

A Survey on Prioritization Methodologies to Prioritize Non-Functional Requirements

Saranya. B.

Department of Computer Science and Engineering Sri Krishna College of Technology, Coimbatore, Tamilnadu, India

Subha. R.

Department of Computer Science and Engineering Sri Krishna College of Technology, Coimbatore, Tamilnadu, India

Dr. Palaniswami. S.

Principal, Government College of Engineering, Bodinayakanur, India

ABSTRACT

Nonfunctional Requirements are as important as functional requirements. But they have been often neglected, poorly understood and not considered adequately in software development process. If the NFRs are not met properly, it will lead to the dissatisfaction of customers. NFRs may be more critical than functional requirements as there can be mutual dependencies among the NFR, which may affect the completion of the project. Hence it is necessary to prioritize the NFRs effectively. But prioritizing such NFR is a challenging task in Software development. Many techniques are used to prioritize the requirements in various dimensions. It is important to choose the appropriate requirement prioritization technique for a particular software development process. One can select the appropriate techniques based on the various factors such as, the stakeholders involved, available resources, and the product he develop and so on. The goal of this paper is to increase the awareness about the importance of NFRs and to analyze the various techniques that are used to prioritize the NFRs.

Keywords

Requirements Engineering, Non Functional Requirements, Prioritization of NFRs, Prioritization techniques, Quality requirements, NFR algorithm

1. INTRODUCTION

Requirements Engineering (RE) is the subfield of Software engineering; it involves formulating, documenting and maintaining of software requirements [16]. Requirements are generally described as what the system is required to do along with the environment, it is intended to operate in. Requirements provide the description of the system, its behavior, application domain information, system constraints, specifications and attributes [7].



Requirements may be Functional or Nonfunctional.

- Functional requirements (FRs) describe system services or function.
- Nonfunctional requirements (NFRs) are a constraint on the system or on the development process.

The purpose of "identifying non-functional requirements" is to get a handle on these absolutely necessary requirements that are normally not functional. Some of these NFRs may conflict with each other. As like of the functional requirements the NFRs a will also vary during the development process. So it is essential to keep track to the NFRs throughout the development process. NFRs are of three types Product requirements, Organisational requirements, and External requirements [12].

- **Product requirements:** Requirements which specify that the delivered product must behave in a particular way, e.g. execution speed, reliability, etc.
- **Organisational requirements:** Requirements which are a consequence of organizational policies and procedures, e.g. process standards used implementation requirements, etc.
- **External requirements:** Requirements which arise from factors which are external to the system and its development process, e.g. interoperability requirements, legislative requirements, etc.

The software development market can be divided into two major types, namely market-driven development and bespoken development. In marketdriven development the product is developed for an open market, whereas in bespoken market the product will be developed for the particular customer based on their wishes. In the bespoken development if there is only one customer there will not be any problem. But in real time, many customers and developers will be involved in the software development and everyone has different views and opinions. In such situation requirement prioritization plays a major role in software development.

Requirement Prioritization is done in order to determine which NFRs to be implemented in a software product in the current release or in order to understand the mandatory NFRs that should be implemented in the product to obtain the satisfaction of the customers. During a project development, decision makers in software development need to make many different decisions regarding the release plan. Requirement prioritization plays a major role in decision making. It helps the decision makers to decide when to release a product, how to develop the project in time, how to reach milestones, etc. It helps them to resolve issues and risks arise during the



software development. The main reason for prioritizing is that, all NFRs cannot be implemented by the given time or with the given resources. In this paper, we analyze the prioritization techniques. The paper is structured as follows: After the introduction Sect. 2 provides a review of related work, Sect 3 explains about requirement prioritization, Sect. 4 & Sect. 5 describes the various techniques to prioritize the NFRs; Sect. 6 concludes the paper.

2. RELATED WORK

Quality requirements are defined during the early stages of development, but they are not incorporated properly during software development. The NFRs are prioritized for various reasons such as to determine whether the particular requirement is mandatory or not, to eliminate the requirements that are unnecessary to the software product and to schedule the requirements for implementation [5]. When the requirements are properly prioritized it provides the significant benefits such as improved customer satisfaction, Lower risk of cancellation and it also helps to identify the hidden requirements. It helps to estimate the benefits of the project and also priorities of requirements can help to determine how to utilize the limited project resources. The various factors involved in prioritizing requirements are cost, risk, value, benefits, dependency constraints, effort business value dimensions etc. [1].

In web development NFRs are not given importance as like the functional requirements, they are not discussed properly in the early stages. But web developers found that NFRs become an issue during the later stages of development. As a result, it leads to frequent change in the system design. So paying more attention to the NFRs increases the quality of the web system and also reduces the development time. In many techniques while prioritizing NFRs, mostly they are converted into Functional requirements. For example, the security requirement is For example, the security requirements for cost estimation [6]. So when NFRs are elicited properly, it also leads to the discovery of new Functional requirements (FRs) [11].

Many challenges arise while prioritizing requirements. For example, some stakeholders believe that all the requirements have high priority, even though they accept theoretically that different requirements have different priority they always try to push for having most requirements under high priority. So we should make them to understand that the requirement is not necessary for time being [5]. So when NFRs are elicited properly, it also leads to the discovery of new Functional requirements (FRs) [11].



Many challenges arise while prioritizing requirements. For example, some stakeholders believe that all the requirements have high priority, even though they accept theoretically that different requirements have different priority they always try to push for having most requirements under high priority. So we should make them to understand that the requirement is not necessary for time being [5]. The main difficulties in prioritizing quality requirements are: First it is difficult to determine the cost and effort required to implement NFRs in advance. Second for quality requirement values are continuous, so it is tough to determine the right required value [3].

3. REQUIREMENT PRIORITIZATION

Requirement prioritization is an important activity in software development; it helps to manage the relative importance of requirements and to manage the software development when the resources are limited.[13]. All the Stakeholders should collaborate to prioritize the requirements efficiently. Requirement prioritization supports the following activities:

- To estimate expected customer satisfaction.
- To decide the core requirements.
- To schedule the implementation of requirements.
- To handle the dependency between the requirement.
- To establish the relative importance of each requirement.

Requirements can be prioritized under various dimensions. They are as follows [1]:

Cost: It denotes the cost required to implement the requirement successfully. Often the developers will prioritize the requirements in the terms of money.

Value: Value is the importance of the each requirement when compared to other requirements. The requirements with high importance are implemented first. As every stakeholder has various views on a requirement, it is difficult to determine the value of a requirement.

Risk: It denotes the risk arises while implementing a Requirement. The requirements with the high risk should be implemented first else it would extend the time of implementation.

Dependency Constraints: It denotes how two requirements are dependent on one another.

- **Precedence Constraints:** This condition occurs when implementation of one requirement requires the completion of any other requirements.
- **Coupling Constraints:** It represents the requirements that can be implemented in parallel.

Business Values: It represents the mission of an organization. Example of Business values is Continuous Improvement, customer satisfaction etc.



Effort: It is the effort required to implement the requirement during the development process.

Resources: Resources refer to the budget, staff and schedule. It is the crucial factor that is not considered in most prioritizing techniques.

Approach: Requirements prioritization involves two approaches.

- Absolute Assessment: Every requirement is assessed against each criterion.
- **Relative Assessment:** With this approach, each every requirement is compared with every other requirement.

As the requirements are analyzed in various perspectives, the requirement prioritization also provides many other benefits. For example, it helps to find the hidden requirements, defects in requirements. By implementing the high priority requirements before lower priority requirements it is possible to reduce the time and cost required for that project. The require prioritization can be divided into sub process [5]:

Convince stakeholders: Convince the stakeholders and make them to understand the requirements.

Train stakeholders: Train the stakeholders to participate in the prioritization process,

Categorize raw potential requirements: Then the raw potential requirements should be categorized into actual requirements, useful capabilities and desirable capabilities.

Prioritize the actual requirements: Prioritize the identified requirements it involves negotiating with stakeholders and validating the results. To determine the priorities of requirements the prioritization process should be done in incremental basis. For the effective prioritization, representatives from all stakeholder groups should be involved in requirement team and it should be led by a professional requirements engineer.

Publish the priorities: After prioritization the requirement team should publish the results and let stakeholders know about it.

Estimate effort: After finding the priorities calculate the effort required to implement the requirements.

Schedule development: Then schedule the development process according to the priorities calculated. As requirements are allocated in incremental basisthe requirements team, development team and the management team should work together to allocate the requirements properly.

Maintain priorities. During the development process the requirements and priorities may vary. So store the priorities in the repository and maintain it when changes occur.Simply the requirement prioritization can be carried down in three stages [14]. The preparation stage, in this stage the team and the team leader will prepare the requirements according to the principles of the prioritization techniques to be used. The second stage is an execution



stage where the decision makers actually prioritize the requirements. The final stage is a presentation stage where the results of the prioritization are presented. But while prioritizing the requirements numerous ricks and challenges arises [5].

The major challenges in prioritization are:

- When requirements are large it is difficult to prioritize it efficiently.
- As different stakeholders have different views, finding the mandatory requirement is difficult.
- When resources are limited then prioritization is very challenging.
- As the requirements may vary in later stages, it also leads to changes in priorities.
- Some requirements may incompatible with another. So implementing one requirement denies the implementation of other requirements.
- The customer may misunderstand the prioritization process.
- The customer may try to pull all requirements in mandatory group.

4. PRIORITIZATION METHODOLOGIES

4.1 Numerical Assignment

Numerical assignment involves the process of grouping the requirements. At first the requirements are divided into different groups, then they are given to the stakeholders to enter the scale value from 1-5 for each requirement based on the importance. Finally, the average value given by all stakeholders are considered as the ranking for that requirement [9]. The main disadvantage of this method is [1], as different users have different opinions the information obtained is relative and it is difficult to determine the absolute information. If the stakeholders prioritize the requirements, they will pull 85% of the requirements into the high priority group.

4.2 Analytical Hierarchy Process (AHP)

AHP [1] is decision making technique which involves pairwise comparison. With the help of multiple objectives or criteria AHP allows decision makers to choose the best requirements from the several decision alternatives. AHP involves three steps. They are

- Making pairwise comparison.
- Calculating the priorities of requirements and decision making
- Checking consistency.

In this is technique each requirement is compared with every other requirement to determine to what extent one requirement is more important



than other requirements. So for n requirements n. (N - 1)/2 pairwise comparisons are required [9].

The fundamental scale used in AHP is given below [1]:

- 1 Of equal importance
- 2 Moderate difference in importance.
- 5 Essential differences in importance.
- 7 Major differences in importance.
- 9 Extreme differences in importance.

Reciprocals-If requirement I have one of the above numbers assigned to it when compared with requirement j, then j has the reciprocal value when compared with I.

The redundant pair-wise comparison makes this methodology very trustworthy during decision making. But when the number of requirements becomes large, it requires a large number of comparisons so the system will become complex.

4.3 Value-Oriented Prioritization (VOP)

It involves two steps, first establishing the Framework. The first step is identifying the business core values, and then the organizational executives must provide the importance of those values to the organization. To assign weights simple scale ranging from 0 (Not important) to 10 (Critical) is used. VOP also supports weighting using business risk categories. Then Applying the Framework that is by using the identified core values and risk weight category's requirements are prioritized by constructing prioritization matrix. VOP is the only technique which considers the business core values. In VOP decision making is transparent to all the stakeholders involved, so it avoids lengthy arguments about a particular requirement [8].

4.4 Cumulative Voting

Cumulative voting (CV) or 100 dollars method is a prioritization technique where each stakeholder will be given 100 imaginary units. They can use these units to vote for the requirement that they consider as important. One can use all this 100 units for a single requirement or it can be equally distributed to all requirements. In this way, units assigned to a requirement represent the respondent's relative preference to a requirement in relation to the other requirement. Because of this cumulative voting is also known as "proportional voting". The term "proportional" also reflects that if the amount of units assigned to a requirement is divided by the constant number of units available to each stakeholder, the result will be a proportion between zero and one.

In CV only one chance will be given to a stakeholder because if more chance is given they may use all their units on their favorite requirements to make it as the highest priority one [15]. Some stakeholders will pull all their



units into a requirement that is not highly preferred by any others. So it will lead to conflicts in the prioritization process.

4.5 Binary Search Tree

Binary search tree (BST) uses basic BST data structure concept. BST algorithm stores the information and it can be retrieved when it is necessary. The BST is the special case of binary tree in which each node has at most two children. The child nodes to the right have greater value/importance than the root node, and the child nodes to the left have less value/importance than the root node. If the BST is traversed in order, the requirements will be in sorted order [14]. For n requirements binary tree will consist of n requirements and it requires n log n comparisons until all requirements are inserted.

The three stages involve in the BST are as the preparation step requirements are gathered in execution step the requirements are inserted one by one. First insert the one requirement in the first node, while inserting the priority then it is added as the right child. Finally, at the presentation stage traverse the tree in order and prioritize the requirements.

5. PRIORITIZATION METHODOLOGIES PARTICULARLY DEFINED FOR NFRS

With the above mentioned methodologies there are also many other methodologies such as Top ten requirements, Ranking, Planning Game were used to prioritize NFRs [8]. But all of these techniques were designed for FRs and can be used for NFRs. So it does not give efficient results when we use to prioritize NFRs. There are some methodologies that are particularly defined to prioritize the NFRs. They are discussed below.

5.1 Prioritization Using Architecture Feedback

This architectural design starts with the initial Requirements Engineering activities as the initial understanding of NFRs is required. The steps are shown in Fig. 1

The steps involved in architectural feedback are:

- Identify the NFRs relevant to the software development.
- Specify the NFRs.
- Prioritize the NFRs based on the views of the stakeholders involved.
- Based on the initially prioritized NFRs design the initial architecture and create the architecture model with quality annotations required for evaluation.
- Evaluate software architecture and explore the design space using the design space exploration tool such as Per- Opteryx.





Figure 1. Prioritization using architectural feedback

- Based on the design space explorations Analyze trade-offs (the above 3 steps are done by software architects).
- Based on the results obtained, stakeholders discuss and negotiate on the required quality levels and reprioritize the NFRs.
- The software architect updates the architecture accordingly.
- The architectural design is used to implement the system and it should be updated continuously.
- If any changes in NFRs then above steps should be repeated.

This process helps the stakeholders and architects to understand the NFRs effectively. But this method is applicable only to the quantitatively evaluated quality properties.

5.2 Prioritizing By Goal-Decomposition

This approach uses goal decomposition method to prioritize NFRs. In this method nonoperation specifications are translated in to operation definitions. After the system is developed the operation definitions are evaluated [4]. Initially requirements are prioritized while eliciting. Here the stakeholders will group the requirements which they think as critical and they are implemented first. But developers do not have any influence in this step. The second step is executed during the requirement negotiation meeting. In this step the requirement conflicts are analyzed in the presence of all stakeholders. To support this step link pairs of requirements is used with the right sign that indicates their positive (+), negative (-) or neutral ("") interaction.

The rules for assigning the link pairs are

• The value "-" is assigned to a pair of NFRs when one NFR in the pair has a negative effect on the other at the same functionality.



- The value "+" is assigned to a pair of NFRs where one NFR in the pair has a positive effect on the other.
- The value "" is assigned to a pair of NFRs in the set of NFRs that do not interact.
- The value "?" is assigned to a pair of NFRs when the influence is unknown.
- The conflicts in among the NFRs can be resolved using the scale values, for example very important, important, average and don't cares can be used.
- Then identify the stakeholder goals that are important to the success of the system. In this the hard goals will be given higher priority than the soft goals. Apply Goal Question Metrics to the NFRs and prioritize them.

5.3 NFR Algorithm

The NFR algorithm is based on the mutual dependencies among the NFRs and the preferences of the stakeholders. The priority of NFRs is determined by using the respondents of the project and the business organization are involved [9].It uses a simple heuristic to prioritize the NFRs. The steps involved in NFR algorithm are followed:

- Business organization representatives identify the NFRs that are concerned with the particular project.
- Based upon the NFRs Business process Hierarchy is created. This scenario defines the final product that should be delivered to the customer.
- The identified NFRs are given to stakeholders to provide ratings between 0-9 based various Business Value Dimensions (value, penalty, cost, risk).



Figure 2. NFR Algorithm



- The ratings are then combined to calculate the relative importance of the NFRs
- The relative priority is calculated using the formula

$$Relative importance_{i} = \frac{(Value + Penalty)\%}{(Cost\% + Risk\%)}$$

(1)

Where

$$(Value + Penalty) \% = \frac{Value_{i} + Penalty_{i}}{\sum_{N} Value_{i} + \sum_{N} Penalty_{i}} * 100$$
(2)
$$Cost \% = \frac{Cost_{i}}{\sum_{N} Cost_{i}} * 100$$
(3)
$$Risk \% = \frac{Risk_{i}}{\sum_{N} Risk_{i}} * 100$$
(4)

 $i = i_{th}$ NFR considered for implementation.

• Identify the association among the NFRs. The association between the NFRs may be positive, negative or neutral. The association can be expressed clearly in the form of association matrix.

	X1	X2	X3	X4
X1	+m2			+m6
X2		+m3		
X3			-m4	-m7
X4	-m1			
Net % change	NC1	NC2	NC3	NC4

• So construct NFR association matrix to the scenario.



- Indicators mi's in the table shows that if the sign is + then there is a positive association between the NFRs and if the sign is then there is a negative association between the NFRs.
- 'Net % Change' row indicates the aggregate percentage improvement or degradation of capability identified by NFRs.
- Adjusted importance is calculated using the formula:

Adjusted Importance $_{i}$ = Relative Importance $_{i}$ * (1+ Net % Change $_{i}$ /100) (5) Where $i = i_{th}$ NFR considered for implementation.

- The NFRs are dropped by using the following heuristics.
- Calculate the mean of the Adjusted Importance value of all the NFRs.
- Drop the NFRs with the negative adjusted importance and whose value is less than the mean value of adjusted importance.

6. CONCLUSION

In the competitive Business environment, the quality of a product plays a very crucial role in its success. So we have to prioritize the NFRs efficiently. In the above techniques we found that the NFR algorithm is a most suitable methodology to prioritize the NFRs. It is because the algorithm is particularly designed for NFR prioritization. As the business process hierarchy is created it can able to identify all the NFRs easily, it prioritizes the NFRs in various dimensions and from various stakeholder views. The heuristics involved is very simple to calculate and it also considers the mutual dependencies among the NFR. So NFR algorithm can prioritize the NFRs efficiently and in the cost effective manner.

REFERENCES

[1]. Aaqib Iqbal, Farhan, M. Khan, Shahbaz, A. Khan, 2009. A Critical Analysis of Techniques for Requirement Prioritization and Open Research Issues. International Journal of Reviews in Computing. Vol. 1.

[2]. Anne Koziolek, 2012. Research Preview: Prioritizing Quality Requirements Based On Software Architecture Evaluation Feedback. REFSQ 2012. pp. 52-58.

[3]. Berntsson Svensson, R., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., Feldt, R., and Aurum, A. 2011. Prioritization of quality requirements state of practice in eleven companies. RE'11 IEEE. pp. 69.

[4]. Daneva1, M., Kassab, M., Ponisio, M.L., Wieringa, R. J., Ormandjieva, O., 2007. Exploiting A Goal-Decomposition Technique to Prioritize Non-Functional Requirements. Proceedings of the 10th Workshop on Requirements Engineering WER.

[5]. Donald Firesmith, 2004 "Prioritizing Requirements", Journal of Object Technology, Vol 3, No 8.



[6]. Herrmann, A., Daneva, M., 2008. Requirements Prioritization Based On Benefit And Cost Prediction: An Agenda For Future Research. In Proceedings of The 16th IEEE International Requirements Engineering Conference. pp. 125–134.

[7]. Kotonya, G., Sommerville, I., 1998. Requirements engineering: Processes and techniques", Chichester. UK: John Wiley & Sons.

[8]. ManjuKhari ,Nikunj Kumar, 2013. Comparison of six prioritization techniques for software requirements. Journal of Global Research in Computer Science, Vol. 4, No. 1.

[9]. Muhammad Ramzan, ArfanJaffar, M., Arshad Ali Shahid, 2011. Value Based Intelligent Requirement Prioritization (Virp): Expert Driven Fuzzy Logic Based Prioritization Technique. International Journal Of Innovative Computing, Information And Control.Vol. 7, No. 3.

[10]. Rahul Thakurta, 2013. A Framework for Prioritization of Quality Requirements for Inclusion In A Software Project. Springer Software Quality Journal. Vol. 21, No 4, pp. 573-597.

[11]. Yusop, N., Zowghi, D., Lowe, D., 2008. The Impacts of Non-Functional Requirements in Web System Projects. International Journal of Value Chain Management, Vol. 12, pp. 18–32.

[12]. www.inf.ed.ac.uk/teaching/courses/cs2/LectureNotes /CS2Ah/SoftEng/se02.pdf.

[13].http://www.corpedgroup.com/resources/ba/ReqsPrioritization.asp

[14]. Joachim Karlssona, Claes Wohlinb, Bjorn Regnell, 1998. An evaluation of methods for prioritizing software requirements. Information and Software Technology. Vol. 39, pp. 939–947.

[15]. Patrik Berander Anneliese Andrews, 2005. "Requirements Prioritization", Engineering and Managing Software Requirements, pp. 69-94.

[16]. http://en.wikipedia.org/wiki/Requirement_engineering.

This paper may be cited as:

Saranya. B., Subha. R. and Palaniswami. S., 2014. A Survey on Prioritization Methodologies to Prioritize Non-Functional Requirements. *International Journal of Computer Science and Business Informatics, Vol.* 12, No. 1, pp. 32-44.