# Prioritizing Requirements: An Experiment to Test the Perceived Reliability, Usability and Time Consumption of Bubblesort and the Analytical Hierarchy Process

Geurt Johan van Tuijl, Wouter Leenen, Zhengru Shen,
Inge van de Weerd, Sjaak Brinkkemper

Department of Information and Computing Sciences, Utrecht University,
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
{g.j.vantuijl1, w.a.leenen, z.shen}@students.uu.nl
{i.vandeweerd, s.brinkkemper}@cs.uu.nl

**Abstract.** Software vendors often face the difficult task to deal with large amounts of requirements that enter the company every day. When dealing with this vast amount of requirements, the notion of deciding which requirements will be addressed first, and which will be addressed later, is an important decision. To support software development teams in decision-making, different prioritization techniques are discussed in previous literature. In this paper, two existing prioritization techniques called Analytical Hierarchy Processing (AHP) and Bubblesort are investigated in order to measure their outcome in terms of usability, time consumption and perceived reliability. By conducting an experiment among Dutch Master students, we discovered that Bubblesort outpaced AHP on all aspects, although this could not be supported statistically. However, based on our findings, we can conclude that there is a strong indication that Bubblesort is considered favorable compared to AHP even though it receives less attention in current literature.

**Keywords:** Requirements prioritization, Bubblesort, Analytic Hierarchy Processing, Software Product Management

## 1 Introduction

Software vendors have to handle the requirements that enter the company through customers, sales & marketing, and research & development every day. Typically, customer demand exceeds the pace at which software is developed [1], usually causing the final release to be outdated before it has been published. Eliciting the required information from customers can be difficult to achieve, especially when multiple customers with diverse expectations are involved [2]. Customer involvement is therefore a major contributing factor to company success [3].

An important issue when dealing with this vast amount of requirements is the notion of deciding which requirements are being addressed first and which will be addressed later. This difficulty, i.e. the prioritization of requirements, is recognized as an import activity in product development [4, 5] and product management [6, 7]. The process for prioritizing software requirements faces the difficult task to be simple and fast, however, also needs to provide accurate and trustworthy results [8]. A solution to this problem can be found in existing techniques such as the Binary Priority List [1], Analytical Hierarchy Process [9] and Hierarchical Cumulative Voting [10]. Regardless of which technique is carried out by the user, each technique should be able to support the product manager in decision-making but also needs to decide which requirements are the least important.

### 1.1 An Overview of Requirements Prioritization Techniques

Researchers, as well as professionals in industry, have proposed many requirements prioritization methods and techniques. Different techniques have been discussed considerably and compared with others in experimental studies [11], case studies [12], empirical studies [5, 13] and literature studies [18]. According to a review proposed by Racheva et al. [14], these techniques could be classified into two main categories: techniques that are applied to small amounts of requirements (small-scale) and techniques that scale up very well (medium-scale or large-scale). Examples of small-scale techniques include round-the-group prioritization, multi-voting system, pair-wise analysis, weighted criteria analysis, and the Quality Function Deployment approach. Among medium-scale or large-scale techniques, the MoSCoW technique, the Binary Priority List, the Planning Game and the Wiegers's matrix approach are frequently used. Considering the sometimes complex algorithms used in existing techniques, often supporting tools are needed to deal with large amounts of requirements.

Another classification for requirements prioritization techniques is given by Berander et al. [15]. Similar to Racheva et al. [14], they also divided existing techniques into two main categories: (1) techniques which assume that values can be assigned, by an expert, to different aspects of requirements and (2) methods that include negotiation approaches in which requirements priorities result from an agreement among subjective evaluation by different stakeholders. Examples of techniques that apply to the first category are Analytical Hierarchy Process (AHP) [9], Cumulative Voting, Numerical Assignment, Planning Game and Wieger's method. An example of the second category would be the Win-Win approach.

The prioritization technique that is popular among companies and researchers is the AHP technique. AHP is a multiple criteria decision-making method that has been adapted for prioritization of software requirements [16, 8]. In a software project that has $n$ requirements, a decision maker requires $n(n-1)/2$ pair-wise comparisons. Advantages of AHP are that it is trustworthy and has a strong fault tolerance. Along with AHP, another technique called Bubblesort [17] is one of the simplest and most basic methods that have been used for requirements prioritization and is similar to AHP in many ways [18]. However, Bubblesort receives little attention in literature compared to AHP even though both perform well according to other studies [16]. A problem that arises for both AHP and Bubblesort is the scalability. As soon as the number of requirements exceeds twenty, the prioritizing procedure takes too long. However, solutions such as structuring the requirements in a hierarchy of interrelated requirements [16], or applying machine learning techniques [18] have been proposed to overcome this problem.

The variety of prioritization techniques makes it difficult to select the most appropriate one for a specific project situation. Still, the prioritization of requirements proves to be a fundamental part of the responsibilities for a software product manager but it can be a quite demanding, difficult and time consuming task [18, 19]. Even small software product companies that often deal with smaller amounts of product requirements, could benefit from structured techniques to guide them to the best possible software product releases.

### 1.2 Problem Statement

Concluding from a research of Karlsson et al. [16], Bubblesort and AHP performed the best compared to four other prioritization techniques in terms of reliability, ease of use and fault tolerance when dealing with relatively small amounts of requirements. The reason to center research upon AHP and Bubblesort is because both perform really well according to other studies [16]. Still, there is significantly more literature devoted to AHP. Furthermore, both techniques can easily be applied on a small amount of requirements, which makes it easier to use in an experiment setting [21]. As stated, AHP receives a lot of attention whereas Bubblesort does not, and for this reason, we will discuss both techniques and elaborate on the specific differences between them by providing an experiment with the aim to extend current literature on Bubblesort and AHP. The application of the techniques will be performed by assessing and comparing the results of both techniques through a prioritization experiment with students at a large university in the Netherlands. This experiment will provide specific information about the perceived reliability, time consumption, and usability for both techniques. The research question we want to answer is:

*How do the requirement prioritization techniques Bubblesort and Analytical Hierarchy Process perform when compared to each other in terms of perceived reliability, time consumption, and usability?*

The next five sections are structured as follows. Section 2 presents the applied research approach and provides a clear explanation for this choice. Section 3 provides information about the execution of the experiment. Section 4 provides a brief explanation of how to interpret the results, and follows with the actual results for the variables tested for each of the techniques. Section 5 includes the overall conclusions. Finally, section 6 provides a discussion, goes into the limitations of this research, and some suggestions for future research.

## 2 Experimental Design

The goal of this experiment is to compare two prioritization techniques, AHP and Bubblesort, in terms of usability, time consumption and perceived reliability.

## 2.1 Subjects

For this experiment, we provide twelve students from Utrecht University, The Netherlands, with a list of requirements for a new version of Google Maps. Each subject uses Google Maps regularly. Furthermore, each subject has knowledge of the importance of requirements prioritization and has approximately the same age in order to maintain integrity and diminish variable influences on the results. The subjects did not receive any credits or reward for their participation.

## 2.2 Objects and Instrumentation

We created a list of twenty requirements[1] for a widely used web-based software package, called Google Maps. Google Maps is a web mapping service application and technology provided by Google. It allows the user to navigate anywhere on earth to view satellite imagery, maps, terrain, 3D buildings and street views. The reason for selecting Google Maps is because it is a well-known web based software product and thus widely-used on a daily basis, causing users to have different experiences and deficiencies about the application.

Each subject is provided with the list of requirements and with Excel based custom-made tools for AHP as well as Bubblesort. Both tools provide the subject with a graphical representation of the algorithm used behind AHP and Bubblesort. Since both tools were made in an Excel environment, any possible influences caused by the spreadsheets can counterbalance each other. Along with the tools, subjects were provided with instructions on how to use both tools.

## 2.3 Data Collection and Analysis Procedure

Data is collected by means of a questionnaire. The time consumption is registered through identifying the exact starting time and the exact ending time of the subjects excluding any breaks in between. The data is collected in an informal experiment setting at the university, where the subjects use the provided excel based tools along with the questionnaire. When the subjects are finished, the data is entered manually in SPSS (a statistical analysis tool) in order to measure predefined outcomes.

To perform an effective and meaningful comparison of both requirements prioritization techniques, we take three evaluation factors into account: (1) *time consumption*, which is defined as the interval between the time the user starts prioritizing the requirements and the time when the user is finished prioritizing [21] using either of both prioritization techniques; (2) *ease of use*, which is measured by means of direct questions to each participating subject concerning complexity and number of comparisons; and (3) *perceived reliability*, which is defined as the degree to which the final requirements ranking differs from the ideal target ranking. In this research, the ideal target ranking is the ranking the subject has in mind based upon

[1] The list of requirements can be found at http://people.cs.uu.nl/weerd/reew-appendixa.pdf

implicit knowledge. This ideal ranking per subject is the ranking the subject made manually based upon implicit knowledge and without the use of a tool or technique.

### 2.4 Validity Evaluation

In this experiment, students are used as subjects instead of businessmen; hence reasoning and interpretation of requirements might not be representative for software product companies [21]. Still, the students do have a thorough knowledge about the software product and the presented requirements at hand, and therefore, reasoning and interpretation threats should influence the results only minimally.

On an individual level, there is the risk for students being influenced by familiarity with the requirements or learning from their experience with the first prioritization method upon switching techniques. However, by counterbalancing the groups we tried to minimize this threat. Furthermore, the subjects of the experiment could be influenced by fatigue. Hence, we limited this threat by keeping the requirements understandable and the number of requirements low.

## 3 Execution

Before starting with the experiment, the subjects were randomly divided into two groups. Each subject that is part of group A will apply Bubblesort on the list of requirements whereas the subjects from group B will apply AHP. When both groups are finished, they will switch techniques, i.e. subjects from group A will then apply AHP and subjects from group B will then apply Bubblesort. The reason for alternating the techniques is to take away any habituation influences of the subjects on the results; i.e. any advantage gained by using one technique before the other for the whole group. In addition, if all subjects start with the same technique, they might be able to influence the results of the other technique due to time constraints or time consumption. When both groups are finished, each subject will fill in a brief questionnaire. This questionnaire encompasses the evaluation of both tools by assigning a grade to the perceived reliability, the usability, and the time consumption of both tools. Afterwards, the results of each subject will be analysed.

## 4 Analysis of the Results

The results from the questionnaire can be found in Table 1. Each subject that took part in this experiment is assigned to a group. Subjects 1 to 6 were part of group A whereas subjects 7 to 12 were placed in group B. The requirements prioritization process started simultaneously, where group A started with the Bubblesort tool (the abbreviation BS in Table 1) and group B started with the AHP tool. When both groups were finished, each subject had to fill in a brief questionnaire with respect to their experiences towards both tools.

In Table 1, each number in the table represents a grade (on a scale of 1-10) that the subject assigned towards the tool. For instance, subject 1 assigned an 8 to the reliability for AHP and a 4 for the reliability of Bubblesort. In addition to these grades, the consistency ratio is also taken into account for the AHP tool. Since this ratio only applies to the AHP tool, this ratio is not available for the Bubblesort tool. The consistency ratio is presented in the AHP column between the two brackets and describes the consistency of the prioritization process, i.e. how consistent was the subject when prioritizing requirements. According to Saaty [9], a ratio of less than 0.10 is considered acceptable. Furthermore, the time consumption (in minutes) is measured throughout the prioritization process to indicate how much time it took to complete the prioritization process. The results are located within the AHP and Bubblesort columns underneath the overarching 'time consumption' column.

**Table 1.** Questionnaire results, specified per subject and group

| | Reliability | | Ease of use | | Time consumption | | Overall score | |
|---|---|---|---|---|---|---|---|---|
| | AHP | BS | AHP | BS | AHP | BS | AHP | BS |
| Group A | | | | | | | | |
| Subject 1 | 8(.172) | 4 | 5 | 3 | 6(31) | 2(59) | 8 | 6 |
| Subject 2 | 8(.093) | 7 | 7 | 6 | 5(66) | 6(42) | 8 | 6 |
| Subject 3 | 7(.054) | 9 | 8 | 9 | 8(14) | 7(16) | 7 | 8 |
| Subject 4 | 5(.023) | 7 | 5 | 9 | 5(33) | 5(29) | 5 | 8 |
| Subject 5 | 7(.039) | 8 | 6 | 4 | 7(29) | 6(35) | 7 | 7 |
| Subject 6 | 7(.040) | 8 | 6 | 9 | 5(75) | 9(40) | 7 | 9 |
| *Average* | *7(.070)* | *7,2* | *6,2* | *6,7* | *6(41)* | *5,8(37)* | *7* | *7,3* |
| Group B | | | | | | | | |
| Subject 7 | 5(.075) | 7 | 2 | 4 | 10(30) | 9(20) | 4 | 6 |
| Subject 8 | 6(.019) | 9 | 7 | 9 | 4(60) | 9(30) | 6 | 8 |
| Subject 9 | 6(.144) | 8 | 7 | 9 | 5(57) | 8(30) | 6 | 8 |
| Subject 10 | 8(.152) | 10 | 6 | 9 | 6(24) | 9(18) | 6 | 9 |
| Subject 11 | 7(.061) | 7 | 7 | 8 | 3(25) | 6(13) | 6 | 7 |
| Subject 12 | 8(.061) | 4 | 5 | 6 | 4(30) | 5(30) | 6 | 4 |
| *Average* | *6,7(.085)* | *7,5* | *5,7* | *7,5* | *5,3(38)* | *7,7(24)* | *5,7* | *7* |
| **Average** | **6,83** | **7,33** | **5,92** | **7,08** | **5,67** | **6,75** | **6,33** | **7,17** |

In addition to Table 1, a graphical overview is presented in the form of a bar graph in Figure 1. In the following subsections, we will elaborate upon each measurement.

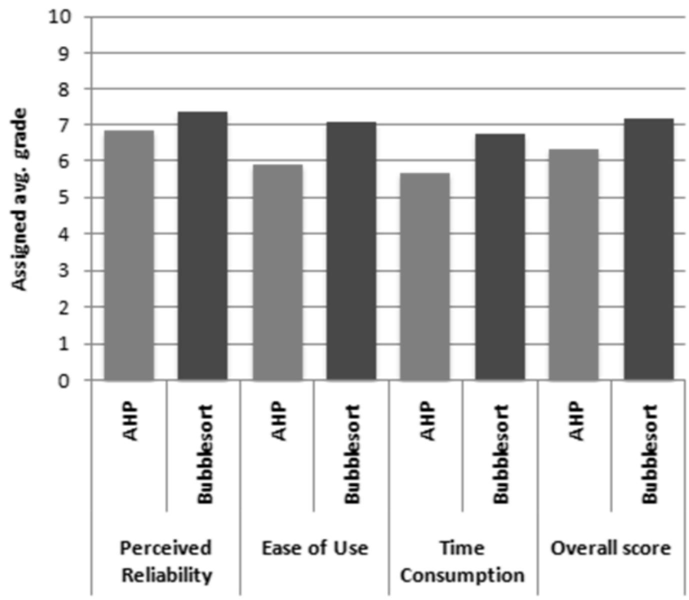


**Figure 1.** Graphical overview of the results, divided per measurement.

### 4.1 Perceived Reliability

The most left column of Table 1 shows the obtained reliability scores from the subjects involved in the experiment. These scores, on a scale of 1 (low) - 10 (high), were assigned by each subject based on their consensus with the actual ranking obtained through each technique. Furthermore, these results should be considered as 'perceived' because the results are based on the basic reasoning process of non-actual stakeholders in the development of this Google Maps release. When testing for significance between both groups A and B, using the Mann-Whitney Test, there proved to be no significant difference between both groups ($\alpha$ =.05). Therefore, group A did not score significantly better than group B en thus we can merge both group scores and assess the full group result.

Before continuing with the evaluation of the reliability scores, the consistency of the AHP technique needs to be taken into consideration. As can be seen from table 1, the contents of the brackets, next to the assigned reliability score for AHP, provide the Consistency Ratio (CR) for the valuations provided by the subjects. This consistency ratio is composed of the consistency index and the random index [8, 20]. Any CR lower than .10 is considered acceptable, however, according to Saaty [9] these 'acceptable' scores are hard to obtain. Looking at the results, consistency proves to be very good with only three people obtaining CR-scores higher than .10. For one particular subject we observed a remarkable consistency rate in AHP for filling in the matrix cells, indicating a very inconsistent valuation (.172) of the requirements, while his appraisal of the obtained ranking (8) was quite high. For that particular subject we can conclude that by providing a very inconsistent valuation of the requirements, his results should be considered unreliable.

Using the Wicoxon Signed Ranks test, the overall deviation between the scores assigned by the subjects for AHP (avg. 6.83) and Bubblesort (avg. 7.33) proved not to be significant ($\alpha$ =.05). Nevertheless, the difference between the mean of the scores of the perceived reliability for the two techniques suggests that Bubblesort provides better results on 'perceived' reliability.

### 4.2 Ease of Use

Ease of use (i.e. the usability) is measured through the use of a questionnaire. The subjects were kindly asked to rate the usability of Bubblesort and AHP on a scale from 1-10, where 10 is the highest score.

When calculating the average score of the usability of both tools, we see that Bubblesort has a mean of 7.08, whereas AHP does not reach beyond 5.91. This result is as we expected since the Bubblesort tool required less time to complete, and is therefore regarded as easier to use. This conclusion could also be derived from the fact that group A started with Bubblesort first. Since group A did not have a clue how the AHP tool looked like, they gave Bubblesort an average of 6.7. Group B, which started with AHP first, gave Bubblesort an average grade of 7.5. Since the tool of AHP perceived to be more complex in use, the subjects subsequently found the Bubblesort tool easier to use and thus assigned a higher grade to it. The usability score of both techniques could be influenced by the created Excel-based tools. During the prioritization process, some subjects faced difficulties with respect to the completion of the Excel document whereas other subjects did not face any troubles. Nonetheless, considering both tools were made with Excel, any possible drawbacks caused by the limitations of spreadsheet software during the prioritization process would likely counterbalance each other, thus negating their effect.

Analyzing the usability between group A and group B, based on the Mann-Whitney Test ($\alpha$ =.05) we again did not find a significant difference between both groups. Since no significant difference was found, we decided to merge both groups A and B. When we subsequently use the Wilcoxon Signed Ranks test to test for significant differences between the scores of AHP and the scores of Bubbelsort, we again did not find any significant differences ($\alpha$ =.05). Although the average scores deviated, based on our sample size, this deviation cannot be assigned to the population yet. There is only a strong presumption that Bubblesort might be easier to use than AHP.

### 4.3 Time Consumption

The averages in Table 2 show that Bubblesort performs better compared to AHP in terms of the actual time consumption. The results from start/end time were recorded by the subjects. As illustrated in Table 2, the difference in time consumption (sample mean) of the two prioritization techniques is 9.3 minutes, which is relatively large.

Even though both medians are nearly the same (30.5 for AHP and 30.0 for Bubblesort); the difference in time consumption can easily be noticed when taking standard deviation into account. In our experiment, the distribution of the actual time consumption is not skewed and there is no extreme outlier as well. Hence, we can conclude that the Bubblesort tool is more time saving compared to the AHP tool.

Aside from measuring time consumption in minutes, we use the time consumption score, which is a subjective judgment given by the subjects after they use both techniques. The higher the score, the less time consuming the tool is. As shown in Table 2, Bubblesort achieves a higher score, which means that the subjects believe Bubblesort is less time consuming. The mean score that Bubblesort obtains from the subjects is 6.7, which is higher than the 5.7 of AHP, and the median of the scores for these two techniques are 6.5 and 5, respectively. Although, according to the standard deviation, the scores obtained by Bubblesort are more dispersed than AHP, this does not change the fact that there still is an indication Bubblesort is a better technique with respect to time consumption.

**Table 2.** Time consumption in minutes and grade per tool.

| Method | Median | Mean | Std. deviation(SD) |
|---|---|---|---|
| AHP (in minutes) | 30.5 | 39.5 | 19.5 |
| Bubblesort (in minutes) | 30 | 30.2 | 12.9 |
| AHP (in grade) | 5 | 5.7 | 1.9 |
| Bubblesort (in grade) | 6.5 | 6.7 | 2.2 |

### 4.4 Overall Rating

After taking reliability, ease of use, and time consumption into account, we also asked the subjects to assign an overall score to both techniques. This was again based on a scale of 1-10; the higher the score, the higher the appreciation of the technique.

If we look at the overall scores in Table 1, we notice a higher score for Bubblesort in comparison to AHP. This is coherent with the other scores obtained for reliability, ease of use, and time consumption, because those also indicated better averages for Bubblesort. The average score for AHP is 6.33, while Bubblesort shows an average score of 7.17. Furthermore, subjects were asked to explain their overall score and indicated that the combination of the quickness and the reliability of Bubblesort, in comparison to AHP, were the main reason for assigning higher scores to the former.

Analyzing the overall scores for both techniques within group A and group B, based on the Mann-Whitney test, did not show any significant differences ($\alpha = .05$) when comparing for Bubblesort but there was a minimal significant deviation between group A and B for AHP. This is considered to be quite remarkable since the overall score should be derived from the reliability, ease of use, and time consumption. Still, merging of both groups was difficult and a general comparison would be less reliable. Nevertheless, overall comparison, although not encouraged here, showed no significant deviation between Bubblesort and AHP.

# 5 Conclusion

Throughout this research, we attempted to provide an answer to the following research question:

*How do the requirement prioritization techniques Bubblesort and Analytical Hierarchy Process perform when compared to each other in terms of perceived reliability, time consumption, and usability?*

For this research we conducted an experiment among twelve Master students of a large University in the Netherlands to test the applicability of Bubblesort and AHP. The subjects used both techniques in prioritizing twenty fictional requirements for a new software product release of Google Maps. The perceived reliability, ease of use and time consumption was assessed by comparing both techniques. Taking both groups into account, there were some distinct differences found.

On average, Bubblesort scored better than AHP in all aspects of the comparison. The perceived reliability, the time consumption and the usability of Bubblesort was greater than AHP. Nevertheless, the division of both groups did have one minor implication on the results: the overall valuations of the tools, assigned by the subjects, seemed to be influenced by the group division.

During the experiment we tried to eliminate any advantages gained from using one of the tools before the other. When looking at the results for both groups, we saw no further significant differences between both groups ($\alpha = .05$). Therefore, we can conclude that both groups scored the same on both techniques.

After merging both groups to create an overall average, we can conclude that Bubblesort still scores better than AHP for perceived reliability of the results. Furthermore, Bubblesort also scored better on time consumption and ease of use by a decent margin. Unfortunately, the sample size proved to be an obstacle in obtaining significant results.

Concluding from our research, Bubblesort scores overall better compared to AHP. There is a strong indication that when the sample group size will be increased, Bubblesort would probably still outpace AHP when both techniques are applied in a similar situation.

# 6 Discussion, Limitations, and Further Research

Throughout our experiment we measured the perceived reliability, ease of use, and time consumption for both requirement prioritization techniques AHP and Bubblesort. However, the research presented in this paper has certain limitations. These limitations predominantly originate from the size of our research sample and the unavailability of established tools for both techniques.

Firstly, due to a lack of availability of subjects as well as time constraints, we were not able to increase the size of the sample. The time provided to assemble a representative large group of subjects proved to be limited. Therefore, the reliability of the obtained results can be influenced. Upon further review, the small sample size was not considered to be a normal distribution and non-parametric tests had to be performed. However, the results seem to give a decent indication of expected results when the sample size is increased. Therefore, we would suggest further research with an increased sample size.

Secondly, the unavailability of a usable and implementable tool for both techniques could have influenced the results. Currently, there is no established tool for Bubblesort available. Therefore we developed a custom Excel-based tool for Bubblesort fitting the number of requirements used within our experiment. Furthermore, we tried to use the IBM© focal point™ tool for AHP but encountered some problems with implementing it. Therefore, we developed a custom Excel-based tool for AHP too. Although this eliminates any problems on comparing the two techniques, there is a possibility that the tools were not comprehensive and sufficient enough due to a lack of validation of the tools. Since the participating subjects did not experience any difficulties nor did we hear any complaints, we believe the tools were sufficient enough to obtain reliable results on perceived reliability. However the results of this experiment on time consumption and ease of use will be very tool dependent and therefore specific for our custom tools.

A final note should be made on the issue of prioritizing large volumes of requirements. In this experiment, a set of twenty requirements was used. For software vendors dealing with small amounts of requirements, Bubblesort may be a good solution to prioritize requirement and reach consensus among the stakeholders. However, in many real-life settings, the amount of requirements easily exceeds the amount of requirements we used. Further research should be done to tool support in which Bubblesort is integrated with proposed solutions as structuring the requirements in a hierarchy.

## References


1. Bebensee, T., Weerd, I. vd., Brinkkemper, S.: Binary Priority List for Prioritizing Software Requirements. In: Requirements Engineering: Foundation for Software Quality. LNCS, vol. 6182/2010, pp. 67--78. Springer Heidelberg (2010)
2. Wiegers, K.: First Things First: Prioritizing Requirements. Software Development 7(9), (1999)
3. Kabbedijk, J., Brinkkemper, S., Jansen, S.: Customer Involvement in Requirements Management: Lessons from Mass Market Software Development. In: 17th IEEE International Requirements Engineering Conference, pp. 281--286. Atlanta (2009)
4. Lehtola, L., Kauppinen, M., Kujala, S.: Requirements Prioritization Challenges in Practice. In: Proceedings of 5th International Conference on Product Focused Software Process Improvement, pp. 497-508. Japan (2004)
5. Lehtola, L., Kauppinen, M.: Empirical evaluation of two requirements prioritization methods in Product Development Projects. In: Proc. European Software Process Improvement Conference, pp. 161-170. Trondheim, Norway (2004).
6. Bekkers, W., Weerd, I. van de, Spruit, M., Brinkkemper, S.: A framework for process improvement in software product management. In: A.Riel et al. (Eds.): EuroSPI 2010, CCIS 99, pp. 1--12 (2010)
7. Weerd, I. van de, Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., & Bijlsma, L.: Towards a reference framework for software product management. Proceedings of the 14th International Requirements Engineering Conference, Minneapolis/St. Paul, Minnesota, USA, 319-322 (2006)

8. Karlsson, J., Ryan, K.: A cost-value approach for prioritizing requirements. IEEE Software 14(5) 67--74 (1997)
9. Saaty, T. L.: The Analytical Hierarchy Process, McGraw-Hill (1980)
10. Berander, P., Jönsson, P.: Hierarchical Cumulative Voting (HCV) - prioritization of requirements in Hierarchies. International Journal of Software Engineering and Knowledge Engineering 16(6), 819--849 (2006)
11. Karlsson, L, Thelin, T., Regnell, B., Berander, P., Wohlin, C.: Pair-wise comparisons versus planning game partitioning – experiments on requirements prioritisation techniques. Empirical Software Engineering 12(1), 3--33 (2007)
12. Karlsson, J.: Software requirements prioritizing. In: Proceedings of 2nd International Conference on Requirements Engineering, pp. 110–116 (1996)
13. Perini, A., Ricca, F., Susi, A.: Tool-supported Requirements Prioritization: Comparing the AHP and CBRank Methods. Information and Software Technology 51(6), 1021--1032 (2009)
14. Racheva, Z., Daneva, M., Buglione, L.: Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products. In: Proceedings of the Second International Workshop on Software Product Management 2008, Barcelona, pp. 49--58 (2008)
15. Berander, P., Andrews, A.: Requirements prioritization. In: A. Aurum, C. Wohlin (Eds.), Engineering and Managing Software Requirements, Springer, (2005)
16. Karlsson, J., Wohlin, C., Regnell, B.: An evaluation of methods for prioritizing software requirements. Information and Software Technology 39(14--15), 939--947 (1998)
17. Aho, A. V., Hopcroft, J. E., Ullman, J. D.: Data structures and Algorithms. Addison-Wesley, Massachusetts (1983)
18. Avesani, P., Bazzanella, C., Perini A., Susi, A.: Facing scalability issues in requirements prioritization with machine learning techniques, In: Proceedings of 13th IEEE International Conference on Requirements Engineering, IEEE Computer Society, Paris, France, pp. 297–306 (2005)
19. Firesmith, D.: Prioritizing Requirements. In: Journal of Object Technology 3(8), pp. 35--47 (2004)
20. Random RI for AHP, http://www.docin.com/p-23136057.html
21. Berander, P.: Using Students as Subjects in Requirements Prioritization. In: International Symposium on Empirical Software Engineering (ISESE'04), pp. 167--176 (2004)