Comparing AHP and ELECTRE I for prioritizing software requirements

João M. Fernandes* Susana Prozil Rodrigues* Lino A. Costa†
orcid.org/0000-0003-1174-1966 orcid.org/0000-0003-4772-4404
*Dep. Informática / Centro ALGORITMI
†Dep. Produção e Sistemas / Centro ALGORITMI
Universidade do Minho, Braga, Portugal

Abstract

Requirement prioritization is a process that allows selection of the "key" candidate requirements, the ones that are the most important for the construction of quality and cost-controlled software. Requirement prioritization brings certain issues and challenges related with the different stakeholders involved in the project, as well as with the prioritization techniques used, which differ in procedures, criteria and metrics. This manuscript compares two multi-criteria decision methods (MCDM), AHP and ELECTRE I, seeking to justify which one is the most feasible in the requirement prioritization process of a real-world case study. To accomplish this aim, several criteria were used to compare the applicability and performance of both MCDMs. In order to reflect reality as close as possible, several stakeholders, including software professionals directly related to the case study, were involved. The results confirm the intuition that ELECTRE I is more easily applicable than AHP. ELECTRE I is subject to fewer mistakes in comparisons of the requirements than the AHP method, as these are carried out differently. In fact, due to its inherent complexity, AHP becomes even impractical in software projects with a large number of requirements.

1. Introduction

The decision of which features to include in a software product is in many cases quite difficult to take, due to several factors. This fact can be confirmed in the following words of Frederik Brooks [3]:

> *The hardest single part of building a software system is deciding precisely what to build. [...] No other part of work so cripples the resulting system if done wrong. The other part is more difficult to rectify later.*

Given a set of candidate requirements, that at the outset are considered as relevant, it is necessary to be able to select the subset with the most important requirements to develop a quality software and simultaneously to reduce costs [10]. The prioritization of requirements helps in identifying these "key" requirements and can then be seen as a process that ranks the set of requirements [2]. The requirements prioritization process is not stagnant and fixed in time and it exposes certain problems and challenges related either with the different stakeholders involved in the project or with the prioritization techniques themselves (that differ in criteria and metrics). Thus, a big issue when prioritizing requirements is the choice of the most suitable technique, based on its applicability and the obtained results.

The aim of this manuscript is to contribute to this issue by comparing requirements prioritization methods, in particular, by assessing their practical relevance, ease of use and intuitiveness of the results. Therefore this manuscript does not analyze the methods from a mathematical point of view. The focus of the analysis takes a practical point of view, seeking to draw conclusions about the applicability and efficiency of the methods in a real case study. More specifically, this manuscript compares the AHP and ELECTRE I methods. AHP was chosen due to its popularity as a method for software requirements prioritization. ELECTRE I was also chosen, because we are not aware of any usage within the software requirements prioritization, even if it is widely adopted to assist the decision making process in multi-criteria decision problems. The evaluation of the implementation of the methods was carried out based on the consistency of the methods,

978-1-4799-8676-7/15/$31.00 copyright 2015 IEEE
SNPD 2015, June 1-3 2015, Takamatsu, Japan

the results and the questionnaires answered by the stakeholders.

This manuscript follows the structure next presented. Sect. 2 discusses some articles that compare requirements prioritization techniques within the development of software. In sect. 3, we present the main activities related to the process of prioritizing requirements within the development of software. Sect. 4 provides a short description of the MCDMs under analysis in this manuscript: AHP and ELECTRE I. We compare in sect. 5 these two MCDMs with respect to the ease of use and present the results obtained when the methods are applied to a real case study provided by a company that develops mass-market software products. The conclusions of the work and proposals for future work are discussed in sect. 6.

2. Related work

The scientific literature provides some publications that compare requirements prioritization techniques in the context of developing software products or systems. However, as far as we are aware of, the use of ELECTRE I in the context of prioritizing software requirements is not reported in the scientific literature.

An experimental comparison of five prioritization methods (AHP, binary search tree algorithm, XP planning game, 100 points method, and a combination of planning game with AHP) is presented in the master thesis written by Ahl [1]. The thesis puts those 5 methods into a controlled experiment. Ahl concludes that the binary search tree yields accurate results, is able to scale up and was the easiest method to use, making it the best method to use for prioritizing requirements.

Perini et al. present an empirical study aiming at evaluating two tool-supported requirements prioritization methods, AHP and CBRank [12], [13]. The authors focus on three measures: (1) ease of use, (2) time-consumption, and (3) accuracy. The experiment has been conducted with 23 experienced subjects on a real project with 20 requirements. Even if the resulting ranks from the two methods are quite similar, results show that for the first two characteristics CBRank overcomes AHP, while AHP performs better than CBRank for the accuracy. The majority of the users found CBRank the overall best method.

Svensson et al. describe a study that identifies how quality requirements (also known as non-functional requirements) are prioritized in practice at 11 companies developing software systems [17]. They found that ad-hoc prioritization and priority grouping of requirements are the dominant methods for prioritizing quality requirements. The results also show that it is common to use customer input as criteria for prioritization but absence of any criteria was also common. The results suggest that non-functional requirements by default have a lower priority than functional requirements.

Dabbagh and Lee propose an approach that considers both functional and non-functional requirements during the prioritization stage [4]. The outcome of applying the approach produces two separate prioritized lists of functional and non-functional requirements. The effectiveness of the proposed approach was evaluated through an empirical experiment aimed at comparing the approach with AHP and the hybrid assessment method (HAM). The results show that the approach outperforms AHP and HAM in terms of actual time-consumption, while preserving the quality of the results at a high level of agreement.

Another interesting source of material are descriptions of MCDMs, like the ones presented in [5], [18].

3. Requirements prioritization

Prioritization of requirements is an activity that can be framed within the process of negotiating the requirements. In most software projects, the number of candidate requirements is very high, making it literally impossible to implement all of them, due to restrictions and constraints in time and budget [8]. Given the set of candidate requirements, the main challenge is to identify the subset that maximizes the performance of the technical constraints (time and resources) and the preferences and critical needs of the stakeholders. The prioritization of requirements helps in finding this subset of candidate requirements.

The requirements prioritization is the process that defines a total order for a set of candidate requirements on the basis of which they can be divided into subsets, one for each iteration of the product development process. The prioritization process consists of three consecutive steps [9]:

Preparation: This is the step where the requirements are structured in accordance with the principles of the prioritization method being used. The set of stakeholders is selected, being provided with all the necessary information (about the requirements and the prioritization technique).

Implementation: In this step, stakeholders perform the prioritization of the requirements using the information that has been provided in the previous step. The criteria should be agreed by stakeholders before starting the prioritization.

Presentation: In this step, results are presented to the stakeholders. Some prioritization techniques involve some sort of calculations that must be performed before the results are presented.

3.1. Stakeholders

In software development, there are three situations related to the stakeholders: (i) one stakeholder; (ii) various stakeholders (a known number); (iii) many stakeholders (an unknown number).

In the case where there is only one stakeholder, the prioritization process becomes straightforward, because it is only necessary to consider the opinion of that stakeholder. When there are multiple stakeholders the process is more complex, because there are various opinions which are often in conflict. In these cases, it is necessary to reconcile the views of the stakeholders. Even more complex are the cases in which there are many stakeholders, in an unknown number. In these cases, it is necessary to select a small group of stakeholders that is representative of all of them. This choice must assure that the various roles or actors are equally and fairly represented. These representatives are the persons that participate in the process of prioritizing the requirements.

3.2. Criteria

The requirements can be assigned priorities according to different criteria. When using a single criterion to assign priorities to the requirements, it is easy to rank the requirements. If one uses more than one criterion, the challenges and the difficulties increase. In this case, it is not always obvious how to relate the various criteria to rank the requirements. In particular, it is known that the introduction of the cost, as a criterion, typically implies changing the priority of some requirements to a lower level, if their inclusion in the software product turns out to be costly.

Several criteria may be considered in prioritization: importance, utility, urgency, penalty, user satisfaction, time, cost, risk and volatility. However, it is not practical to consider too many criteria. The ones that must effectively be considered in a particular project should be chosen according to the specific situation.

3.3. Techniques

There are a number of techniques for requirements prioritization that have been proposed so far, and some provide support tools that can be used in real software projects [2], [7], [11], [12].

One of the factors that differentiates the techniques is the adopted metric. There are three types of metrics: ordinal scale, ratio scale and the absolute scale. The ordinal scale is less effective, since the requirements are ordered by order of importance, This scale identifies if a given requirement is more important than another one, but not how much important it is. The ratio scale is more powerful since it allows to quantify how much more important a requirement is relative to another one, using, for example, percentages (0–100%). However, the most powerful scale is the absolute one, because it can be used in situations where you can assign a number [2].

The following techniques are among the most popular ones used for prioritizing software requirements.

Top-Ten Requirements: In this technique, the stakeholders choose the set of the ten most important requirements, without establishing any order between them [2]. This makes the approach suitable for various stakeholders of equal importance.

Numerical Assignment (Grouping): This technique consists in grouping the requirements into groups with different priorities. The number of groups can vary, but the most common is to use three groups (e.g., critical, standard, optional). A problem that arises is the tendency of the stakeholders to classify all requirements as critical. A way to go around this problem is to set a limit number of requirements for each group. However, this solution forces the stakeholders to divide the requirements according to certain groups. The result of applying this technique are requirements prioritized in accordance to an ordinal scale, i.e., all the requirements in a group have the same priority [2].

Ranking: In this technique, the requirements are totally ordered, based on an ordinal scale. The requirements are ordered without ties in the ranking. This means that in a project with n requirements, the most important requirement has level 1 and the less important requirement has level n. Each requirement has a single value, but there is no way of knowing the relative difference between two requirements [2].

100-Dollar: Stakeholders take in 100 imaginary units to distribute among the requirements. These units may represent different aspects: money (cost of implementation), importance, penalty, hours [10]. The result is presented on a scale ratio. Problems arise when a stakeholder decides to put all his/her units on a single requirement, heavily distorting the result of the prioritization. A solution is to limit the amount of units to assign to each requirement. However, this can prevent the stakeholders to prioritize the requirements according to their genuine needs.

AHP (Analytic Hierarchy Process): This popular MCDM is described in section 4.1.

Setting a process for prioritizing requirements for a given context is not straightforward and has many challenges. Among these challenges is the choice of the most adequate technique to be adopted in a given project. In this manuscript, we study two possible techniques: AHP e ELECTRE I. More information about these two techniques can be found in the next section.

4. AHP and ELECTRE I

In this section, we provide a short description of the MCDMs under analysis in this manuscript: AHP and ELECTRE I. More details about these MCDMs can be obtained in the references.

4.1. AHP

The AHP (Analytic Hierarchy Process) [15], [16] is based on the human capacity to make pertinent judgments about small problems. Thus, AHP organizes the criteria in a hierarchical way to divide the general problem into smaller ones.

AHP is based on four main axis: (1) reciprocal judgements; (2) homogeneous elements; (3) hierarchical structure; (4) ranked expectations.

In AHP, one compares all the possible pairs of requirements to determine an ordered list of the requirements according to their importance. Usually a scale of 1 to 9 is used, where 1 represents equal importance and 9 represents absolutely more importance. During the process, if n requirements are considered, $\frac{n \times (n-1)}{2}$ comparisons need to be made, which for a large number of requirements does not ease the application of the technique. The result is a set of requirements prioritized along a ratio scale. The synthesis of AHP combines multidimensional scales of measurement into a unidimensional scale of priorities. AHP is highly dependable, since the great level of redundancy in the pairwise comparisons makes the process immune to comparison errors [9]. Another advantage is the fact that the values assigned in the pairwise comparisons are based on experience, intuition and real data. Thus, AHP can handle both the qualitative and the quantitative aspects of a decision problem. Additionally, the fact that the resulting priorities are related and based on a ration scale allows useful evaluations of the requirements.

However, AHP also presents some limitations, such as the case with its inadequate application, i.e., in unfavorable environments where applying AHP is perceived as an excessive simplification or as a waste of time [6]. Another evident limitation is the excessive number of comparisons that one needs to perform.

In AHP, the problem is structured as an hierarchy, followed by the prioritization process itself. Thus, the process can be divided in four steps: (1) organization of the hierarchical structure, (2) comparison of the criteria and the requirements, (3) calculation of the criteria weights and the requirements priorities, and (4) calculation of the consistency ratios.

4.2. ELECTRE I

The ELECTRE I MCDM is one of the methods of the ELECTRE family (ELimination Et Choix TRaduisant La REalit), which integrates a total of seven methods (ELECTRE I, ELECTRE Iv, ELECTRE IS, ELECTRE II, ELECTRE III, ELECTRE IV, ELECTRE TRI) [14]. This family belongs to the outranking methods that are based on building a outranking relationship that incorporates the preferences established by the decision-maker when faced with problems and available alternatives.

ELECTRE I is intended to multi-criteria problems involving choice and selection. This method can be applied in problems where the alternatives can be represented in different scales. The aim is to get the subset of alternatives, such that any alternative that does not belong to this subset is more important than, at least, one alternative in the subset. This subset, the smallest possible, is not the set of the best alternatives, but rather those which are preferred in most assessment criteria and that do not cause an unacceptable level of dissatisfaction in the other criteria. In ELECTRE I, the preferences are modeled using outranking binary relations, S, which means that "is at least as good as" [5]. Considering two actions a and b, four situations can occur:

- aSb and not bSa, i.e., aPb (a is preferable with respect to b)
- bSa and not aSb, i.e., bPa (b is preferable with respect to a)
- aSb and bSa, i.e., aIb (a is indifferent to b)
- not aSb and not bSa, i.e., aRb (a is incomparable to b)

The construction of an outranking relationship is based on two fundamental concepts [5]:

Concordance: for an outranking relationship aSb to be validated, a sufficient majority of the criteria should be in favor of the statement.

Discordance: when the condition of agreement is valid, none of the criteria of the minority is strongly in opposition to the aSb statement.

These two conditions must be satisfied to validate statement aSb.

The ELECTRE methods follow two general procedures: construction of one or more outranking relationships, followed by an exploration of the procedure. The construction of one or more outranking relationships helps to compare each pair of actions in a comprehensive manner. The exploitation is used to develop recommendations based on the results obtained in the first phase.

5. Comparing AHP and ELECTRE I

The main aim of this manuscript is to compare the AHP and ELECTRE I MCDMs, when applied to a real problem, with respect to the ease of use. Thus, these two methods were applied to a real problem provided by a software company that develops mass-market software products.

5.1. Application of the methods

The real example used to compare the MCDMs is a software application for managing projects (human resources, cost, deadlines, etc.). As the project has been provided by the software company, the subset of candidate requirements and the criteria used in the comparison were selected by the members of that company. Similarly, the software company also selected the stakeholders involved in the requirements prioritization process. Two of the stakeholders were member of the software company and the other two were members of the client.

In total, 20 requirements of the project were selected to be prioritized, 6 prioritization criteria were considered and 4 stakeholders S1-S4 (2 belonging to the company and 2 belonging to the client) were involved. Since the stakeholders have different perspectives on the project, the criteria were not exactly the same for all of them. We decided to make this distinction in the criteria, since adding a criterion and compare it with indifference (the whole matrix with 1 in the AHP or all requirements with the same value in the ELECTRE I) gives the same result as deleting that prioritization criterion. So, each stakeholder only performed the prioritization criteria related to his/her perspective on the project.

To facilitate the application of both methods, two spreadsheets were created, one for each method. After introducing the data in the spreadsheets, the final results are automatically created. A document, describing the requirements, the prioritization criteria and the application of methods, was provided to the stakeholders to make them aware of all the important issues related to the prioritization process.

The stakeholders applied the methods, only after being familiar with the requirements, the criteria, and the methods. Each stakeholder initially applied ELECTRE I and later AHP. After verifying the results obtained with both methods, stakeholders filled in a questionnaire to compare them.

5.2. Consistency analysis for AHP method

After obtaining the results of the application of AHP by all stakeholders, we analyzed the consistency of all comparison tables in order to verify the validity of the obtained results. The consistency ratios were calculated according to the method proposed by Saaty, obtaining the results presented in table 1.
These results show that $1/6$ (3 out of 18) of the data are inconsistent (consistency ratio > 0.10), which can be caused by the large number of comparisons (in this example, $4 \times \frac{20 \times 19}{2} = 760$) that each stakeholder had to undertake to apply the AHP method. For this reason and because this represents a small percentage of the data, the comparisons were not made again. Note that we have 18 different comparisons, since stakeholders S1 and S2 both adopted 4 criteria, while stakeholders S3 and S4 both adopted 5 criteria.

criterion	S1	S2	S3	S4
product differentiation	0.1397	0.1328	0.0290	0.0339
customer impact	—	—	0.0484	0.0339
penalty	0.0744	0.0916	0.0650	0.0339
cost	0.0755	0.0926	0.0000	0.0000
strategic benefit	—	—	0.0563	0.0339
integration with tools	0.0234	0.1166	—	—

Table 1. Consistency ratios for AHP.

	equal/identical positions	very different positions
S1	20%	5%
S2	15%	15%
S3	10%	30%
S4	55%	5%

Table 2. Comparison of the individual results for AHP and ELECTRE I.

5.3. Analysis of the results

The results obtained with the application of the two methods for each stakeholder were compared by calculating the percentage of requirements which are in equal/identical positions and the percentage of those that are in very different positions (i.e. whose positions differ more than 9). The results are shown in table 2. Requirements R9 e R16 are in equal positions. It is important to notice that in ELECTRE I, requirements

R12 and R16 are tied in the same position, so R16 can be at the 9th or 10th positions. Requirements R13, R15, and R20 are in very different positions in both methods, which means that they are ranked in positions with a difference higher than 10. For example, requirement R13 is at the 5th position in AHP and at the 17th position in ELECTRE I. One can verify that for 3 of the 4 stakeholders (S1, S2, S4), the number of requirements in equal/identical positions is equal to or greater than the number of requirements in very different positions. One was expecting that result for all stakeholders, since they are prioritizing the same requirements with the same criteria.

Next, we have created an overall score for each method. For the AHP method, since there are numeric values, one calculates for each requirement the weighted average for the priorities of all stakeholders, in order to obtain the global priority for this requirement, according to the following formulae.

$$GP_r = \frac{\sum_{i=1}^{n} P_{r,i}}{n} \quad \text{where:}$$
$GP_r =$ global priority for requirement r
$P_{r,i} =$ priority for requirement r for stakeholder i
$n =$ number of stakeholders
(1)

After calculating the global priorities for all requirements, the global result for AHP was obtained. For ELECTRE I, since there are no numeric values associated with the priorities, it was necessary to assign values, so that we could calculate a global result. The example presented next shows an application of the adopted procedure. A numerical scale, from 1 to 20 (the number of requirements), was defined. A given value is assigned to each requirement. the most important requirement gets the lowest value. Since there are groups where all requirements have the same priority, to those requirements one assigns the average of the values that would be assigned to them if there were no ties among the requirements.

Once the global results were obtained, the two methods were also compared by calculating the percentage of requirements in equal/identical positions and in very different positions, as was accomplished for each stakeholder.

Table 3 presents the global results for both methods and their comparison. Only 15% of the requirements are in very different positions. One can also observe that 10% of the requirements are in equal/identical positions, being the other 75% of the requirements in close positions. This result is somehow expected, since the comparison of the individual results for the stakeholders presents similar results.

Finally, the results of each stakeholder were compared with the overall result of each method. The result of this comparison can be seen in tables 4 and 5.

It turns out that there is a great difference between the results of AHP and ELECTRE I. For AHP and for most of the stakeholders, the percentage of requirements in very different positions is higher than the percentage of requirements in equal/identical positions. With respect to ELECTRE I, it happens exactly the opposite. The percentage of requirements in equal/identical positions for all stakeholders is higher than the percentage of requirements in very different positions and it presents much higher values than those recorded for AHP. This difference is due to the fact that the results for AHP for the various stakeholders are more different among them than in ELECTRE I. So, the overall result for AHP diverges more from the individual results. By the contrary, for ELECTRE I, the overall result turns out to be closer to each one of the individual results.

	equal/identical positions	**very different positions**
S1	15%	5%
S2	5%	25%
S3	0%	5%
S4	5%	10%

Table 4. Comparison of the global results for AHP.

	equal/identical positions	**very different positions**
S1	15%	0%
S2	10%	5%
S3	25%	10%
S4	45%	5%

Table 5. Comparison of the global results for ELECTRE I.

5.4. Analysis of the questionnaire

As previously indicated, each stakeholder has filled in a questionnaire so that we could get his/her opinion with respect to the methods. Based on the answers to the questionnaires, the following results were obtained:

1. Did you already know some of the prioritization methods?

AHP	25%
ELECTRE I	25%
AHP and ELECTRE I	0%
None	50%

#	AHP		ELECTRE I		#	AHP		ELECTRE I	
1st	R8	0.0723	R2	4.000	**11th**	R7	0.0494	R1	10.625
2nd	R5	0.0723	R20	6.500	**12th**	R11	0.0472	R6	10.875
3rd	R9	0.0699	R9	8.250	**13th**	R1	0.0462	R19	11.125
4th	R2	0.0656	R15	9.000	**14th**	R17	0.0450	R4	11.250
5th	R13	0.0644	R7	9.125	**15th**	R15	0.0355	R17	12.125
6th	R12	0.0614	R8	9.250	**16th**	R20	0.0347	R10	12.375
7th	R4	0.0584	R11	9.250	**17th**	R10	0.0330	R3	13.375
8th	R3	0.0557	R5	9.500	**18th**	R14	0.0307	R18	13.750
9th	R16	0.0511	R12	10.000	**19th**	R19	0.0287	R13	14.500
10th	R6	0.0508	R16	10.000	**20th**	R18	0.0276	R14	15.125

Equal/identical positions	10%
Very different positions	15%

Table 3. Comparison of the global results for AHP and ELECTRE I.

2. Which method is easier to apply?

AHP	0%
ELECTRE I	100%
They are similar	0%

3. Which type of results do you prefer?

AHP	0%
ELECTRE I	50%
They are indistinct	25%
No answer	25%

4. Which method presents more reliable results?

AHP	75%
ELECTRE I	0%
They are similar	25%

5. Globally, which is the most feasible method?

AHP	25%
ELECTRE I	75%
They are similar	0%

6. Average time (min) to apply the methods

AHP	94
ELECTRE I	30

One can thus conclude that all stakeholders found the ELECTRE I method much easier to apply, taking on average $1/3$ of the time for applying AHP. In relation to the types of results, most stakeholders prefer the results of AHP since it presents the requirements totally ordering, with numerical priorities assigned to all requirements, thus allowing one to know how much more (or less) important is a requirement in relation to another. However, due to the complexity of applying the AHP method, the stakeholders claim that only ELECTRE I can be applied in software projects.

6. Conclusions

After analyzing the obtained results, it is possible to confirm the intuition that ELECTRE I is more easily applicable than AHP. Since both MCDMs were applied to the same case study, it is possible not only to draw conclusions about each of the separate methods, but also to compare the advantages and disadvantages of one over the other.

Although we are not aware of any use of ELECTRE I for prioritizing software requirements, this method is subject to fewer mistakes in the comparisons of the requirements than AHP , as these are carried out differently, resulting in a large difference in the numbers. In fact, due to its inherent complexity, AHP becomes even impractical in software projects with a large number of requirements.

Despite the innovation that resulted from applying ELECTRE I in the context of a software project, this MCDM proved to be a good alternative to consider, since it combines an easy application with a great consistency in the obtained results, thereby becoming a fairly reliable method.

The stakeholders were an integral and very important part in this study, not only because they performed the methods, but also because through the questionnaires they expressed their opinions and preferences about both methods. Despite we reached the conclusion that ELECTRE I is easier to apply than AHP, the majority of the stakeholders prefer the kind of results provided by AHP. This fact can be explained due precisely to the kind of results obtained through this method. While in ELECTRE I, requirements are grouped into ordered groups but without assigning values, in AHP there is a total order, being assigned values to all requirements. To know how much more important a requirement is relative to another seems to weigh more on the preference of the involved stakeholders.

Although the results are presented in distinct sorting scales, there were no significant differences both in the individual and the overall results.

The major limitation of the study which needs to be acknowledged when interpreting the results was the small number of participants. It would be favorable to include more participants, especially with different backgrounds, roles, ages, etc. Another limitation results from the fact that only one real-world case study was used. Again, gathering data from studies related to a larger number of case studies would make the study more solid. Finally, obtaining the opinions of the participants through questionnaires is limited. due to their inability to probe responses. Complementing the questionnaires with interviews is a possible solution, to permit a personal contact with the participants and thus obtain information that cannot be obtained in a written form. Therefore, all results and findings presented in this study should be taken with some reserve. Anyway, it is believed that the results obtained allow to obtain valid conclusions and that they may serve as the basis to conduct a more detailed study for the two methods (AHP and ELECTRE I) or even as the basis for comparing the application of these and other methods in the prioritization of software requirements.

Another line of future work is related with studying the adoption of ELECTRE I for prioritizing software requirements, since the method proved to have practical application in real software engineering projects.

References

[1] Ahl V. *An experimental comparison of five prioritization methods: investigating ease of use, accuracy and scalability*. Master Thesis, Blekinge Institute of Technology, Ronneby, Sweden, 2005.

[2] Berander P. and Andrews A. *Requirements prioritization*. In Engineering and Managing Software Requirements, Aurum A. and Wohlin C. (eds). Springer, chap. 4, 2005. DOI 10.1007/3-540-28244-0_4.

[3] Brooks F. *The mythical man-month: essays on software engineering* Addison-Wesley Longman, 1995.

[4] Dabbagh M. and Lee S. P. *An approach for integrating the prioritization of functional and nonfunctional requirements*. The Scientific World Journal, article ID 737626, 2014. DOI 10.1155/2014/737626.

[5] Figueira J., Greco S. and Ehrgott M. *Multiple criteria decision analysis: state of the art surveys*. Springer, 2005.

[6] Grandzol J. *Improving the faculty selection process in higher education: a case for the Analytic Hierarchy Process*. Association for Institutional Research, v. 6, 2005.

[7] Karlsson J. and Ryan K. *Supporting the selection of software requirements*. In 8th International Workshop on Software Specification and Design (IWSSD ’96), pp. 146–149, IEEE CS Press, 1996. DOI 10.1109/IWSSD.1996.501157.

[8] Karlsson J. and Ryan K. *A cost-value approach for prioritizing requirements*. IEEE Software 14(5):67–74, 1997. DOI 10.1109/52.605933.

[9] Karlsson J., Wohlin C. and Regnell B. *An evaluation of methods for prioritizing software requirements*. Information and Software Technology 39(14-15):939–947, 1998. DOI 10.1016/S0950-5849(97)00053-0.

[10] Khan K. *A systematic review of software requirements prioritization*. M.Sc. Thesis, Bleking Institute of Technology, Ronneby, Sweden, 2006.

[11] Lehtola L. and Kauppinen M. *Suitability of requirements prioritization methods for market-driven software product development*. Software Process Improvement and Practice 11(1):7–19, 2006. DOI 10.1002/spip.249.

[12] Perini A., Ricca F. and Susi A. *Tool-supported requirements prioritization: comparing the AHP and CBRank methods*. Information and Software Technology 51(6):1021–1032, 2009. DOI 10.1016/j.infsof.2008.12.001.

[13] Perini A., Susi A. and Avesani P. *A machine learning approach to software requirements prioritization*. IEEE Transactions on Software Engineering 39(4):445–461, 2013. DOI 10.1109/TSE.2012.52.

[14] Roy B. *The outranking approach and the foundations of ELECTRE methods*. Theory and Decision 31(1):49–73, 1991. DOI 10.1007/BF00134132.

[15] Saaty T.L. *How to make a decision: the Analytic Hierarchy Process*. Interfaces 24(6):19–43, 1994.

[16] Saaty T.L. and Vargas L. G. *Models, methods, concepts and applications of the analytic hierarchy process*. Springer, 2001.

[17] Svensson R.B., Gorschek T., Regnell B., Torkar R., Shahrokni A., Feldt, R. and Aurum, A. *Prioritization of quality requirements: state of practice in eleven companies*. In 19th IEEE International Conference on Requirements Engineering (RE 2011), pp. 69–78, 2011. DOI 10.1109/RE.2011.6051652.

[18] Zavadskas, E.K., Turskis, Z. and Kildiene, S. *State of art surveys of overviews on MCDM/MADM methods*. Technological and Economic Development of Economy 20(1):165–179, 2014. DOI 10.3846/20294913.2014.892037.