Criteria-Based Requirements Prioritization for Software Product Management

Samer I. Mohamed
Ain Shams University, Egypt
Department of computer and system engineering
+20102221982
samer.mohamed@eds.com

Islam A. El-Maddah
Ain Shams University, Egypt
Department of computer and system engineering
+20127820019
islam_elmaddah@yahoo.co.uk

Ayman M. Wahba
Ain Shams University, Egypt
Department of computer and system engineering
+20127912256
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 focus on, and pays a lot of efforts 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. This makes it one of the most important component of software requirement decision support in incremental software development [8]. This paper illustrates the Criteria-Based requirement prioritization approach for software product management. The technique proposed on this paper designed based on the Hierarchical Cumulative Voting (HCV) and Value-Oriented Prioritization (VOP) techniques. The proposed technique, Value-Oriented HCV (VOHCV) can select the best candidate requirements for each release based on the stakeholders input values for each requirement. These values reflect the importance of each requirement in terms of associated anticipated cost, technical risk, relative impact, market-related aspects and perceived value to the stakeholder. The VOHCV inherits the strengths of the HCV and VOP techniques. It also provides a mechanism that takes different stakeholders’ aspects into account while selecting the best candidate release requirement, to maximize stakeholders’ value and satisfaction [11].

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 identify 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 [18].

Value-Oriented Prioritization (VOP) is a terminology 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 to the stakeholders themselves [28]. Since focusing on value provide 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 core business values [17].

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 on 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. In 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 will summarize our conclusions and introduces our future research.

2. Related work

2.1. Hierarchical Cumulative Voting prioritization

Hierarchical Cumulative Voting (HCV) prioritization technique was designed to overcome the drawbacks for Analytical Hierarchy Protocol (AHP) and Cumulative Voting (CV) 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 ration scale, gives it the opportunity to calculate the total importance of a set of requirements by adding together their priorities. It also helps to 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 [19].

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 levels of the hierarchy; high-level requirements (HLR) and low-level requirements (LLR). Only those requirements within the same block (grey area in the figure) are prioritized at the same time. This will make the prioritization process much easier and the risk of neglecting any requirement will decrease [5].

2.2. Value Oriented Prioritization

Value-Oriented Prioritization (VOP) proved to be the process which align 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 [29].

The whole idea behind VOP is to focus on the core business values that leads to stakeholders 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 [25]. A requirement attractiveness is proportional to the value it provides and inversely proportional to it’s cost, associated risk, impact and market aspects. Each business value is given a weight based on the organization objectives and vision. Each stakeholder put 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 and cost as business values into account while prioritizing the requirements [19], but it also will take into account the other business values like associated risk, relative implementation impact and market-related aspects. This will yield to a 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 methodology

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 to the importance of the prioritization process in handling these challenges [17,20,21,23,24].
- Literature review for the prioritization techniques that helps to 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 [1,12,16,18].
- Identifying the pros and cons for those prioritization techniques [13,14,15,22].
- A prototype implementation for the VOHCV technique based on the knowledge gained from the previous points.
- Design a framework with the core engine based on the proposed prioritization technique to facilitate testing and evaluating the effectiveness of VOHCV.
- Use 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 are a series of steps needs 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 such as 1 reflects lowest importance and 10 reflects highest importance.
- Step 2: Assign the weights for each business value features. These weights will reflect how much each feature important to the stakeholders and controlled by the organization objectives. These weights will be common to all the stakeholder sharing in the requirements prioritization process. These weights will range from 1 to 10 such as 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 such as 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 calculated, let us assume the following parameters:

1- $W_c$: Weight for the global (Cost) business value.
2- $W_v$: Weight for the global (Value) business value.
3- $W_r$: Weight for the global (Risk) business value.
4- $W_i$: Weight for the global (Impact) business value.
5- $W_a$: Weight for the global (Aspect) business value.
6- $W_{i,j}$: Weight assigned to requirement ($R_i$) with respect to business value feature ($F_j$).
7- $V_{i,j}$: Value assigned to requirement ($R_i$) with respect to business value feature ($F_j$).
8- $N_c$: Count of features per (Cost) business value.
9- $N_r$: Count of features per (Risk) business value.
10- $N_i$: Count of features per (Impact) business value.
11- $N_a$: Count of features per (Aspect) business value.
12- $C_f$: 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
     \[ CBV_{avr} = \left( \sum \sum \sum \sum \right) W_{ij} X V_{ij} / N_C \] (1)
   - Average value for the (Risk) business value
     \[ RBV_{avr} = \left( \sum \sum \sum \sum \right) W_{ij} X V_{ij} / N_R \] (2)
   - Average value for the (Impact) business value
     \[ IBV_{avr} = \left( \sum \sum \sum \sum \right) W_{ij} X V_{ij} / N_I \] (3)
   - Average value for the (Aspect) business value
     \[ ABV_{avr} = \left( \sum \sum \sum \sum \right) W_{ij} X V_{ij} / N_A \] (4)

2-Calculate the total distribution number for each requirement.
   \[ TN_{dist} = W_c \times CBV_{avr} + W_r \times RBV_{avr} + W_i \times IBV_{avr} + W_a \times ABV_{avr} + W_v \times V_{ij} \] (5)

3-Calculate the Average distribution number assigned to each requirement. We will refer to this number later as the assigned priority.
   \[ AN_{dist} = TN_{dist} / (N_b \times C_f) \] (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. This can be calculated by using the relation between the LLRs distribution numbers and the below equation.
   \[ \sum R_{Pt} = 100 \] (7)
   \[ \text{HLR/LLR} \]

- Step 5: Calculate the intermediate priorities for the requirements either through the straight or compensated calculation. To show how the intermediate priority been 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.

\[ Pi,LLR_u = CHLR_v \times Pa,LLR_u \times Pa,HLR_v \] (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 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 \( HLR_v \).

\[ Pf,LLR_u = Pi,LLR_u / \sum \sum Pi,LLR_k \] (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 been 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- \( Pi,LLR_u,S_k \) : Final priority value for stakeholders \( S_k \) of the Lower Level Requirement (LLR) (u).
  3- \( W_k \) : Stakeholder normalized weight.

\[ Pmf,LLR_u = \sum \sum W_k \times Pf,LLR_u,S_k \] (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.

Table 1. Input values and weights

<table>
<thead>
<tr>
<th>Business value</th>
<th>Cost</th>
<th>Risk</th>
<th>Impact</th>
<th>Aspect</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.V. Weight</td>
<td>9</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Feature type</td>
<td>C1 C2 R1 R2 H1 A1 A2 V1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F. Weight</td>
<td>5</td>
<td>8</td>
<td>10</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>HLR1(Value)</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>HLR2(Value)</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>LLR1(Value)</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>LLR2(Value)</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>LLR3(Value)</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>LLR4(Value)</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>LLR5(Value)</td>
<td>9</td>
<td>8</td>
<td>10</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

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. This can be done by applying equations 1 through 7, given the values and weights of table 1. These points 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 illustrated in the fourth column of table 2. The next step is calculate 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 illustrated in the sixth column of table 2.

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 account for (40%) of the importance of all the LLRs while LLR (9%) is consider 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 fix this by taking the other business perspectives into account through the prioritization process.

To show that, we will used 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 are 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 concluded by 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 2. VOHCV Requirements output ranks

<table>
<thead>
<tr>
<th>HLR/LLR</th>
<th>HLR points</th>
<th>LLR point</th>
<th>Comp. factor</th>
<th>Inter. priority</th>
<th>Final priority</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLR1/LLR1</td>
<td>30</td>
<td>40</td>
<td>2</td>
<td>2400</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>HLR1/LLR2</td>
<td>30</td>
<td>60</td>
<td>2</td>
<td>3600</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>HLR2/LLR3</td>
<td>70</td>
<td>33</td>
<td>3</td>
<td>6930</td>
<td>26</td>
<td>2</td>
</tr>
<tr>
<td>HLR2/LLR4</td>
<td>70</td>
<td>15</td>
<td>3</td>
<td>3150</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>HLR2/LLR5</td>
<td>70</td>
<td>52</td>
<td>3</td>
<td>10920</td>
<td>40</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3. HCV Requirements output ranks

<table>
<thead>
<tr>
<th>HLR/LLR</th>
<th>HLR points</th>
<th>LLR point</th>
<th>Comp. factor</th>
<th>Inter. priority</th>
<th>Final priority</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLR1/LLR1</td>
<td>22</td>
<td>71</td>
<td>2</td>
<td>3124</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>HLR1/LLR2</td>
<td>22</td>
<td>29</td>
<td>2</td>
<td>1276</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>HLR2/LLR3</td>
<td>78</td>
<td>43</td>
<td>3</td>
<td>10062</td>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>HLR2/LLR4</td>
<td>78</td>
<td>14</td>
<td>3</td>
<td>3276</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>HLR2/LLR5</td>
<td>78</td>
<td>43</td>
<td>3</td>
<td>10062</td>
<td>36</td>
<td>2</td>
</tr>
</tbody>
</table>
Comparing the results between HCV and VOHCV as illustrated from table 1 and 2, we conclude the followings:

- LLR5 is considered as 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 lowest 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 lowest 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” yield 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 indicates that the stakeholder assign to them small values to indicate low importance of this LLR in compare to the others. While the values for LLR5 indicates that the stakeholder assign to it a large values to indicate how important this LLR in compare to the others which detected by VOHCV.
- The results quality from VOHCV is much higher than HCV and reflect the real business case. The real business case not take only the effect of “Value” and neglect 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 [26].

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 come 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 criteria-based 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 [23].

We already designed a release management framework with as impeded VOHCV to get benefit from the results produced by VOHCV in the release planning and software product management [30]. VOHCV acts as the main core engine for the designed framework. The results from the VOHCV taken as inputs for further release planning activates to support and enable product manager to easily control and manage the product [27].

Since there are 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 [2,12,22]. 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.

8. References
