A Method for Collaborative Requirements Elicitation and Decision-Supported Requirements Analysis

Michael Geisser and Tobias Hildenbrand
University of Mannheim, Schloss, 68131 Mannheim, Germany,
{geisser, hildenbrand}@uni-mannheim.de,
WWW home page: http://wifo1.bwl.uni-mannheim.de/team.html

Abstract. As software systems become more and more complex with a multitude of stakeholders involved in development activities, novel ways of conducting the process of requirements elicitation and analysis are to be found. Therefore, this paper introduces a method for collaborative requirements elicitation and decision-supported requirements analysis. Accompanying this method, appropriate tools and techniques, both existing and custom-made, are referred to. The method is designed for a geographically distributed collaborative environment in order to support software manufacturers as well as IT departments which develop software solutions for multiple users or even consortiums of customers.

1 Introduction

Since the ‘60s, numerous methods for a more systematic approach to software development have been devised as part of the newly created software engineering (SE) discipline. SE in general aims at consistently producing high-quality software within predictable budget restrictions and project schedules. However, even today surveys show that the majority of all software projects significantly run out of schedule and budget. This and further problems in software projects are mostly caused by a lack of understanding of the customers’ needs at the beginning of the project as well as by unsystematic approaches to early development activities [14, 24, 25]. The discipline of requirements engineering (RE) focuses on these early stages of software development projects.

Introducing a more systematic method for RE constitutes a fundamental prerequisite for realizing the goals of SE. This task is even more complicated when

considering consortiums of multiple customers: This implies the involvement of numerous stakeholders from different organizations. In this particular scenario, it is of high importance to systematically guide the stakeholders with their respective opinions through the RE process in order to reach a consensus which the consequent stages of SE can build upon.

In an aim to support software manufacturers in addressing these complications, this paper provides a theoretically sound method accompanied by appropriate tools for collaborative requirements elicitation including decision support for requirements analysis. The CoREA method (Collaborative Requirements Elicitation and Analysis) aims at enabling software companies to systematically elicit requirements in a distributed environment and provides profound and objective decision support for analyzing and selecting relevant requirements.

After having already outlined the paper's underlying problem statement and objective, an overview and critical evaluation of related RE approaches and methods will be given as theoretical framework. Section 3 contains a description of the method consisting of two major parts: (a) eliciting a complete set of requirements with regards to a distributed collaborative scenario and (b) analyzing those requirements in order to find a reasonable and objective choice for implementation. Supportive tools for each step of the method will also be presented. The concluding section summarizes the results of our research, including a demarcation to previous work, and provides an outlook on future research questions.

2 Related Work

As already indicated, most problems in software development stem from a poor initial understanding of the customers' needs. RE deals with this difficulty and tries to systematically create a better understanding in the early stages of a SE project. The most common definition of the RE process is that of Ian Sommerville: "The requirements for a system are the descriptions of the services provided by the system and its operational constraints. [...] The process of finding out, analysing, documenting and checking these services and constraints is called requirements engineering" [23]. This process is subdivided into four phases, namely feasibility study, requirements elicitation and analysis, requirements specification, and requirements validation. Parallel and subsequent to these phases, requirements management covers all activities concerning the management of emerging changes to requirements during the whole software development process [23].

2.1 Collaborative Requirements Engineering

As Cook und Churcher observed, „Software Engineering is inherently a team-based activity" [6], and thus, SE, and RE in particular, are not feasible without a certain degree of collaboration, in most cases. Moreover, involving all relevant stakeholders early on in the process is particularly crucial for successful software projects [2]. Among all RE phases, requirements elicitation and analysis is an especially

collaborative stage: first, stakeholders from both the software company and the customer need to be identified, and second, requirements from all these stakeholders have to be gathered collaboratively. In particular, requirements analysis takes place among stakeholders from the ordering party supported by the software vendor. Requirements specification is carried out collaboratively as well: the pivotal activity (modeling), can only be successful after continuous consultation with the customers' stakeholders. Many computer scientists advocate an even deeper involvement of all stakeholders within the requirements specification phase by means of collaborative methods [1, 9]. The remaining phases of RE are by far less collaborative than the two previously mentioned. In the following, existing approaches to collaborative requirements elicitation and analysis will be the center of attention.

2.2 Collaborative Requirements Elicitation and Analysis

Considering scientific approaches for collaborative requirements elicitation and analysis, there is only one established research endeavor, namely the WinWin approach. Originating from Boehm's Theory W [5], WinWin has evolved over four iterations from an extended spiral model of software development [4] to the latest version, called EasyWinWin (EWW) [10]. This approach propagates a change from traditional, contract-oriented mechanisms to collaborative practices based on trustful relationships among stakeholders. EWW does not aim at rigid agreements and detailed requirements specifications. It rather tries to provide the stakeholders involved with a shared vision and common beliefs in order to be able to react to both unforeseen problems and opportunities in an adaptive and quick manner [3]. The establishment of trust among all team members is an integral constituent of the EWW method. Additionally, this approach leads to more realistic expectations among stakeholders, since they exchange and scrutinize their respective beliefs by means of intensive discourse. Moreover, EWW is able to reveal tacit knowledge as well as conflicts and inconsistencies in very early stages of the requirements elicitation and analysis phase [12]. Other advantageous features of this method include its detailed process description, which provides certainty and guidance for participating stakeholders, as well as its supportive groupware tools. Thus, EWW combines the WinWin spiral model of SE with collaborative knowledge techniques and automation of a custom-built group support system [5].

The relatively high complexity constitutes the major downside of this approach since the process is not very intuitional and necessitates training for both moderators and participants. Moreover, the process is not tailored to a distributed environment as physical discussions are a fundamental element of the method. The relatively high subjectivity of requirements selection accounts for another disadvantage. Although EWW tries to guarantee a certain degree of objectiveness by means of a prioritization mechanism, the absolute character of this mechanism is inferior to comparative ones [15]. Another drawback is the "ease of realization" criterion for assessing requirements. Since this criterion incorporates numerous factors it is arguable whether all stakeholders are capable of rating this property on an absolute scale. The directive not to vote unless stakeholders feel able to assess this criterion is

also problematic, since the participants' subjective appraisement may differ significantly from their actual abilities. Table 1 provides an overview of EWW's advantages and disadvantages in context of our initial problem statement.

Table 1. Evaluation of the EasyWinWin method

Advantages	Disadvantages
+ Flexibility	- Not very intuitional
+ Establishment of trust	- Not suitable for distributed development
+ Realistic expectations	- Relatively high subjectivity
+ Revelation of tacit knowledge	
+ Early detection of conflicts	
+ Detailed process description	
+ Tool support (groupware)	

2.3 Distributed Requirements Elicitation and Analysis

The gradual globalization of economies makes highly distributed software development techniques indispensable. The driving force and rationale behind this development is the opportunity to share resources and to use wage differentials on a global scale. Against this background, not only the distributed SE process as a whole has been subject to researchers' investigations [1] but also distributed RE, and particularly requirements elicitation, has been studied empirically [7, 8, 13]. However, these studies unanimously deal with distributed elicitation activities using traditional techniques and methods not necessarily suitable for distributed environments. Furthermore, many asynchronous techniques (e.g. shared glossaries and discussion forums) are not explicitly taken into consideration. However, all studies deem distributed requirements elicitation possible and even favorable compared to collocated approaches. In order to realize this potential advantage, methodical principles need to be taken into consideration and requirements for tool support have to be granted. E.g. initial face-to-face meetings are considered essential in order to establish trustful relationships among the persons involved [8]. Important requirements for collaborative tools include support for both synchronous and asynchronous collaboration capabilities [13].

With regards to EWW's original groupware, geographically distributed stakeholders were only integrated in a rudimental way. Therefore, a web-based tool for distributed requirements elicitation supporting the EWW approach was developed: ARENA [11]. However, this tool does not complement the existing groupware tools but replaces them. Therefore, in order to conduct collaborative, distributed requirements elicitation and analysis, the whole process has to be run within the boundaries of the ARENA tool. This, in turn, is very problematic, since ARENA solely supports web-based asynchronous collaboration. Thus, it is impossible to arrange synchronous meetings which play a pivotal role within the original method. Besides ARENA, two other applications supporting EWW were developed especially for mobile devices. Thus, it is possible not only to conduct requirements elicitation in a geographically distributed setting but also without any

tie to fixed desktop workplaces. These mobile tools are especially useful in scenarios where collocated workshops are held in combination with interviewing geographically distributed stakeholders [22].

Open source software development (OSSD) constitutes another source of insight into techniques for distributed requirements elicitation and analysis. In OSSD, the overall development process is primarily distributed. Therefore, further findings for the course of this paper can be derived – especially when considering the major downsides of EWW, namely being non-intuitional and not suitable for distributed environments. However, major differences between commercial software projects and OSSD can be found, in particular when comparing requirements processes. Unlike commercial developers, open source developers are mostly among the future users of the software product [17]. Empirical studies reveal that requirements processes in OSSD projects run much more implicitly and informally than in any other kind of development project – sometimes even omitting some of the generally accepted RE activities [17, 21]. In particular, requirements elicitation and analysis is carried out much more informally than in traditional RE, as requirements are elicited, elaborated, and discussed in forums and via mailing lists. Especially in case of distributed environments, forums represent an efficient way of asynchronously eliciting requirements even in commercial settings – particularly in terms of resource consumption. However, these forums should be structured and supervised by a moderator, in order to coach those stakeholders not so familiar with the medium and to run the process as systematically as possible.

2.4 Quantitative Approaches to Requirements Engineering

The RE process has to consider various requests from diverse stakeholders, each having a different view on the system to be built and thus having varying priorities. Furthermore, most stakeholders are unaware of the implementation costs of the respective requirements. Due to budget restrictions, it is generally impossible to incorporate all the stakeholders' requirements in the final software product. Therefore, a reasonable selection has to be conducted in order to maximize customer value [19]. In the literature, two major methods supporting quantitative RE can be found: the Cost-Value Approach [15], and Quantitative WinWin [18, 19]. Both methods base upon the Analytic Hierarchy Process (AHP) [20], a supportive method for complex team decision processes which has proved to be superior to other requirements prioritization algorithms in RE [16].

The **Cost-Value Approach** (CVA) features intuitional and easy handling. In addition, this method leads to better results than absolute ones due to its solid mathematical foundation. The AHP's pairwise comparisons have a detrimental effect, since the method's complexity rises exponentially compared to the number of requirements. Neither are possible interdependencies between requirements considered [15]. Thus, e.g. a requirement with a very low value-cost ratio might be indispensable for implementing another requirement with a very high value-cost ratio. The CVA would advise to omit this indispensable requirement, even though the global maximum of customer value could thus never be attained.

Quantitative WinWin (QWW), on the other hand, considerably reduces the number of comparisons by using the AHP hierarchically [18]. However, the effect of the AHP's pairwise comparisons still has a negative impact on the process, since several iterations are extremely demanding in terms of the stakeholders' cooperation and willingness to participate. Therefore, QWW is still more complex than the CVA. It also features a solid mathematical foundation and thus overcomes the limitations of a subjective requirements selection. The stakeholders' cooperation is even more mission-critical when evaluating the relative importance of requirements as proposed in the extended version of this approach [19]. Nevertheless, the method's original assumption that the relative importance values of requirements are given has to be considered quite unrealistic. Moreover, when estimating costs (as well as duration and quality in the extended version) using the proposed simulation system represents more of a risk than an improvement, since the expected quality of results from this estimation is at least arguable [19]. Furthermore, neither consistency checks of the stakeholders' AHP comparisons nor interdependent requirements are taken into consideration. These interdependencies are particularly crucial, since it can be assumed that both value and complexity of respective requirements will not stay constant but will rise with a growing number of implemented features [19]. Finally, the method's name is somehow misleading, because it has nothing in common with the original WinWin approach but the iterative nature of the process. Table 2 outlines the results of the quantitative methods' evaluation.

Table 2. Comparison of Cost-Value Approach and Quantitative WinWin

Method	Advantages	Disadvantages
Cost-Value Approach	+ Mathematical foundation + Cost-value consideration + Consistency check + Intuitional handling	- No consideration of interdependencies among requirements - Complexity
Quantitative WinWin	+ Mathematical foundation + Cost-value consideration + Hierarchical AHP	- No consideration of interdependencies among requirements - High complexity - No consistency check - Cost estimate problematic - Close cooperation among stakeholders needed

3 Introducing the CoREA method

Based on the analysis and evaluation of existing approaches, we now introduce the CoREA method for collaborative RE. CoREA covers collaborative requirements elicitation in a distributed environment as well as quantitative decision support for distributed requirements prioritization and selection. The CoREA method consists of two distinct phases: Phase I is predominantly concerned with the iterative and collaborative elicitation of requirements from different stakeholders, while explicitly taking into account geographically distributed work. Subsequently, in phase II, costs

and values of the respective requirements are analyzed in order to support the selection process with regards to the ensuing design and implementation phases.

3.1 Collaborative Requirements Elicitation

In phase I of the CoREA method, requirements are elicited both collaboratively and iteratively. The method builds upon EWW but uses techniques from OSSD in order to achieve both a more intuitional procedure and consistent support for distributed collaboration. The objective of this first phase is to capture the requirements as completely as possible. Hence, a vague vision conceptualizing the customers' needs serves as starting input. Moreover, an initial list of relevant stakeholders must be available. The set of relevant stakeholders as well as the central vision evolves over time, as several iterations of the process will be traversed. The respective process steps for CoREA's collaborative requirements elicitation phase will be described in detail in the following sections.

Step 1: Initial Meeting

Within the scope of the initial meeting the vision statement along with a first list of stakeholders is handed over to the software company. This meeting enables the establishment of interpersonal relationships among the stakeholders who are supposed to collaborate predominantly asynchronously and geographically distributed within the following steps.

Step 2: Brainstorming

Asynchronous brainstorming aims at generating first ideas about the software to be developed in the project. Web-based forums are utilized to enable geographically distributed collaboration among stakeholders. Thus, they are able to generate new ideas as well as complement and comment existing entries. Whereas criticism during brainstorming sessions is often interdicted, CoREA prescribes this explicitly in order to reject unrealistic requirements as soon as possible in the RE process. This second step is supposed to be supported intensely by a moderator from the software manufacturer who supervises and adjusts the detail level of discussion, if necessary. Furthermore, the moderator ensures the correct and consistent usage of technical terms, e.g. by systematically asking questions. In addition, he fosters active participation of all stakeholders by purposefully addressing people.

Step 3: Revise Vision and Identify Categories

After having completed the brainstorming step, the vision document has to be revised by the moderator and a further SE expert from the software company. Their task is to incorporate the ideas previously generated in step 2. In addition, categories for upcoming requirements need to be identified from the given sets of ideas in order to guarantee a structured procedure in the subsequent steps. At the same time, a SE expert tries to identify and reject unrealistic proposals and thus ensures the system's realizability and technical feasibility. Moreover, the expert detects technical terms, which have to be defined in a common glossary.

Step 4: Prioritize Categories & Discussion

The prioritization of requirement categories and subsequent discussion occurs within the scope of a virtual meeting. Alongside the moderator who guides all participants

through the process and all stakeholders provided by the customer, the SE expert from step 3 also has to participate in this meeting. In order to realize such a virtual meeting, multimedia-based groupware is necessary. In particular, audio and video conferencing as well as anonymous polling features are vital for conducting this step. At first, the proposals rejected in step 3 will be paid attention to and the SE expert has to justify their exclusion. Afterwards, the stakeholders have to conduct an anonymous prioritization of requirement categories. In doing so, each category's importance has to be assessed from the customers' organizations' points of view on a scale ranging from 0 (not important at all) up to 3 (extremely important). A more detailed graduation of the scale would not be appropriate at this point, since the stakeholders' perceptions are still relatively imprecise and significant differences in categorization are yet to be detected. In case of substantial differences in the stakeholders' assessments of particular categories the meaning and the relevance of this category have to be discussed intensely. This discussion aims at reaching a consensus among all stakeholders involved. After the discussion, the moderator presents the revised vision and incorporates further changes if necessary. The list of technical terms identified for the glossary will also be shown and, if required, complemented by further terms. This step concludes by deciding whether new stakeholders have to be involved for the ongoing course of the elicitation process and which of the current stakeholders are dispensable for the time being.

Step 5: Create or Revise Glossary

The creation of the glossary containing technical terms identified in the previous steps is supposed to be conducted asynchronously and geographically distributed. For this purpose, a web-based technology, e.g. a Wiki system or comparable groupware systems allowing collaborative, asynchronous document editing over the Internet, can be utilized.

Step 6: Submit and Comment Requirements

Again, a structured web-enabled discussion forum is utilized in order to be able to both submit new and comment on existing requirements asynchronously and from different geographic locations. In this forum, the moderator creates different areas for the respective requirement categories as well as one additional area for requirements that could not have been categorized so far. As in step 2, the moderator tries to resolve ambiguities by asking questions, requests more precise explanations and fosters active participation by all stakeholders.

Step 7: Consolidate and Categorize Requirements

In this step, the requirements submitted and annotated via the discussion forum have to be consolidated by the moderator and the SE expert from the software manufacturer. Thereby, all findings from the respective discussion threads have to be merged. After that, these consolidated requirements are allocated either to existing categories or newly created ones. While allocating requirements the SE expert pays attention to the fact that interdependent requirements are not classified in different categories. He also tries to identify and eliminate proposals for unrealistic requirements. In addition he compiles technical terms to be specified in the glossary. If necessary, the vision might be revised and adapted as well.

Step 8: Prioritize Categories and Requirements & Discussion

In order to collaborate effectively in terms of costs and time consumption as well as to establish trust and interpersonal relationships among stakeholders, organizing alternating physical and virtual meetings is a promising approach. Thus, in case step 8 has to be traversed several times and the most recent meeting was a virtual one, the following iteration demands for a physical meeting. This step is conducted analogously to step 4. However, besides prioritizing and discussing categories, requirements themselves are also to be dealt with at this point. In case the glossary has to be revised or new stakeholders have been identified, another iteration starting with step 5 has to be traversed. Otherwise, all participants check the categories in terms of completeness. If there are uncompleted categories, another partial iteration traversing steps 5 to 8 is required. If no further iterations are required, the phase I of CoREA is considered completed. Figure 1 depicts a spiral model of the requirements elicitation process in order to visualize the method's iterative character and contextualize the respective steps.

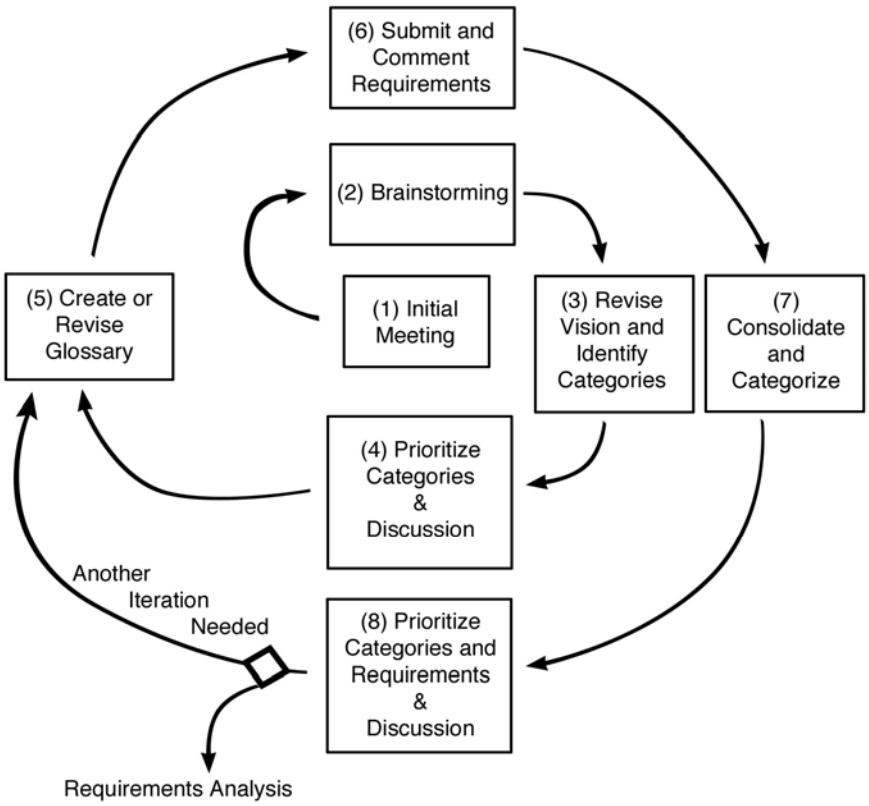

Fig. 1. CoREA Spiral Model of Collaborative Requirements Elicitation

3.2 Decision-Supported Requirements Analysis

Within the second phase of the CoREA method, requirements are selected for actual implementation based upon a quantitative analysis of costs and customer value. The starting point of this process is a list of requirements, as it was gathered and consolidated during the requirements elicitation phase. From an economic point of view, implementing only those requirements providing satisfactory value as compared to their costs is considered reasonable. Monetary budget restrictions can also necessitate a more deliberate selection of requirements. Thereby, this selection is conducted according to the value-cost ratio: the requirements with the highest ratios will be implemented.

Step 1: Form Requirements Sets

Since requirements always bear interdependencies among each other, they cannot be compared in a way that neglects these interdependencies. If one or more categories (cp. section 3.1) contain interdependent requirements, so-called requirements sets have to be formed. Figure 2 shows a graphical representation of interdependent requirements and requirements sets. In this example, requirement A2 is a prerequisite for A3. The latter, together with A0, is in turn a precondition for A4 and A5. Taken together the directed graph forms a self-contained requirements set. A1 does not depend on any other requirements and thus forms a set of its own. Requirements set 3 consists of two interdependent requirements B1 and B2 and the implementation of the former is a prerequisite for the latter.

Step 2: Estimate Costs and Values

As soon as the requirements sets have been formed within the different categories, costs and values for requirements and requirements sets have to be estimated. While the software company's SE expert is exclusively responsible for realistic cost estimations, estimating the requirements' value is up to the stakeholders provided by the customers. Costs are estimated on the one hand on a quantity basis (e.g. by man-days) and on the other hand on a value basis (e.g. daily rate per employee). Customer value is determined by means of the AHP (see section 2.4).

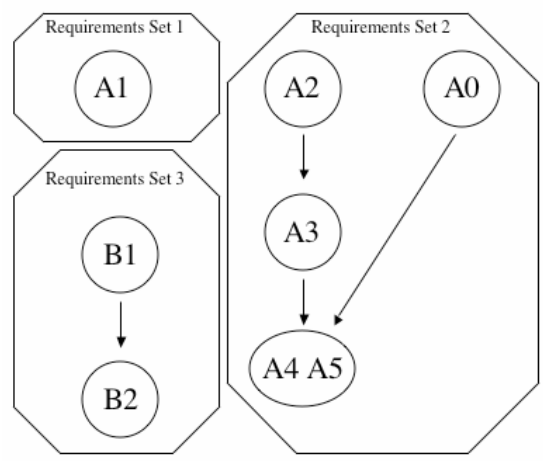

Fig. 2. Graphical Representation of Interdependencies

Step 3: Graphical Representation of Results

Results from step 2 are represented graphically with regards to interdependent requirements by depicting all possible combinations originating from root requirements in the directed graphs with their respective aggregated cost and value estimates. Figure 3 takes on the example given in step 1 (see Figure 2) displaying possible combinations of requirements. The diagram displays the different combinations and their respective cost-value characteristics. In order to support cost and value estimation especially for the CoREA method, a prototypical web application has been implemented. This prototype enables geographically distributed stakeholders to be securely guided through the estimation process. It implements the AHP algorithm and is able to visualize the results in the form of a cost-value diagram as shown in Figure 3 (see appendix).

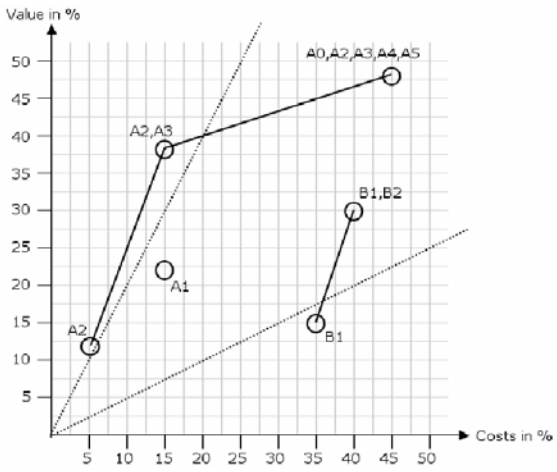

Fig. 3. CoREA Cost-Value Diagram

Step 4: Decide Upon Selection

Finally, a physical meeting of all stakeholders is conducted. The moderator presents the cost-value diagram with all possible requirements combinations and their cost and value estimations resulting from step 3. Based on this objectified foundation, it has to be decided which requirements will be implemented immediately, totally discarded, or preserved for upcoming releases. In order to provide additional decision support, the diagram contains two straight lines: requirements with at least two times more relative value than relative cost should be implemented in any case, whereas those with twice the relative costs should not be considered for implementation. These equations have been empirically tested and proven themselves suitable to distinguish preferable requirements with high value-cost ratios from those with a low ratio [15]. Finally, Figure 4 gives a visual overview of CoREA's requirements analysis phase. In combination with Figure 1 this depicts the overall CoREA method.

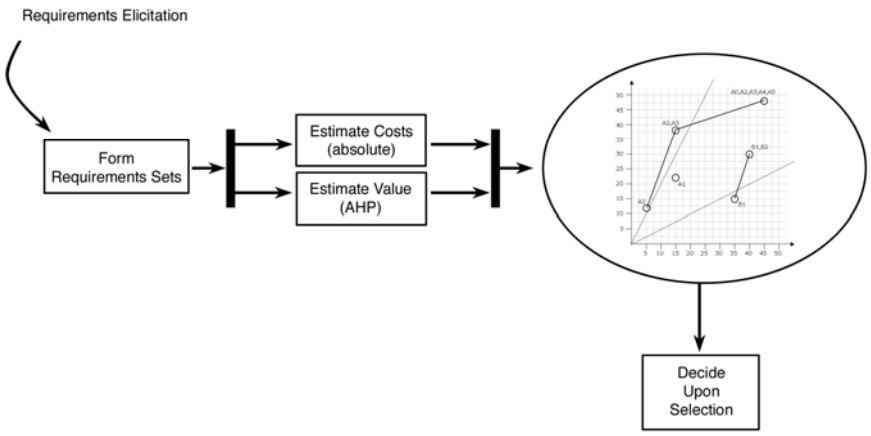

Fig. 4. Decision-Supported Requirements Analysis in CoREA

4 Conclusion

Based upon a critical evaluation of existing approaches, this paper introduces a novel, decision-supported method for collaborative requirements elicitation and analysis suitable for a distributed environment. This method consists of two subsequent phases. While requirements are elicited iteratively and as completely as possible in the first phase of the CoREA method, phase II provides methodic guidance for selecting those requirements that will actually be implemented. CoREA thus enables software manufacturers to systematically elicit the requirements collaboratively with customers in a distributed environment. This effect is achieved by transferring the established WinWin approach into a geographically distributed environment. Moreover, CoREA improves WinWin in terms of intuitional handling and objective requirements selection procedures. By enhancing WinWin's core properties, our method builds upon the vast theoretical and empirical knowledge gathered in the field of collaborative requirements elicitation. We are able to eliminate WinWin's well-known weaknesses through additional insights in the fields of distributed software development and quantitative methods for requirements evaluation. Besides enhancing EWW, CoREA for the first time takes interdependencies into account by introducing requirements sets as units of evaluation. This method and the tool prototype have been developed in close cooperation with the IT departments of two large German financial institutions.

To be able to gain additional empirical evidence, the method will be applied within several case studies. Since CoREA was developed within the scope of a larger research consortium, access to practical settings is ensured. Based on the practical experience from upcoming case studies, both tool support and the method itself will be improved and adapted.

Alongside prototypical evaluation, it is useful to complement CoREA through broadening the theoretical foundations. Even though it is deemed hard to design domain-specific methods for RE, it has yet to be analyzed, whether domain-specific process instances can be generated by means of ontologies and other semantic technologies. Moreover, requirements analysis and selection can be extended by time-related aspects as the current estimation of the requirements' costs and value might be complemented by taking development time into consideration. This in turn, is useful for process planning and control. Furthermore, the method's integrability with product line concepts in SE and traceability capabilities have to be analyzed in order to facilitate proactive reuse of requirements. Considering component-based software development methodologies, techniques for matching standard sets of requirements with standard infrastructure and business components are an open field of research as well.

In order to develop an integrated methodology for collaborative RE, future work also has to deal with adapting requirements specification and validation processes for distributed environments. Thus, the full potential of distribution, specialization and collaborative work can be exploited in the early stages of SE. Such an integrated methodology allows a better focus on the very early stages of SE. Hence, it provides a sound basis for inter-organizational division of labor, and faster realization of new software solutions. In doing so, higher quality is eventually achieved through the integration of multiple stakeholders with diverse competencies. In addition, an improved RE process leads to less consequential defects in later phases which become more expensive the later they emerge. The issues discussed in this paper do not only apply for RE but for the whole SE process and software lifecycle respectively. Enabling and improving distributed work, whether organizationally or geographically distributed, will play an important role in the course of the global industrialization process within the software sector. Therefore, considering the entire SE process, integrated methodic and technological support for collaborative software development projects are becoming more and more important in the future.

5 Appendix: Tool Prototype

In order to support cost and value estimation for requirements evaluation an internet-based prototype has been developed. This prototype is called IBERE (Internet-Based Empirical Requirements Evaluation) and guides distributed participants securely through the requirements estimation procedure. IBERE is also able to visualize the results of the requirements evaluation process in the form of a cost-value diagram by utilizing the AHP algorithm for calculating the utility value for each requirement. Thus, this prototype supports steps 2 and 3 of CoREA's decision-supported requirements analysis (cp. section 3.2). The screenshot in Figure 5 depicts pairwise comparisons of requirements within one set as part of the AHP procedure.

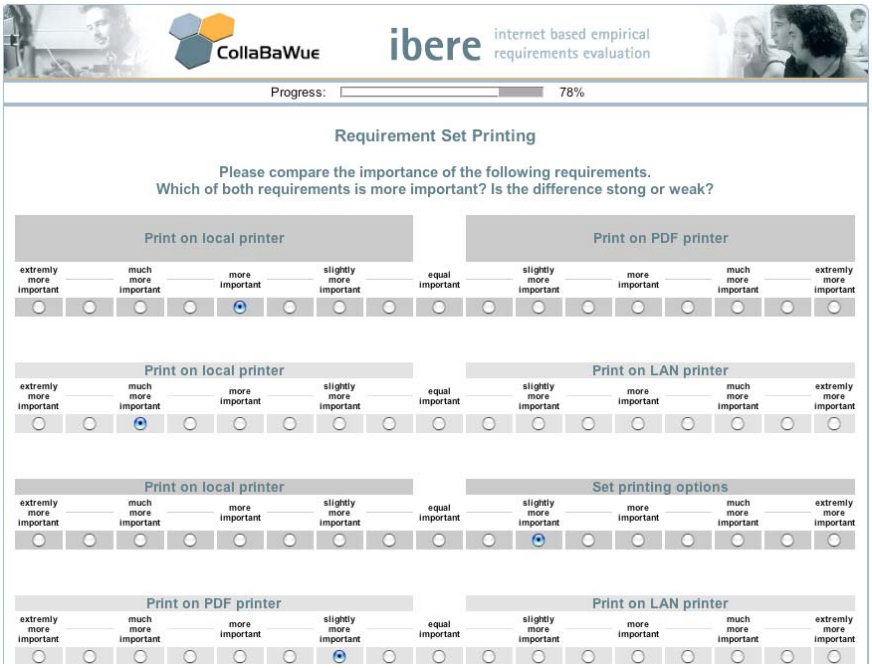

Fig. 5. Pairwise Comparison of Requirements with IBERE

Figure 6 depicts the graphical representation of the AHP's results (cp. step 3 in section 3.2). In this example, requirements 1.1, 1.3 and 2.1 should be implemented due to their high value-cost ratios, as indicated by their positions above the upper straight line. In contrast, the requirements 1.4 and 2.3 should not be taken into consideration for the final software product because of their unfavorable value-cost ratios. The consideration of requirements interdependencies (cp. Figures 2 and 3) in IBERE is currently under development and therefore cannot be shown in this screenshot.