Towards Value-Based requirements prioritization for software product management

Samer I. Mohamed[(1),] Islam A. M. ElMaddah [(2),], and Ayman M. Wahba[(3)]

(1) Department of Computer and Systems.Ain Shams University (Egypt)
E-mail: samer.mohamed@eds.Com

(2) Department of Computer and Systems.Ain Shams University (Egypt)
E-mail: islam_elmaddah@yahoo.co.uk

(3) Department of Computer and Systems.Ain Shams University (Egypt)
E-mail: ayman.wahba@gmail.Com

ABSTRACT

Meeting stakeholders' requirements and expectations becomes one of the critical aspects on which any software organization in market-driven environment focuses on, and pays a lot of effort and expenses to maximize the satisfaction of their stakeholders. Therefore identifying the software product release contents becomes one of the critical decisions for software product success. Requirements prioritization refers to that activity through which product releases contents that maximize stakeholder satisfaction can be identified [8]. This paper illustrates the Value-Oriented requirement prioritization approach for software product management. The technique proposed in this paper is based on the Hierarchical Cumulative Voting (HCV) and Value-Oriented Prioritization (VOP) techniques. The proposed technique, Value-Oriented HCV (VOHCV) addresses the weakness of HCV through selecting the best candidate requirements for each release not only based on the stakeholder's perceived value as HCV but also in terms of associated anticipated cost, technical risk, relative impact and market-related aspects. The VOHCV also addresses the weakness of VOP through supporting not only requirements flat structure as VOP but also through supporting hierarchical structure. By this means VOHCV inherits the strengths of both VOP and HCV and addresses their weaknesses while selecting the best candidate release requirement, to maximize stakeholders' value and satisfaction [11].

Keywords: Requirements prioritization, Value-Oriented prioritization, Hierarchical Cumulative Voting prioritization.

1- INTRODUCTION

Due to the continuous increase in the number of software requirements for market-driven products, there is an increasing need for methods capable of prioritizing candidate requirements since not all requirements can usually be met with available time and resource constraints in one software release [9]. Thus

many organizations believe that it is not only important to enable their customers to assign priorities to requirements and to make decisions about them but also to provide them with different alternative solutions tailored for their own needs [7]. By this way they will provide more value for their customers through selecting the most valuable requirements to be implemented in each one of the product releases [31].

Managing requirements for any software product becomes a key factor that identifies not only the project success or failure but also the organization destiny. The critical portion of this process is to identify those requirements that balance the stakeholders' needs, customer expectations, business values, total cost and schedule [8]. Therefore requirements prioritization and selection processes that maximize the stakeholder value have a great impact on the product success [17].

Value-Oriented Prioritization (VOP) refers to that process which evaluates the requirements from different stakeholders based on the impact on specific business core values for both the organization and the stakeholders themselves [29] since focusing on value provides the opportunity to create a strategy to achieve long-term profitable growth and sustainable competitive advantage [6]. VOP also supports the stakeholders with a visible mechanism during decision-making to be able to provide their values and weights for each requirement. Using the quantitative and visible approach of VOP, it becomes much easier for the stakeholders to emphasize the business values [16].

The organization of the paper is as follows. In the next section, we will refer to the related work for our research. In section three we will elaborate the rationale for the VOHCV algorithm, and the research methodology we have applied to develop it. In section four, we will discuss the VOHCV algorithm specifications. Section five will illustrate the practical advantages from the VOHCV algorithm through a case study. In section six, we will validate the practical benefits from VOHCV algorithm through a comparison between HCV and VOHCV algorithms. The final section summarizes our conclusions and introduces our future research.

2- RELATED WORK

2-1 Measurement scale

Measurements systems are classified into four different scale types ordered based on richness: Nominal, Ordinal, Interval and Ratio. Ratio-scale type is said to be richer than the ordinal-scale type as all the relations in the ratio-scale are contained in the ordinal-scale [4]. Thus the scale type is used to determine which arithmetic operations are allowed and which kind of analysis can be performed based on these operations. For the requirements prioritization, both ordinal-scale and ratio-scale types are the most common techniques. Ordinal-scale preserves the order of the requirements according to the value assigned to each requirement, which indicates that no arithmetic operations are allowed. Ratio-scale on the other hand supports all arithmetic operations. Thus ratio-scale provides not only the order of the requirements but also the relative distance between the dif-

ferent requirements. Ratio-scale can also identify by how much one requirement is more important than another [5]. When using a prioritization technique that provides relative priorities on a ratio-scale, it is possible to calculate the total importance of a set of requirements by adding together their priorities. One of the most interesting advantages of the ratio-scale is that it allows sophisticated calculations for preparing different candidate solutions to base decisions on.

2-2 Requirements hierarchy

Requirements exist naturally on different levels of abstractions. This will make the process of requirements prioritization more difficult especially when the requirements exist in different abstraction levels [32]. This difficulty arises because requirements in different levels of abstractions have different importance since lower-level requirements (leafs) were considered less-important then high-level requirements (goals). Thus only those requirements exist on the same level of hierarchy should be compared while performing prioritization [33]. For multi-level prioritization, lower-level requirements can either inherit the priorities from high-level requirements, or be assigned these priorities directly. The former case is used when the lower-level requirements have AND-relationship between themselves, while the latter is used when the lower-level requirements have OR-relationship between themselves.

2-3 Hierarchical Cumulative Voting prioritization

Hierarchical Cumulative Voting (HCV) prioritization is a ratio-scale prioritization technique. HCV was designed to overcome the drawbacks for Analytical Hierarchy Protocol (AHP) [28] and Cumulative Voting (CV) [35] techniques, and to inherit the advantages and good features of both techniques [4]. By other means HCV is taken to be an extension for the CV technique by supporting hierarchy. This feature enables HCV to solve multi-aspect decision problems like AHP. Having HCV provides relative priorities based on a ratio scale, gives it the opportunity to calculate the total importance of a set of requirements by adding together their priorities. It also helps combine the different aspects and calculate rations in between these aspects. For example, you can calculate the cost-value ration for the requirement that represents how much value each requirement adds relative to the implementation anticipated cost [18]. The strength of HCV will increase when the number of requirements grows because the need for a structural/hierarchical approach gets larger. HCV provides this structure through using natural relationships between the requirements to perform the prioritization in a series of steps and hence minimize requirements prioritized at a time.

The main concept behind HCV is to quantify the requirement importance by distributing points between the requirements to reflect this importance. However, when prioritizing with HCV, not all requirements are prioritized at the same time. Instead, prioritizations are performed at different levels of the hierarchy, and within different blocks of the requirements in the hierarchy as (shown in figure 1). As illustrated in figure 1, the requirements are distributed over two lev-

els of the hierarchy: high-level requirements (HLR) and low-level requirements (LLR). The relationship between the lower-level requirements are OR relationship among themselves. Only those requirements within the same block (grey area in the figure) are prioritized at the same time. This will make the prioritization process smoother and the risk of neglecting any requirement will decrease.

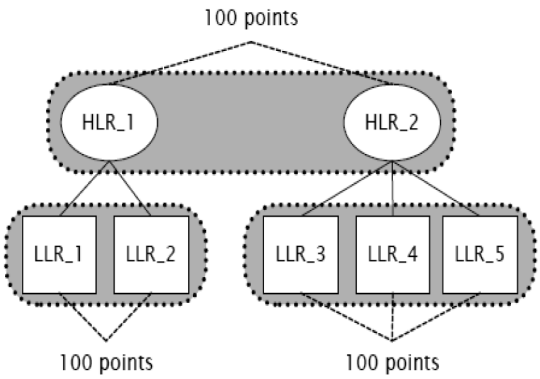

Figure 1 HCV requirements hierarchy

2-4 Value-Oriented Prioritization

Value-Oriented Prioritization (VOP) is proved to be the process which aligns product demands with company goals and stakeholders expectations through providing a visible and defined process for prioritizing and managing requirements over the product life cycle [28]. It helps out the stakeholder to view the whole picture for the sake of the organization targets and vision, rather than arguing over which product requirements to implement [28].

The whole idea behind VOP is to focus on the core business values that lead to stakeholder's satisfaction while prioritizing the product requirements as indicated by Karl Wiegers [30]. Examples of these core business values are the customer value gained from implementing the requirement, the implementation cost, risk associated with implementing this requirement, impact that will occur if this requirement is not implemented and other market-related aspects that will be affected if this requirement is not implemented [24]. Requirement attractiveness is proportional to the value it provides and inversely proportional to its cost, associated risk, impact and market aspects. Each business value is given a weight based on the organization objectives and vision. Each stakeholder puts his estimate against each business value for each requirement. All these input values are consolidated together while generating the requirement rank [3].

3- RATIONALE AND RESEARCH TECHNIQUE

3-1 Rationale

The rationale behind the VOHCV is to combine both HCV and VOP techniques to gain the advantages of both. VOHCV will not only take value as in HCV or cost and risk as in VOP into account while prioritizing the requirements [18], but it will also take into account the other business values like relative implementation impact and market-related aspects. This will yield to higher quality results because it takes the different features that affect the requirement throughout the product life cycle into account while producing whole release ranks [10].

3-2 Research technique

The research methodology we followed for the conception of this technique is based on the incremental software delivery approach and described as follows:

- Literature review for the current and practical challenges for the software product management industry from both business and strategic perspectives. The outcomes from this review point out the importance of the prioritization process in handling these challenges [16, 19, 20, 22, and 23].
- Literature review for the prioritization techniques that helps achieve the software product management challenges. The outcomes from this review pointed out that both HCV and VOP techniques are the best candidates which focus the value gained while prioritizing the product requirements [1, 12, 15, 17, and 34].
- Identifying the pros and cons for those prioritization techniques [13, 14, 15, and 21].
- A prototype implementation for the VOHCV technique based on the knowledge gained from the previous points.
- Designing a framework with the core engine based on the proposed prioritization technique to facilitate testing and evaluating the effectiveness of VOHCV.
- Using the designed framework to address a group of the open issues with HCV to help maintaining the robustness of VOHCV [3].

4- VOHCV ALGORITHM DETAILS

To handle the requirements prioritization using VOHCV, there is a series of steps needed to be followed as follows:

- Step 1: Assign the core business values global weights. The supported business values for each requirement in VOHCV are, the anticipated implementation cost, associated implementation risk, perceived customer value, relative impact and market-related aspects. These weights are assigned based on the organization strategic goals and future vision. These weights will range from 1 to 10 (1 reflects lowest importance and 10 re-

flects highest importance).

- Step 2: Assign the weights for each business value features. These weights will reflect how much each feature is important to the stakeholders and controlled by the organization objectives. These weights will be common to the entire stakeholder sharing in the requirements prioritization process. These weights will range from 1 to 10 (1 reflects lowest importance and 10 reflects highest importance).
- Step 3: Each stakeholder will enter his point of view for each business value feature in terms of feature value. This value will reflect how this feature will affect the requirement from his own point of view. All business values mentioned before have different features except (Value) business value which has only one feature. These values will range from 1 to 10 (1 reflects lowest importance and 10 reflects highest importance).
- Step 4: Calculate the requirement distribution points assigned to each requirement based on the above feature weights and values. This should be done by each stakeholder sharing in the requirements prioritization process. To show how this distribution points are calculated, let us assume the following parameters :

1- Wc: Weight for the global (Cost) business value.
2- Wv: Weight for the global (Value) business value.
3- Wr: Weight for the global (Risk) business value.
4- Wi: Weight for the global Impact business value.
5- Wa: Weight for the global (Aspect) business value.
6- Wi,j: Weight assigned to requirement (Ri) with respect to business value feature (Fj).
7- Vi,j: Value assigned to requirement (Ri) with respect to business value feature (Fj).
8- Nc: Count of features per (Cost) business value.
9- Nr: Count of features per (Risk) business value.
10- Ni: Count of features per (Impact) business value.
11- Na: Count of features per (Aspect) business value.
12- Nb: Count of business values that affect requirement.
13- CBVavr: Average value for the (Cost) business value affect requirement.
14- RBVavr: Average value for the (Risk) business value affect requirement.
15- IBVavr: Average value for the (Impact) business value affect requirement.
16- ABVavr: Average value for the (Aspect) business value affect requirement.
17- TNdist: Total distribution number for the requirement.
18- ANdist: Average distribution number for the requirement.
19- Cf: Compensation factor to control the range of the distribution number. It will be set to 10 to have the distribution number range between 1 and 100 similar to ordinary HCV.
20- R_Pt : Number of points assigned to each requirement.

The process of calculating the average distribution number for each requirement can be shown as follows:

1-Calculate the average business value for each requirement.

- Average value for the (Cost) business value

$$CBVavr = (\sum_{Nc} WijXVij\ ) / Nc \quad (1)$$

- Average value for the (Risk) business value

$$RBVavr = (\sum_{Nr} WijXVij\ ) / Nr \quad (2)$$

- Average value for the (Impact) business value

$$IBVavr = (\sum_{Ni} WijXVij\ ) / Ni \quad (3)$$

- Average value for the (Aspect) business value

$$ABVavr = (\sum_{Na} WijXVij\ ) / Na \quad (4)$$

2-Calculate the total distribution number for each requirement.

$$TNdist = WcX\ CBVavr + WrX\ RBVavr + WiX\ IBVavr + WaX\ ABVavr + WvXVij \quad (5)$$

3-Calculate the Average distribution number assigned to each requirement. We will refer to this number later as the assigned priority.

$$ANdist = TNdist / (NbXCf) \quad (6)$$

4-Calibrate the distribution number between the different LLRs of the same HLRs to have the sum of all the LLRs points equals 100 as indicated by CV technique [4] . This can be calculated by using the relation between the LLRs distribution numbers and the below equation.

$$\sum_{HLR/LLR} R\_Pt = 100 \quad (7)$$

- Step 5: Calculate the intermediate priorities for the requirements either through the straight or compensated calculation. To show how the intermediate priority is calculated, let us assume the following parameters:
  1- Pi,LLR_u : Intermediate priority value for the Lower Level Requirement (LLR) (u).
  2- Pa,LLR_u : Assigned priority value for the Lower Level Requirement (LLR) (u) calculated from the previous step.
  3- Pa,HLR_v: Assigned priority value for the Higher Level Requirement (HLR) (v), or the parent of LLR_u .
  4- CHLR_v: Block specific compensation factor, this could be the number of requirements within the prioritization block.

$$Pi,LLR\_u = CHLR\_v\ X\ Pa,LLR\_u\ X\ Pa,HLR\_v \quad (8)$$

- Step 6: Calculate the final priorities for the requirements at the level of interest. The calculation is performed across the blocks within the same level. This indicates that all requirements located at this specific level will be prioritized relative to each other. To show how the final priority has been calculated, let us assume the following parameters:

  1- Pf,LLR_u : Final priority value for the Lower Level Requirement (LLR) (u).
  2- Pi,LLR_k : Intermediate priority value for all the Lower Level Requirement (LLR) (k) of the (HLR_v).

$$Pf,LLR\_u = Pi,LLR\_u / \sum_{Pk} Pi,LLR\_k \quad (9)$$

- Step 7: Calculate the final priorities based on the consolidated stakeholders weighted priorities calculated from the previous steps. To show how the final priority is calculated, let us assume the following parameters:
  1- Pmf,LLR_u : Final priority value for all stakeholders of the Lower Level Requirement (LLR) (u).
  2- Pf,LLR_u,S_k : Final priority value for stakeholders (S_k) of the Lower Level Requirement (LLR) (u).
  3- Wk : Stakeholder normalized weight.

$$Pmf,LLR\_u = \sum_{k} Wk \; X \; Pf,LLR\_u,S\_k \quad (10)$$

- Step 8: Calculate the final ranks based on the final priority value assigned to each requirement.

5- CASE STUDY

In order to show the practical advantage from VOHCV, we will illustrate that through an example based on the requirement hierarchy structure (shown in figure 1). In this example, there are two abstraction levels and one stakeholder. Furthermore there are two high-level requirements (HLRs) and five low-level requirement (LLRs). Given the business value weights, features weights and features values for each requirement from table 1, we will be able to calculate the requirements by following the VOHCV algorithm steps mentioned in the previous section.
The first step in the VOHCV algorithm is to calculate the number of points assigned to each requirement of the same block, or by other means distribute 100 points over the requirements of the same block.

Table 1. Input values and weights

Business value (B.V.)	Cost		Risk		Impact	Aspect		Value
B.V. Weight	9		10		6	4		5
Feature type	C1	C2	R1	R2	I1	A1	A2	V1
F. Weight	5	8	10	6	3	7	8	10
HLR1(Value)	3	4	5	3	1	1	5	2
HLR2(Value)	9	6	6	8	9	10	3	7
LLR1(Value)	1	10	10	7	2	5	9	4
LLR2(Value)	8	10	9	10	10	8	9	10
LLR3(Value)	3	5	6	5	4	3	2	10
LLR4(Value)	1	2	2	4	3	2	4	3
LLR5(Value)	9	8	10	7	9	10	9	10

This can be done by applying equations 1 through 7, given the values and weights of table 1. These points are illustrated in the first two columns in table 2. After all the requirements in the prioritization blocks have been assigned priorities, the next step is to calculate the intermediate LLR priority using equation 8, given that the compensation factor is equivalent to the block size as illustrated in the third column of table 2. The intermediate LLR priority is illustrated in the fourth column of table 2. The next step is calculating the final normalized LLR priority using equation 9. The final LLR priority illustrated in the fifth column of table 2. The last step is to rank the LLRs based on the final LLR priority. The LLR ranks priority is illustrated in the sixth column of table 2.

Table 2. VOHCV Requirements output ranks

HLR/LLR	HLR points	LLR point	Compansa-tion factor	Intermediate priority	Final priority	Rank
HLR1/LLR1	30	40	2	2400	9	5
HLR1/LLR2	30	60	2	3600	13	3
HLR2/LLR3	70	33	3	6930	26	2
HLR2/LLR4	70	15	3	3150	12	4
HLR2/LLR5	70	52	3	10920	40	1

As shown from the first two columns of table 2, HLR2 (70%) is more important than HLR1 (30%) and LLR5 is considered to be the most important LLR and accounts for (40%) of the importance of all the LLRs while LLR (9%) is considered to be the lowest important LLR over all the other LLRs.

6- EVALUATION OF VOHCV IN COMPARISON TO HCV

In order to show the strength of the new proposed technique (VOHCV) compared to the ordinary (HCV), an empirical evaluation should be conducted. The main drawback of HCV is that it takes only the "Value" perspective into account while prioritizing the requirements and neglecting the other business perspectives. On the other hand VOHCV fixes this by taking the other business perspectives into account through the prioritization process.
To show that, we will use the example introduced in the previous section and exclude all perspectives except "Value" perspective to gain the HCV ranking. After that a detailed comparison between the two techniques will be conducted based on the results.
In order to calculate the distribution points for both LLRs and HLRs for HCV based on the values mentioned in the ninth column of table 1, we will use the relation between these values that belongs to the same prioritizing block. For example both LLR1 and LLR2 belong to the same block and have value equal to 4 and 10 respectively. To distribute 100 points over these two LLRs with keeping the relation between the assigned values, we can conclude by a simple mathematical calculation that LLR1 can be assigned 71 points and LLR2 can be assigned 29 points. The same can be done for both LLR3,LLR4 and LLR5 as they are belong to the same block. The calculation for the later case will yield to LLR3 will be assigned 43 points, LLR4 will be assigned 14 points and LLR5 will be assigned 43 points. The prioritization calculation will be (shown in table 3) as follows:

Table 3. HCV Requirements output ranks

HLR/LLR	HLR points	LLR point	Comp. factor	Inter. priority	Final priority	Rank
HLR1/LLR1	22	71	2	3124	11	4
HLR1/LLR2	22	29	2	1276	4	5
HLR2/LLR3	78	43	3	10062	36	1
HLR2/LLR4	78	14	3	3276	13	3
HLR2/LLR5	78	43	3	10062	36	2

Comparing the results between HCV and VOHCV as illustrated from table 1 and 2, we conclude the followings:

- LLR5 is considered the most important of the LLRs and accounted for

(40%) of the importance of all the LLRs as result of applying VOHCV technique.
- LLR3 is considered as the most important of the LLRs and accounted for (36%) of the importance of all the LLRs as result applying HCV technique.
- LLR1 is considered as the least important of the LLRs and accounted for (9%) of the importance of all the LLRs as result of applying VOHCV technique.
- LLR2 is considered as the least important of the LLRs and accounted for (4%) of the importance of all the LLRs as result of applying HCV technique.
- Neglecting the effect of the other business perspectives rather than "Value" yields to a miss leading result. This can be shown by comparing the features values for both LLR5 (The most important of the LLRs from VOHCV) and LLR3 (The most important of the LLRs from HCV) from table 1. The values of LLR3 indicate that the stakeholder assigns to them small values to indicate low importance of this LLR compared to the others. While the values for LLR5 indicate that the stakeholder assigns to it a large value to indicate how important this LLR is compared to the others which are detected by VOHCV.
- The results quality from VOHCV is much higher than HCV and reflects the real business case. The real business case not only takes the effect of "Value" and neglects the other perspectives, but it takes all the different perspectives that affect the requirement throughout the product life cycle into account.
- Adopting VOHCV as a methodology for prioritizing the requirements yields to higher credibility with the release contents that results in customer satisfaction [25].

7- CONCLUSIONS AND FUTURE WORK

In this paper, we have presented the strengths and weaknesses of both Value-Oriented Prioritization (VOP) and Hierarchical Cumulative voting (HCV) techniques. From these strengths and weaknesses, we came up with the new prioritization technique which is Value-Oriented Hierarchical Cumulative Voting (VOHCV). VOHCV combines the strengths of both VOP and HCV to improve the quality of the requirements prioritization process. The main difference between the HCV and VOHCV is that VOHCV uses the value-oriented approach not only to get the effect of value but also to get the effect of different other core business values like cost, risk, aspect and impact [10]. By this way VOHCV enables the product manager to take all the aspects that affect the requirement into account while selecting the best candidate for product release [22].

We already designed a release management framework with VOHCV embedded to get benefit from the results produced by VOHCV in the release planning and software product management [29]. VOHCV acts as the main core engine for the designed framework. The results from the VOHCV taken as inputs for further release planning activities to support and enable product manager to easily control and manage the software product [26].

Since only a few studies have been performed to evaluate the efficiency and suitability of VOHCV, there is a need to do further studies for some issues that can affect the algorithm

efficiency [2, 12, and 21]. Examples of these open issues like how many points we should distribute over the requirements of the same block, how many hierarchy levels should be prioritized, how large priority blocks are possible to prioritize and also the effect of the requirements order on the requirements ranking, will be our objective for the next phase of our research.

REFERENCES

[1] Ahl, A. (2005): 'An Experimental Comparison of Five Prioritization Techniques - Investigating Ease of Use', Accuracy, and Scalability, Department of Systems and Software Engineering, Blekinge Institute of Technology, pp 101

[2] Avensani, P., Bazzanella, A., Perini, A., and Susi, A. (2005): 'Facing Scalability Issues in Requirements Prioritization with Machine Learning Techniques', Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE2005), Paris, France, pp. 297-305.

[3] Berander, P., Khan, K. A., and Lehtola, L. (2006): 'Towards a Research Framework on Requirements Prioritization', Proceedings of the Sixth Conference on Software Engineering Research and Practice in Sweden(SERPS'06), Umeå, Sweden, pp. 39-48.

[4] Berander, P., and Jönsson, P. (2006): 'Hierarchical Cumulative Voting (HCV) - Prioritization of Requirements in Hierarchies', accepted for publication in International Journal of Software Engineering and Knowledge Engineering (IJSEKE) - Special Issue on Requirements Engineering Decision Support, December 2006, World Scientific Publishing Company, Singapore, pp. 76-84.

[5] Berander, P. and Andrews, A. (2005): 'Requirements Prioritization', in Engineering and Managing Software Requirements, ed. Aurum, A. and Wohlin, C., Springer Verlag, Berlin, Germany, pp. 69-94.

[6] Berander, P. (2004): 'Prioritization of Stakeholder Needs in Software Engineering - Understanding and Evaluation', Blekinge Institute of Technology Licentiate Series No. 2004:12, Department of Systems and Software Engineering, Blekinge Institute of Technology.

[7] Boehm, B. and Huang, L. G. (2003): 'Value-Based Software Engineering: A Case Study', IEEE Computer, 36(3), pp. 22-41.

[8] Boehm, B. (2006): 'Value-Based Software Engineering: Overview and Agenda', in Value-Based Software Engineering ed. Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., and Grünbacher, P., Springer Verlag, Berlin, Germany, pp. 3-14.

[9] Carlshamre, P. (2002): 'Release Planning in Market-Driven Software Product Development: Provoking an Understanding', Requirements Engineering Journal, 7(3), pp. 139-151.

[10] Faulk R. , D. Harmon, and D. Raffo (2000): 'Value-Based Software Engineering (VBSE): A Value-Driven Approach to Product-Line Engineering', Proceedings, First International Conference on Software Product Line Engineering, pp. 205-223

[11] Hill, N., Brierly, J., and MacDougall, R. (1999): 'How to Measure Customer Satisfaction', Gower Publishing, Hampshire, England.

[12] Jönsson, P. and Wohlin, C. (2005): 'A Study on Prioritization of Impact Analysis: A Comparison Between Perspectives', Proceedings of the Fifth Conference on Software Engineering Research and Practice in Sweden (SERPS'05), Västerås, Sweden, pp. 11-19.

[13] Karlsson J., Wohlin C., and Regnell B. (1998): 'An evaluation of methods for prioritizing software requirements', Information and Software Technology, 39,pp. 939-947.
[14] Karlsson, L., Höst, M., and Regnell, B. (2006): 'Evaluating the Practical Use of Different Measurement Scales in Requirements Prioritization', Proceedings of the 5th ACM-IEEE International Symposium on Empirical Software Engineering, Rio de Janeiro, Brazil, pp. 326-335.
[15] Karlsson, L., Berander, P., Regnell, B., and Wohlin, C. (2004): 'Requirements Prioritization: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning', Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering (EASE 2004), Edinburgh, Scotland, pp. 145-154.
[16] Karlsson, L., Dahlstedt, A. G., Regnell, B., Natt och Dag, J., and Persson, A. (2007): 'Requirements Engineering Challenges in Market- Driven Software Development - An Interview Study With Practitioners', Accepted for publication in Information and Software Technology: Special Issue on Understanding the Social Side of Software Engineering, pp. 588-604.
[17] Karlsson, J. (1996): 'Software Requirements Prioritizing', Proceedings of the Second International Conference on Requirements Engineering (ICRE'96), Colorado Springs, CO, pp. 110-116.
[18] Karlsson J., Ryan K., (1993): 'A Cost-Value Approach for Prioritizing Requirements', IEEE Software, 1997, vol. 14, No. 5, pp 67 – 75
[19] Lehtola, L. and Kauppinen, M. (2006): 'Suitability of Requirements Prioritization Methods for Market-Driven Software Product Development', Software Process Improvement and Practice (SPIP), 11(1), pp. 7-19.
[20] Lehtola, L., Kauppinen, M., and Kujala, S. (2005): 'Linking the Business View to Requirements Engineering: Long-Term Product Planning by Roadmapping', Proceedings of the, 13th International Conference on Requirements Engineering, Paris, France, pp. 439-443.
[21] Lehtola, L. and Kauppinen, M. (2004): 'Empirical Evaluation of Two Requirements Prioritization Methods in Product Development Projects', Proceedings of the European Software Process Improvement Conference (EuroSPI 2004), Trondheim, Norway, pp. 161-170.
[22] Ngo-The, A. and Ruhe, G. (2005): 'Decision Support in Requirements Engineering', in Engineering and Managing Software Requirements, ed. Aurum, A. and Wohlin, C., Springer Verlag, Berlin, Germany, pp. 267-286.
[23] Nicholas, J. M. (2001): 'Project Management for Business and Technology - Principles and Practice', 2nd Edition, Prentice Hall, Upper Saddle River, NJ.
[24] Nurmuliani, N., Zowghi, D., and Powell, S. (2004): 'Analysis of Requirements Volatility during Software Development Lifecycle', Proceedings of the 2004 Australian Software Engineering Conference (ASWEC '04), Melbourne, Australia, pp. 28-37.
[25] Moisiadis F. (2002): 'The fundamentals of prioritizing requirements', Proceedings of the Systems Engineering, Test and Evaluation Conference, Sydney, Australia, pp 108–119
[26] Robertson, S. and Robertson, J. (1999): 'Mastering the Requirements Process', Addison-Wesley, Harlow, England.
[27] Smith R. , Azar J. , and Cordes D. (2007): 'A Value-Oriented Approach to Requirements Prioritization', tech. report, Dept. of Computer Science, Univ. of Ala-

bama., vol. 24, no. 1, pp. 32-37.
[28] T. L. Saaty, The Analytic Hierarchy Process (McGraw-Hill, NY, 1980).
[29] Wiegers K. (1999): 'First Things First: Prioritizing Requirements,' Software Development, vol. 7, no. 9, www.processimpact.com/pubs.shtml#requirements.
[30] Wiegers K., Software Requirements Second Edition, Microsoft Press, (2003).
[31] Wohlin, C. and Aurum, A. (2006): 'Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study', in Value-based Software Engineering, ed. Biffl, S., Aurum, A., Boehm, B., Erdogan, H., and Grünbacher, P., Springer Verlag, Berlin, Germany, pp. 179-200.
[32] Wohlin C. and Berander P. (2003): 'Identification of key factors in software process management', A case study, in Proc. Int. Symp. on Empirical Software Engineering (ISESE'03), Rome, Italy, pp. 316-325.
[33] Wohlin C. and Jonsson P. (2005): 'Understanding impact analysis: An empirical study to capture knowledge on different organizational levels', in Proc. 17th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE'05), Taipei, Taiwan, pp. 707-712.
[34] Wohlin C., Karlsson L., Berander P. and Regnell B. (2004): 'Requirements prioritization: An experiment on exhaustive pairwise comparisons versus planning game partitioning' in Proc. 8th Int. Conf. on Empirical Assessment in Software Engineering (EASE 2004), Edinburgh, Scotland, pp. 145-154.