Classifying Software Requirement Prioritization Approaches

Nadina Martínez Carod and Alejandra Cechich
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue, Neuquén, Argentina
{namartin,acechich}@uncoma.edu.ar

Abstract
Defining software requirements is a complex and difficult process, which often leads to costly project failures. Requirements emerge from a collaborative and interactive negotiation process that involves heterogeneous stakeholders (people involved in an elicitation process such as users, analysts, developers, and customers). Practical experience shows that prioritizing requirements is not as straightforward task as the literature suggests. A process for prioritizing requirements must not only be simple and fast, but it must obtain trustworthy results. The objective of this paper is to provide a classification framework to characterize prioritization proposals. We highlight differences among eleven selected approaches by emphasizing their most important features.

Keywords: Software Requirements Prioritization ● Requirements Engineering ● Cognitive Informatics

1. Introduction
Requirements engineering takes care of activities which attempt to understand the exact needs of the users in a software system and to translate such needs into precise and unambiguous statements, which will be subsequently used in the development of the systems. In most cases, defects of the software are originated in the requirements phase. Once defects are embedded in the requirements, they tend to resist removal. According to Young [12], 85% of the defects of developed software is originated in the requirements. The common and more important types of requirement errors are incorrect assumptions (49%), omitted requirements (29%) and inconsistent requirements (13%).

As part of Requirements Engineering, “Elicitation” is the phase where an analyst collects information from the stakeholders, clarifies the problems and the needs of the customers and users, tries to find the best solutions, and makes its planning on what software system will be developed. Elicitation is the process of acquiring all relevant knowledge needed to produce a requirement model of a problem domain. In elicitation, to get well-defined requirements, a consensus among the different stakeholders is needed. There are several elicitation techniques in the literature [1][9][12], however every technique faces the same problem: each stakeholder has different requirements and priorities, which potentially produces conflicting situations. In these cases, stakeholders must negotiate the “right requirements” [24][25] which implies prioritisation of software requirements. Nevertheless, often the strategies implemented to solve conflicts among stakeholders are inadequate; for example, weighting requirements can be problematic because sometimes weights are inconsistent and lead to confusion about which are the most essential customer requirements [16]. More sophisticated methods, such as the AHP, and the Cost-Value [15][26], have received some interest in the application of elicitation procedures, and simpler decision-making techniques [27][28], or visualization techniques [29] have been found out to be appropriate to resolve disagreements promoting a cost-effective use. In any case, clearly defining a way of balancing preferences on requirements is essential to the elicitation process.

1 This work is partially supported by the UNComa project 04E/059.
On the other hand, the requirements elicitation techniques have widely used a family of goal-oriented requirements analysis (GORA) methods \[17\][18][19][20][21] as approaches to refine and decompose the needs of customers into more concrete goals that should be achieved. Particularly, a proposal called AGORA \[4\] extends a version of a Goal-Oriented Requirements Analysis Method by considering detecting and resolving conflicts on goals; the work in \[30\] considers greater priority when there exists a dependency between requirements, and these interdependencies can be identified before they are negotiated. More recently, the Goals-Skills-Preferences Framework \[11\] is used to generate a customizable software design; or techniques from Cognitive Informatics try to find solutions to communication problems during all stages of software engineering \[7\][8][10].

Some comparisons of elicitation methods have clarified common features. Firstly, the comparative study by Thomas and Oliveros \[5\] is centralized in properties and limitations of five of the most significant methods for eliciting requirements in goal-oriented requirements engineering. This comparison is organized from the viewpoint of goal acquisition with especial emphasis in goal elicitation. Secondly, based on an evaluation framework and influenced by an industrial application, Karlsson \[26\], characterizes six different methods for prioritizing software requirements. The objective of Karlsson’s evaluation is outlining the methods’ behaviour for a particular experience, thus the results obtained are not supposed to be generalized by any environment for any application. This evaluation framework is based on inherent characteristics, objective measures and subjective measures.

In this paper, we focus on design and cognitive aspects as main features to characterize different approaches to prioritise requirements, aiming at identifying possible improvements to the processes. The paper is organized as follows. Section 2 briefly introduces our conceptual framework. Then, Section 3 describes some approaches in terms of our framework’s features, and provides some discussion. Finally, conclusions are addressed.

2. A Classification and Comparison Framework

Our classification framework, depicted in Figure 1, is structured into two building blocks – design features and cognitive features.

The design category is composed of four elements which consider different specifications: Process, Stakeholders, Implementation and Requirements. The specific features of each prioritization requirement method are categorized by the Process element. It considers answering some questions, such as: Does the process detect inconsistency?, Is the process referred to as a systematic or a rigorous process? How we address the problem of dealing with different priorities? Conceptually, is it based in goal decomposition? Does it use a priority or an importance order?

The framework also characterizes how prioritizing methods consider stakeholders. There are two parameters to be analyzed: the former refers to the kind of information the method provides with respect to stakeholders. Does the method analyze which stakeholder prioritized a goal, and which priority degree was assigned? The second parameter considers stakeholders geographically distributed. The implementation category depends on the method’s scalability and dynamism, i.e. usability. It is influenced by how many and which calculus the method uses, and by the performance of the method with a huge number of requirements. It is considerably important whether the method is supported by tools, as well as a reference to spread projects it was applied. The framework considers information that can demonstrate the method’s success in pilot studies.
The Requirements element analyzes functional and non-functional requirements as well as interactions among requirements. Interdependency represents requirements interaction. Some methods calculate cost and benefit figures for individual requirements, but if there are significant interactions among requirements, the situation becomes more complex. As an example, if two requirements in a method can be achieved by sharing the same solutions to sub-problems, then the cost of attaining both of them may be significantly less than the sum of their individual costs. Therefore, the main key is whether the method can handle requirements' interdependencies. FR & NFR analyses study if the methods are well suited for functional and non-functional requirements.

The cognitive aspects cover the evaluation of cognitive features as participation and negotiation among stakeholders during the whole process. Evaluation studies what personal characteristics serve to establish priorities. Participation includes defining how priorities were assigned (subjective or objective) from personal experiences and interviews to ensure the success of the developed method.

To compare those features, we have applied a systematic method to validate and evaluate several proposals: the DESMET method [22]. Particularly, its feature analysis allows the framework to be expressed in terms of a set of common attributes, characteristics, or features. To judge the relative order of merit of a specific feature, it is classified in a common judgement scale: Mandatory (M), Highly Desirable (HD), Desirable (D), and Nice to have (N). Then the involved methods have to be judged according to the level of support of a particular feature.

There are two types of features: (1) simple features, that are either present or absent, and are assessed by a simple YES/No nominal scale; and (2) compound features, where the degree of support offered by the method must be measured on an ordinal scale. A different score must be assigned to simple and compound features. The following generic judgment scale is used to assess a method for a particular compound feature: 0 (No support – the feature is not supported); 3 (Moderate support – the feature is supported in some specific cases); and 5 (Strong support – the feature is supported in all cases).

An analysis based on accumulating the absolute scores must assess the relative importance of features. This analysis uses the importance assessment as a weighting factor. Although there is no defined rationale for determining appropriate weights, we use the following ones: Mandatory features (10), Highly desirable (6), Desirable (3), and Nice to have (1). Once each method has been scored for each feature of the framework by using a common scale, the results for the methods have to be compared to decide their relative order of merit.

---

**Figure 1.** A conceptual framework for comparison (compound features)
3. Characterizing Requirement Prioritization Approaches

In this section we classify some relevant approaches on requirements prioritisation presented in the literature. For brevity reasons, following we only introduce the main intent and references to the approaches. Then, we proceed characterizing them through our framework’s elements.

- **AGORA** is an extended version of the Goal-Oriented Requirements Analysis Method [4], which uses a goal graph where attribute values (contribution values and preference matrices) are added. Each stakeholder does not only attach the preference value on his own, but also estimates the preference values of other stakeholders. As a result, these preferences are represented in the form of a matrix. The stakeholders attach the value subjectively.

- The **Analytical Hierarchy Process (AHP)** Model was designed by TL Saaty as a decision making aid [15]. It involves building a hierarchy (ranking) of decision elements (candidate requirements) and then making comparisons between each possible pair in a matrix. This weights each element within a cluster (or level of the hierarchy) and a consistency ratio (useful for checking the consistency of the data). The Analytic Hierarchy Process compares alternatives in a stepwise fashion and measures their contribution to the main objective of the process [14].

- The **Cost-Value Approach**, designed by Karlsson and Ryan prioritizes requirements according to their relative value and cost [26]. In this approach, Value is interpreted in relation to a candidate requirement’s potential contribution to customer satisfaction with the resulting system. Cost is the cost of successfully implementing the candidate requirement. To investigate candidate requirements, it uses AHP to calculate each candidate requirement’s relative value and implementation cost, and plots these on a cost–value diagram. The stakeholders use the cost–value diagram as a conceptual map for analyzing and discussing the candidate requirements. Based on this discussion, software managers prioritize the requirements.

- The **Win-Win** approach [24] is a negotiation process, which enable stakeholders to work out a mutually satisfactory set of shared commitments [6]. In this methodology stakeholders express their goals as win conditions and if everyone concurs, the win conditions become agreements. When stakeholders do not concur, they identify their conflicting win conditions and register their conflicts as issues. The stakeholders are in a Win-Win equilibrium condition when the agreements cover all of the win conditions and there are no outstanding issues.

- **Quantitative Win-Win** [25] is a quantitative evaluation of alternatives of the Win-Win approach to support decision-making [13] that uses an iterative approach. The added value of this approach is its ability to offer quantitative analysis as a backbone for actual decisions. The method consist of three components: firstly it uses the Analytical Hierarchy Process for a stepwise determination of the stakeholders’ preferences in quantitative terms. Secondly these results are combined with methods for early effort estimation. Thirdly, it reflects the increasing knowledge about the requirements at each iteration cycle.

- The **Requirements Interdependencies** technique (RI) uses a conjoint analysis as a tool to determine stakeholder preferences on an individual item, and can be used to detect conflicts among stakeholders [30]. It considers the software project as a product with attributes (functional and non-functional) that define the class of a product. The technique studies the dependencies and correlations between the attributes.
• The Quality Function Deployment method (QFD) is typically applied to small subsystems [27]. A customer’s desire is the quality demanded by the customer. A quality characteristic is a measurable attribute by which one can measure whether a customer is getting the demanded quality. Quality characteristics are defined through brainstorming to generate an affinity diagram.

• The Multi-Criteria Preference Analysis Requirements Negotiation (MPARN) is a systematic model to guide stakeholders from options to agreements using multi-criteria preference analysis techniques [28]. It cooperates with the artifacts of the win-win analysis. Each stakeholder assesses each option's performance on each criterion. Many methods can be used as direct subjective evaluation, the SMART method [23], the ratio pair-wise comparison method or a geometric progression method. At last it realizes a post-analysis for agreements.

• The Visualization technique uses visualization tools to requirement conflict identification and resolves problems with exploration of potential solution approaches [29]. The technique represents stakeholder perceptions, measures consensus among the perception, and visualizes the perceptions (support collaborative prioritization of requirements among a group of stakeholders using visualization aids). It proposes Clustering Analysis as a technique to identify stakeholder subgroups having different opinions.

• The Goals-Skills-Preference framework presented in [11] is used to generate a customizable software design. In the analysis phase, the framework takes requirements as input and generates a set of ranked alternatives for the design phase. An alternative is defined as a set of tasks that together fulfill a set of target goals. In the design phase each alternative correspond to a group of software components forming a particular architecture. Developers select a set of classes according to the user’s profile. The software configuration process can be performed by the user at run time.

• The Psychotherapy for System Requirements approach consists of a series of items that can be used to assist the analysts and quality assurance of customer requirements [1][2]. This methodology is transferred from the discipline of psychotherapy to the field of requirements engineering. It can be practiced in oral and written requirements. Although this set of rules reduces the risk of getting not well-defined requirements, it only helps the analyst in the elicitation process. It is implemented using natural language in informal notation, and is not considered as an acquisition technique since it is not supported by any specification language, or any automated tool.

3.1 Applying the Framework

The “simple features” we considered to analyze processes are: (1) Consistency – specifies whether the process detect inconsistencies; (2) Rigorous – the process (method) is systematic or rigorous; (3) Goal decomposition – the process is based on goal decomposition; (4) Priority – prioritization of goals and precedence are considered; (5) Requirements Interdependence – the process identifies dependences among requirements; and (6) Objective – how the priorities are assigned (subjectively or objectively).

From Table 1, we can observe that there is no complete and simple prioritizing approach, since only some of them provide specific tools to solve conflicts. For example, some approaches as Goals-Skill-Preferences (GSP) and AGORA are based on goals; others such as Win-Win, Quantitative Win-Win and Visualization Issue technique are based on a negotiation process. We can
see both win conditions and candidate requirements as initial goals. Considering this aspect, only GSP and AGORA approaches show decomposition from needs of the customers into sub-goals. Although both AHP and Quantitative Win-Win are reliable, they require a large number of mathematical calculations to prioritize few requirements. Only Psychotherapy from System Requirements takes cognitive aspects into account allowing people specify what they really mean, but it is not a formal or systematic method. Generally, the approaches use cognitive aspects only during the negotiation phase, where the analyst must reach commitment.

Among others, cognitive aspects are one of the compound elements of our framework (Figure 1). Firstly, let us characterize the proposals according to these more detailed features as shown in Table 2. Secondly, we judge the degree of support of the compound features on an ordinal scale (0: no support; 3: moderate support and 5: strong support) with the following meanings,

- **Traceability**: “0” indicates that it is not possible to determine which stakeholder (or what group of stakeholders) prioritized each aspect; “3” indicates that it is possible to determine who prioritized some requirements, but the reason cannot be determined; and “5” is used to score the methods that keep the reason why each participant prioritized requirements.
- **Distributed stakeholders**: “0” indicates that the methods do not support collaborative environment; “3” indicates the methods are supported by distributed groups (Visualization Issue and QFD); and “5” indicates the method can operate with stakeholders in a collaborative environment (Win-Win, and Requirements Interdependence).
- **Computational tools**: “0” indicates methods with no computational support (Psych. P.R.); “3” indicates both – only some processes of the method are supported by computational tools or the computational tools are partially implemented; and “5” indicates the method is completely supported by computational tools.
- **Experience**: “0” means the method has not been empirically validated; “3” indicates small experiences/projects with real requirements; and “5” indicates the method has been used in spread projects.
- **Cognitive aspects**: “0” means the method does not consider cognitive characteristics in any aspect; “3” indicates methods which consider cognitive aspects but they do not use them in order to average weights (GSP); and “5” indicates methods where the weights of stakeholders’ perceptions can be adjusted based on stakeholder profiles (QFD).
- **Human experience**: “0” is assigned to the methods that require much experience and a great number of interviews (or too long processes); “3” is assigned to processes that although do not require much experience, they require a great number of interviews; and “5” is for processes that do not require previous experience nor several interviews (only Psych. P.R.)
- **Non functional requirements**: “0” is for the methods that cannot be used for nonfunctional requirements (AGORA, Visualization Issues, and GSP); “3” is for methods that can use non functional requirements; and “5” is assigned to methods thought for both types of requirements, (FR and NFR).

From descriptions in Tables 1 and 2, we can realize that at least three characteristics considered fundamental (traceability, distributed stakeholders and cognitive aspects) are not supported (or are little supported) by the prioritization methods.

Now, when analyzing each method with respect to its common features, we score the relative importance as mandatory features (10), highly desirable (6), desirable (3), nice to have (1). Then, each feature is assessed by its score and its specific weight depending of its importance.
<table>
<thead>
<tr>
<th>Method</th>
<th>Consistency (HD)</th>
<th>Rigorous/Systematic (HD)</th>
<th>Goal decomposition (D)</th>
<th>Priority (M)</th>
<th>Requirements Dependence (D)</th>
<th>Objective (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGORA</td>
<td>By attaching attribute values as preference matrices.</td>
<td>Rigorous process</td>
<td>It uses the AND-decomposition and OR-decomposition</td>
<td>Priorities are based on conflicting goals</td>
<td>Only in goal decomposition</td>
<td>Attribute values are attached subjectively. But techniques as AHP can be used to obtain more objective values</td>
</tr>
<tr>
<td>AHP</td>
<td>By redundancy of pair-wise comparison</td>
<td>Systematic and rigorous method</td>
<td>No</td>
<td>Compares requirements in three hierarchy level</td>
<td>No</td>
<td>Objective because it represents each term respect to other term.</td>
</tr>
<tr>
<td>Cost-Value</td>
<td>By redundancy of pair-wise comparison</td>
<td>Systematic and rigorous method</td>
<td>No</td>
<td>Idem as AHP</td>
<td>No</td>
<td>Idem as AHP</td>
</tr>
<tr>
<td>Win-Win</td>
<td>By analyzing the priorities with Conflict Consultant tool.</td>
<td>Not rigorous or systematic</td>
<td>No</td>
<td>Idem as AHP</td>
<td>No</td>
<td>Idem as AHP</td>
</tr>
<tr>
<td>Quantitative Win-Win</td>
<td>Between pairs of requirements (AHP process), eliminating some of them and checking the resulting set.</td>
<td>Systematic process</td>
<td>No</td>
<td>Idem as AHP</td>
<td>No</td>
<td>Idem as AHP</td>
</tr>
<tr>
<td>Requirements Interdependence</td>
<td>Although it detects inconsistencies, it does not have an explicit methodology to correct them.</td>
<td>Not rigorous or systematic</td>
<td>Requirement precedence can be given</td>
<td>Requirement precedence can be given</td>
<td>It is more objective than Win-Win because it adds a quantitative analysis</td>
<td></td>
</tr>
<tr>
<td>QFD</td>
<td>It does not detect inconsistencies.</td>
<td>Not rigorous</td>
<td>Precedence can be given because it is based on assigning a numeric value to each requirement</td>
<td>No</td>
<td>Priorities are given subjectively</td>
<td></td>
</tr>
<tr>
<td>Visualization Issue</td>
<td>It does not detect inconsistencies.</td>
<td>Not systematic or rigorous</td>
<td>It considers a precedence that can be shared by one or several requirements</td>
<td>No</td>
<td>Priorities are given subjectively</td>
<td></td>
</tr>
<tr>
<td>GSP</td>
<td>It does not detect inconsistencies.</td>
<td>Not systematic or rigorous</td>
<td>Each goal is a node in a goal graph, and is decomposed in OR/AND relationships into subgoals</td>
<td>It considers a precedence when evaluating the alternatives</td>
<td>No</td>
<td>It is subjective. The first part of the process (identification of objectives) can be made using any technique of elicitation</td>
</tr>
<tr>
<td>Psych. SR</td>
<td>Although it detects divergence between the stakeholders, it does not detect inconsistencies.</td>
<td>Not rigorous or systematic</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>It is subjective</td>
</tr>
</tbody>
</table>

Table 1. Characterization in terms of simple features
<table>
<thead>
<tr>
<th>AGORA</th>
<th>Traceability (M)</th>
<th>Distributed Stakeholders (HD)</th>
<th>Tools (D)</th>
<th>Experience (D)</th>
<th>Cognitive aspects (HD)</th>
<th>Human experience (N)</th>
<th>NFR (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>It allows to maintain information of objectives prioritized by each stakeholder, using the preference matrix, but not why</td>
<td>No</td>
<td>It is still not supported by computational tools</td>
<td>It has not been used in spread projects. The example proposed is a user accounting system on the Web</td>
<td>None</td>
<td>Although it requires little experience, also requires many interviews</td>
<td>It considers only functional requirements</td>
</tr>
<tr>
<td>AHP</td>
<td>The process involves almost all the stakeholders, so it does not maintain information of whom considered each priority or why.</td>
<td>No</td>
<td>An extensive bibliography of reference and several computational tools has been generated</td>
<td>It is applied by main companies and world-wide institutions</td>
<td>None</td>
<td>Although it does not need much experience, it needs several interviews to coordinate the relative values between the stakeholders</td>
<td>Although it is usually used for functional requirements, it could also be used for non-functional requirements.</td>
</tr>
<tr>
<td>Cost-Value</td>
<td>It does not maintain information of whom considered each priority or why</td>
<td>No</td>
<td>The second phase of the method is supported by a program written in language C</td>
<td>It was used in several industrial projects</td>
<td>None</td>
<td>Interviews are necessary to coordinate the relative values between the stakeholders and to review the results of the cost-value diagrams</td>
<td>It is adapted for both types of requirements</td>
</tr>
<tr>
<td>Win-Win</td>
<td>It is possible to know which participants prioritized certain objectives, but not why</td>
<td>Yes, it is designed to be able to be used in collaborative virtual environments</td>
<td>Supported by four generations of tools: 1G Win-Win, 2G Win-Win, 3G Win-Win and Easy Win-Win</td>
<td>It was used in industrial projects</td>
<td>None</td>
<td>Although many interviews are needed, it does not require too much experience</td>
<td>It is adapted for both types of requirements</td>
</tr>
<tr>
<td>Quantitative Win-Win</td>
<td>It is possible to obtain which participants prioritized certain objectives, but not why</td>
<td>No, this method is fed up on the co-participation of the stakeholders to consider new requirements</td>
<td>Some specific tools not widely used such as [31][32]. Boehm also created a prototype for his Win-Win spiral model</td>
<td>It was used in spread projects. It is widely used in industry, independently from the domain</td>
<td>None</td>
<td>Although it does not require too much experience, it requires too many interviews</td>
<td>It can be adapted to both types of requirements</td>
</tr>
<tr>
<td>Requirements Interdependence</td>
<td>It does not maintain information of who assigned each priority or why</td>
<td>Yes, since stakeholders choose products independently</td>
<td>Parts of the method are supported by tools, nevertheless it does not exist a general software that support fully this methodology</td>
<td>It was used in spread projects, usually in industry</td>
<td>It considers the political status of the stakeholders</td>
<td>It needs experience to make the process successful</td>
<td>It can be adapted to both types of requirements</td>
</tr>
</tbody>
</table>

Table 2. Characterization in terms of compound features
<table>
<thead>
<tr>
<th>Method</th>
<th>Traceability (M)</th>
<th>Distributed Stakeholders (HD)</th>
<th>Tools (D)</th>
<th>Experience (D)</th>
<th>Cognitive aspects (HD)</th>
<th>Human experience (N)</th>
<th>NFR (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QFD</td>
<td>It does not maintain any type of information from the stakeholders</td>
<td>The geometric nature of the process allows working better with isolated groups</td>
<td>This technique is partially supported by tools.</td>
<td>It has been applied successfully from 1991 in the industry of health</td>
<td>It considers the political status of the stakeholders</td>
<td>It needs experience to make the process successful.</td>
<td>It can be adapted to both types of requirements</td>
</tr>
<tr>
<td>MPARN</td>
<td>Yes, as in the Win-Win method, it is possible to obtain which participants prioritized certain objectives, but not why. Preference analysis can be a useful tool</td>
<td>No</td>
<td>The MPARN offers supports for generation and negotiation planning, for criteria exploration and assessment of scores and criteria</td>
<td>It does not mention any spread project</td>
<td>None</td>
<td>Similar to the Win-Win method. It does not require too much experience</td>
<td>It can be adapted to both types of requirements</td>
</tr>
<tr>
<td>Visualization Issue</td>
<td>Although the different priorities assigned from each requirement are known, it is not possible to know who assigns each priority or why</td>
<td>Yes, authors are even working to improve this item</td>
<td>Currently working on the elaboration of supporting tools, inspired by the previous Win-Win Distributed Collaboration Priorities Tool (DCPT)</td>
<td>It has not been used in real-world projects for case studies</td>
<td>None</td>
<td>Although it does not need much experience, it needs several interviews to negotiate priorities</td>
<td>It is thought for functional requirements</td>
</tr>
<tr>
<td>GSP</td>
<td>No. As the criteria of all the participants are joined together, it does not register who prioritized each requirement</td>
<td>No</td>
<td>There is no tool yet. It is an ongoing project.</td>
<td>It is applied to a case study involving traumatic brain injury patients</td>
<td>Yes, but it does not use it as a weight to mediate. It is one of the most remarkable characteristics</td>
<td>It needs much experience and many interviews to determine; for each user, goals, skills and preferences</td>
<td>It is developed only for functional requirements</td>
</tr>
<tr>
<td>Psych. SR</td>
<td>No. As the criteria of all the participants are joined together, it does not register who prioritized each requirement</td>
<td>No.</td>
<td>It does not make calculations of any type. It is not supported by tools</td>
<td>It is used in many small projects, but it is not used in great projects.</td>
<td>It does not consider cognitive characteristics of any of the participants</td>
<td>It does not need much experience, which is obtained in two or three days of training</td>
<td>It can be adapted to both types of requirements</td>
</tr>
</tbody>
</table>

Table 2 Characterization in terms of compound features (Cont.)

Figure 1 shows the comparative representation of the results for the methods, with respect to simple features.

In addition, four levels may be defined for this classification by considering simple features according to their importance. As an example, AGORA would be classified into the first level since it supports M and HD features; AHP, Cost-Value, and Quantitative Win-Win would be members of this level too, since they support an M feature and some HD features. The methods Win-Win, Requirements Interdependency, and MPARN would be members of the second level – they do not support any mandatory feature. Finally, GSP, Visual Issue and QFD are members of the third level (they do not support highly desirable features). The fourth level appears for completeness reasons by considering methods that support nice to have features, as in Psych.S.R.
As simple features, the maximum value that can be assigned to a method in this classification is 155, obtained by weighting scores of each feature \((155 = (6+6+3+10+3+3) \times 5)\). For example, AGORA’s result is calculated as \((6+6+3+10+3) \times 5 = 140\); or the AHP’s result is calculated as \((6+6+10+3) \times 5 = 125\). This information can be analyzed from two viewpoints—the first one considering the most significant characteristics, and the second according to the sum of their relative weights. Then, the method to be discharged immediately is “Psych. S.R”, because it does not show any of the mentioned characteristics.

In the case of compound features, the first analysis is more difficult to make since we analyze aggregated features. Therefore we analyze only the sum of the relative weights. Here, we can differentiate four levels again, and discharge the last level because of excessively low values. At the higher level we find the Win-Win method; then the following level includes the methods Quantitative Win-Win, Requirements Interdependence, QFD, MPARN, and Visualization Issue. The third level includes AGORA, AHP and Cost-Value; and finally the methods GSP and Psych.SR are members of the last level. By considering the sum of relative weights and by defining ranks for each level, we establish the following ranks: Level 1 (160-100); Level 2 (99-66); Level 3 (65-38), Level 4 (64-34); Level 5 (63-32); Level 6 (62-30); Level 7 (61-28); Level 8 (60-26); Level 9 (59-24); Level 10 (58-22); Level 11 (57-20); Level 12 (56-18); Level 13 (55-16); Level 14 (54-14); Level 15 (53-12); Level 16 (52-10); Level 17 (51-8); Level 18 (50-6); Level 19 (49-4); Level 20 (48-2); Level 21 (47-0).
Level 4 (37-0). The sum of the scores of all methods by combining values from Figure 1 and Figure 2, are shown in Table 3.

Finally, we proceed normalizing scores to facilitate comparison. Figure 3 shows percentages obtained by all the methods in relation to the maximum possible value (315, which represents 100% in a graphical representation). As we can see, Win-Win, Quantitative Win-Win, and AGORA result with the highest scores.

```
<table>
<thead>
<tr>
<th>Method</th>
<th>Simple Features</th>
<th>Compound Features</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win-Win</td>
<td>95</td>
<td>108</td>
<td>203</td>
</tr>
<tr>
<td>AGORA</td>
<td>140</td>
<td>51</td>
<td>191</td>
</tr>
<tr>
<td>Quantitative Win-Win</td>
<td>125</td>
<td>66</td>
<td>191</td>
</tr>
<tr>
<td>AHP</td>
<td>125</td>
<td>42</td>
<td>167</td>
</tr>
<tr>
<td>Cost-Value</td>
<td>125</td>
<td>39</td>
<td>164</td>
</tr>
<tr>
<td>Requirements Interdependence</td>
<td>95</td>
<td>69</td>
<td>164</td>
</tr>
<tr>
<td>MPARN</td>
<td>95</td>
<td>66</td>
<td>161</td>
</tr>
<tr>
<td>QFD</td>
<td>50</td>
<td>81</td>
<td>131</td>
</tr>
<tr>
<td>Visualization Issue</td>
<td>50</td>
<td>75</td>
<td>125</td>
</tr>
<tr>
<td>GSP</td>
<td>65</td>
<td>27</td>
<td>92</td>
</tr>
<tr>
<td>Psych. SR</td>
<td>0</td>
<td>23</td>
<td>23</td>
</tr>
</tbody>
</table>
```

Table 3. Scores for the analyzed methods

4. Conclusion

Requirements prioritization has been pointed out as a relevant research area in requirements engineering, calling for the definition of effective methods and techniques that enable to rank a whole set of requirements, according to relevant criteria, such as business goals or technical features. We present both a classification framework for requirements elicitation processes and an analysis of eleven methods using the conceptual framework. We hope our work helps requirement engineers to identify and rank functionalities, which are useful during elicitation.

As future work, we are improving prioritization by considering stakeholders’ profiles using cognitive aspects of stakeholders. We suggest improving communication and reduce misunderstandings based on Cognitive Psychology. This can be done by extending the classification mentioned in [3] to consider behavioral characteristics of the way people think and process information.
References