proceeding of the 25th ACM symposium on applied computing. Sierre, Switzerland.

[5] Thakurta R. (2013), "A framework for prioritization of quality requirements for inclusion in a software project", Software Quality journal, vol. 2, pp. 573–597.

[6] Svensson R., Gorschek T., Regnell B., Torkar R., Shahrokni A. and Feldt R. (2011), "Prioritization of Quality Requirements: State of Practice in Eleven Companies", IEEE 19th International Requirements Engineering Conference, pp. 69-78.

[7] Chopra R., Gupta V., and Chauhan DS (2016), "Experimentation on accuracy of non-functional requirement prioritization approaches for different complexity projects", Perspectives in Science, vol. 8, pp. 79-82

Authors

Varun Gupta, has done PhD, M. Tech (Research) and B. Tech (Hon's) in CSE. He is working as assistant Professor-III in amity school of Engineering & Technology (ASET), Amity University, Noida. He is working in the area of Software Engineering and had worked with National Institute of Technology-Hamirpur, PEC University of Technology-Chandigarh and Indian Institute of Technology-Mandi. The author has authored numerous research papers in peer reviewed, referred International Journals and various International and National Conferences that were held in India and Abroad.

Shivam Lohia, is doing Btech in CSE from ASET, Amity University, Noida, U.P.. He had been doing his minor and major research project under the guidance of Dr. Varun Gupta & Dr. Deniz Cetinkaya. He is interested in software engineering especially requirement prioritisation. He has technical expertise in C, C++, java and SPSS. He has implemented various technical projects and has various scholastic achievements to his credit.

Deniz Çetinkaya, received PhD from Delft University of Technology (The Netherlands), her Master from Middle East Technical University and Bachelor degree from Hacettepe University. She is working as an Assistant Professor in the Department of Software Engineering, ATILIM University, Turkey. She is working in Software Engineering, Systems Engineering, Simulation, Conceptual modeling and Model driven development. She had numerous papers in leading Journals and Conferences.