Softw Syst Model (2008) 7:303–328
DOI 10.1007/s10270-007-0055-y

REGULAR PAPER

A methodology for the selection of requirements engineering techniques

Li Jiang · Armin Eberlein · Behrouz H. Far · Majid Mousavi

Received: 15 November 2004 / Revised: 28 January 2007 / Accepted: 5 March 2007 / Published online: 30 June 2007

Abstract The complexity of software projects as well as the multidisciplinary nature of requirements engineering (RE) requires developers to carefully select RE techniques and practices during software development. Nevertheless, the selection of RE techniques is usually based on personal preference or existing company practice rather than on characteristics of the project at hand. Furthermore, there is a lack of guidance on which techniques are suitable for a certain project context. So far, only a limited amount of research has been done regarding the selection of RE techniques based on the attributes of the project under development. The few approaches that currently exist for the selection of RE techniques provide only little guidance for the actual selection process. We believe that the evaluation of RE techniques in the context of an application domain and a specific project is of great importance. This paper describes a Methodology for Requirements Engineering Techniques Selection (MRETS) as an approach that helps requirements engineers select suitable RE techniques for the project at hand. The MRETS has three aspects: Firstly, it aids requirements engineers in establishing a link between the attributes of the project and the attributes of RE techniques. Secondly, based on the evaluation schema proposed in our research, MRETS provides an opportunity to analyze RE techniques in detail using clustering. Thirdly, the objective function used in our approach provides an effective decision support mechanism for the selection of RE techniques. This paper makes contributions to RE techniques analysis, the application of RE techniques in practice, RE research, and software engineering in general. The application of the proposed methodology to an industrial project provides preliminary information on the effectiveness of MRETS for the selection of RE techniques.

Keywords Requirements engineering · Technique evaluation · Decision support · Clustering

Communicated by Professor Roel J. Wieringa.

L. Jiang (✉)
School of Computer Science, The University of Adelaide,
Room 2069, Adelaide, SA 5005, Australia
e-mail: ljiang@cs.adelaide.edu.au

A. Eberlein
Computer Engineering Department,
American University of Sharjah, Sharjah, UAE
e-mail: eberlein@ucalgary.ca

B. H. Far · M. Mousavi
Department of Electrical and Computer Engineering,
University of Calgary, Calgary, Canada
e-mail: far@ucalgary.ca

M. Mousavi
e-mail: amousavi@ucalgary.ca

1 Introduction

The requirements engineering (RE) process is part of the overall software lifecycle and plays an important role in ensuring the overall quality of a software product. Literature already provides empirical evidence for the benefits of RE [1–3], and shows that improving the RE process leads in most cases to improvements in the productivity of large and medium-sized software organizations [4–6]. Currently, there are numerous techniques that address different aspects of the RE process and system development [7] and that can be applied to various types of projects.

The significance of choosing proper techniques and models during software development has already been emphasized by researchers and practitioners. Glass and Vessey [8] stressed that we need to choose the most appropriate software development methodology for the task at hand [9–12]. Davis states that knowing which technique to apply for a given problem is necessary for effective requirements analysis [13]. In our research, we have found that different techniques have different advantages; some techniques are relatively mature while others are not; some techniques have similar functionality but different complexity; and some are functionally complementary to each other [14]. Therefore, the use of the best combination of RE techniques will facilitate the RE process and contribute to the overall quality of the requirements specification.

However, the selection of suitable RE techniques in the context of a specific software project is a challenging issue faced by RE practitioners. Numerous variables influence this decision making process. To alleviate this challenge, different solutions have been proposed from different perspectives: Method engineering provides approaches that help the development or adaptation of existing methodologies to the problem domain [15–21]. It targets the development of methodologies for information system development. Maiden and Rugg [22] proposed a framework which provides general help for requirements engineers to select methods for requirements acquisition. Hickey and Davis [23,24] proposed a fine-grained selection model that helps understand the elicitation process and the selection of elicitation techniques. Bickerton and Siddiqi [25] proposed a framework for the classification of RE techniques built on the social assumptions made about organizations. Based on the identified needs for RE techniques that support the RE process, human communication, knowledge development, documentation and management, Macaulay [12] proposes a wish-list for RE techniques with the aim of identifying what kind of techniques are needed. Kotonya and Sommerville [26] proposed high level, general attributes which can be used for the evaluation and selection of RE techniques. Davis [27] proposed a process for RE methods selection that is based on four strategies of the requirements determination model and focuses on the selection of RE elicitation techniques. Browne and Ramesh [28] also proposed an idea for requirements elicitation technique selection. This technique selection method is built on the human cognitive model. Lobo and Arthur [29] developed an approach for technique selection that is based on a predefined RE process model. Technique analysis and selection is done manually and at a high level to fit the objectives of the activities within the predefined model. In [30], Lausen discusses several techniques, which can be used in the RE process, and various requirements presentation styles in detail. He then briefly explains an idea for RE technique selection using a matrix that contains with RE techniques and objectives that needed to be addressed in an RE process.

Despite the various research efforts to provide support for RE technique selection, the following problems are not yet adequately addressed:

- There is little research into the identification of technique attributes. Such attributes are essential for the comparison of techniques.
- Most existing research classifies RE techniques only at a high level of abstraction, which is not sufficient in order to determine how a technique can be used in the different phases of the RE process. Additionally, it does not allow the detailed comparison of techniques. We believe that an attribute-based comparison provides more suitable information for the selection of techniques. For example, if two techniques t_1 and t_2 have similar functionality and both techniques are deemed suitable for a particular project, then the less complex technique might be the preferred candidate if the project has significant time constraints. On the other hand, the more complex technique might be a better choice if the quality of the requirements is paramount and the technique offers additional means that help improve requirements. The idea of using more complex techniques in situations that require high-quality specifications is consistent with Davis' techniques contingency model [27].
- Most current research into technique selection focuses only on one individual phase of the overall RE process, such as requirements elicitation, documentation, etc. So far we have not found any other approach that provides support for the selection of RE techniques for the whole RE process.

Based on the problems identified, a methodology[1] for Requirements Engineering Techniques Selection,[2] called MRETS, was developed to help find a set of RE techniques that is suitable for a specific project based on the project's attributes. This would meet the critical need of industry for advice on how and when to use already existing RE approaches [31,32]. Ideally, the methodology can be used for all types of software projects; however, it fits best to projects that use a complete RE process. Currently, support for the selection of RE techniques for component-based and COTS-based software development is still weak and subject to further investigation. Additionally, tailoring and customizing

[1] It is worth mentioning that the terms "methodology", "method" and "technique" are used in several different ways in literature [33]. Some researchers argue that "technique" is the only proper word to describe existing methods and methodologies in the software domain [13,26,34–37]. We use the definition given in [38,39] which describes a "methodology as a set of guidelines that prescribes behavior in order to think and act in a situation". Specifically, it means a sequence of systematic steps that aid RE process development. The MRETS methodology imposes a disciplined process upon RE technique selection in order to find the best techniques for a software project.

[2] In this paper, techniques selection refers to the selection of techniques for each individual phase of the RE process, i.e. it addresses all phases of the RE process rather than only one phase.

existing techniques to specific projects have not yet been covered.

In order to provide a unified framework in the research for RE techniques selection, we use the RE process model proposed by Kotonya and Sommerville [26] and adapted it to our specific needs. Kotonya and Sommerville's RE process model is a widely accepted and referenced model and consists of four phases: requirements elicitation, requirements analysis and negotiation, requirements documentation, and requirements validation. In this model, requirements management is part of all the four phases. In our research, we classified RE techniques into the following five categories: requirements elicitation techniques, requirements analysis and negotiation techniques, requirements documentation techniques, requirements verification and validation techniques, and requirements management techniques. Requirements management tools are considered compulsory and are automatically included in all recommendations since research has shown that more than 60% of successful software projects benefited from using requirements management tools during the RE processes [40]. However, the detailed evaluation and selection of requirements management tools is not included in this paper and is subject to further research.

The proposed MRETS methodology has several major differences compared to the other approaches discussed in the previous paragraphs: (1) MRETS is based on the detailed analysis and comparison of RE techniques using our evaluation schema and a clustering method that is widely used in other domains. The evaluation and clustering of techniques provides valuable information related to the similarity of techniques based on a given set of attributes. (2) Knowledge of and experiences with RE techniques described in literature as well as from experts were elicited, documented and used during the research. (3) MRETS helps requirements engineers in establishing a link between the attributes of the project and the attributes of RE techniques. The major contribution of the new methodology is that it provides detailed guidelines that help requirements engineers select the most appropriate techniques for a given project. This has been demonstrated in an industrial case study.

One of the fundamental assumptions behind MRETS is that appropriate use of RE techniques leads to high-quality requirements specifications, which in turn results in high-quality software products. This assumption is generally accepted in the RE community [13,24] and the foundation of all RE process improvement initiatives. Nevertheless, its correctness has to the best of our knowledge not yet been formally proven.

The rest of the paper is organized as follows: Sect. 2 gives an introduction into the techniques analysis and project analysis that was conducted during the research, the results of which serve as the foundation for developing the methodology. The methodology for technique selection is presented in Sect. 3. An industrial case study is described in Sect. 4. The major findings, conclusions and future work are discussed in Sect. 5.

2 Analysis of RE techniques and software projects

In our earlier research, we identified a set of 46 RE techniques which were analyzed in detail (see Table 1) [7,14,41]. There are other techniques that will be included in our future research. However, the currently selected set contains representative RE techniques that cover all phases of the RE process. Their selection was based on the following considerations:

- Maturity of a technique. We consider a technique to be mature if it has well-defined systematic steps or has a well-defined collection of notations, is well-organized and documented, and has been used in industrial projects. Focusing on techniques that meet these criteria enables us to have sufficient information to analyze the techniques. Furthermore, we wanted to consider techniques that are mature enough for industrial use.
- Coverage of all four phases of the RE process. We selected techniques that adequately cover all phases of the RE process.
- Industrial awareness and experience with the technique. In order to get technique evaluations from industry we selected techniques that are at least partially known in industry. This allows us to have academics as well as practitioners score the techniques with respect to the technique attributes we defined.
- Scope and sources of techniques. All techniques were analyzed and evaluated from an RE perspective rather than from an overall software lifecycle perspective. For example, object-oriented analysis (OOA) can be used in several lifecycle phases. We, however, looked at it only from an RE perspective, i.e., its use for object-oriented (OO) requirements modeling to increase understanding. Additionally, since many techniques have several variants, each technique in Table 1 refers to a specific source and version. Detailed references for each techniques presented in Table 1 can be found in [33]. For example, OOA in Table 1 refers to requirements modeling with Booch's OO technique. This version was selected since it emphasizes OO domain analysis which is highly relevant for RE [42]. The basic rationale for assigning a technique to a specific RE phase was based on the following: (1) the major role of a technique as defined in its original documents, (2) the classification of a technique in the research literature [2], and (3) information collected from industry experts about the major purpose of a technique when

Table 1 Summary of analyzed RE techniques

ID	Technique name	Most common area of application in the RE process
1	Brain storming and idea reduction	Elicitation
2	Designer as apprentice	Elicitation
3	Document mining	Elicitation
4	Ethnography (observation is one of its kind)	Elicitation
5	Focus group	Elicitation
6	Interview	Elicitation
7	Contextual inquiry	Elicitation
8	Laddering	Elicitation
9	Viewpoint-based elicitation	Elicitation (later stage)
10	Exploratory prototypes (throw-away prototype)	Elicitation, Analysis and negotiation, Verification and validation
11	Evolutionary prototypes	Elicitation, Analysis and negotiation, Verification and validation
12	Viewpoint-based analysis	Analysis and negotiation
13	Repertory grids	Elicitation
14	Scenario-based approach	Elicitation (later stage), Analysis and negotiation, Documentation, Verification and validation
15	Joint application design (JAD)	Elicitation
16	Soft systems methodology (SSM)	Elicitation
17	Goal-oriented analysis	Analysis and negotiation
18	Viewpoint-based documentation	Documentation
19	Future workshop	Elicitation
20	Representation modeling	Analysis and negotiation, Elicitation
21	Functional decomposition	Analysis and negotiation
22	Decision tables	Analysis and negotiation, Documentation, Verification
23	State machine	Analysis and negotiation, Documentation, Verification
24	State charts (also known as state diagrams)	Analysis and negotiation, Documentation, Verification
25	Petri-nets	Analysis and negotiation, Documentation, Verification
26	Structured analysis (SA)	Analysis and negotiation, Documentation, Verification
27	Real time structured analysis	Analysis and negotiation, Documentation, Verification
28	Object-oriented analysis (OOA)	Analysis and negotiation, Documentation, Verification
29	Problem frame oriented analysis	Analysis and negotiation, Documentation, Verification
30	Goal-oriented verification and validation	Verification and validation
31	Entity relationship diagram (ERD)	Documentation, Analysis and negotiation
32	AHP	Requirements Prioritization
33	Card sorting	Requirements Prioritization
34	Software QFD (SQFD)	Analysis and negotiation, Elicitation
35	Fault tree analysis	Analysis and negotiation, Elicitation
36	Structured natural language specification	Requirements Documentation
37	Viewpoint-based verification and validation	Verification and validation
38	Unified modeling language (UML)	Documentation, Analysis and negotiation, Verification
39	Z	Documentation, Analysis, Verification
40	LOTOS	Documentation, Analysis, Validation
41	SDL	Documentation, Analysis, Validation
42	Extreme programming (XP)	Elicitation, Analysis and negotiation, Documentation, Validation
43	Formal requirements inspection	Verification and validation
44	Requirements testing	Verification and validation
45	Requirements checklists	Verification and validation
46	Utility test	Verification and validation

used in an industrial setting. Additionally, it is also worth to mention that a technique assigned to one phase might also be used in other phases.

It has to be mentioned that the granularity of the RE techniques presented in Table 1 is not the same. This was deliberately done in order to help the developer select the most suitable technique for the task at hand. The difference of granularity of techniques makes the research more applicable to the large diversity of real life situations.

As has been discussed above, the overall objective of this research is to provide decision support for the selection of RE techniques for a given project by considering the attributes of the techniques and the project under development. In order to achieve this objective, two major tasks were carried out: RE techniques analysis and software project analysis. They will be discussed in the following sub-sections.

2.1 RE techniques analysis

In order to provide support for selecting RE techniques, the techniques first have to be analyzed in detail. RE techniques analysis includes the following sub-tasks:

(1) Identification of the attributes of RE techniques.
(2) Evaluation of the RE techniques. Each technique is rated against 31 attributes by researchers and experts from both industry and academia.
(3) Analysis of the techniques using a clustering method.

The following summarizes the results of this analysis:

2.1.1 Attributes of RE techniques

Classification of RE techniques requires the definition of attributes that characterize these techniques. Without proper characterization of the techniques, no effective support for the selection of techniques can be provided. The 46 RE techniques analyzed and compared during this research are listed in Table 1 together with the major phases in which these techniques can be applied.

Based on our analysis, 31 attributes for RE techniques were defined (see Table 2) through a process of abstraction, analysis, synthesis, characterization of research and literature, and validation by a group of experienced software engineers. The first column in Table 2 contains the categories of the attributes which correspond to the four phases of the RE process. The third column lists the actual attributes. Each attribute is defined with a list of criteria to ensure its measurability [33]. An ordinal scale is used for all attributes, i.e. the attribute values are set as none (or "not relevant"), very low, low, medium, high and very high. The interested reader can refer to [14] for more details.

The derivation and justification of these attributes is as follows:

- The attributes in the schema were developed based on the analysis and synthesis of the characteristics of RE techniques as well as schemata proposed by other researchers [12,22,23,26]. For example, the attributes "Ability to facilitate communication", "Ability to represent requirements" and "Maturity of supporting tool" are derived from the technique properties "Scope for communication", "Precision of definition of its notation", "Tool support" proposed by Kotonya and Sommerville [26].
- Inclusion of an attribute in our schema is based on its simplicity, completeness, suitability, applicability and effectiveness in helping the classification and selection of techniques. For instance, some RE elicitation techniques have well-defined procedures for identifying non-functional requirements (NFRs). The identification of NFRs is essential during the RE process and therefore the attribute "Ability to help identify non-functional requirements" is included in the assessment schema of RE techniques. However, since identifying functional requirements is a fundament aspect of pretty much all RE elicitation techniques, we did not include in our assessment schema an attribute, such as "Ability to help identify functionl requirements". Such on attribute would not contribute to the differentiation between techniques.
- The attributes reflect the major COncern of RE process (COREs) model proposed in our previous research [43]. Each major concern is a specific interest or objective of the RE process which needs to be addressed [31,43]. The major COREs model is an RE process assessment model that also serves as guideline for RE process development. It was developed based on theoretical research as well as surveys of software practitioners. The attributes listed in Table 2 are closely related to most of the major COREs, thus providing support for the evaluation of RE techniques.

The schema is consistent with existing schemata proposed by other researchers such as Macaulay's schema [12] and Kotonya and Sommerville's schema [26] (see discussion in Sect. 1) but is more comprehensive.

As can be seen in Table 2, the attributes provide a means to measure different facets of an RE technique. The attributes can therefore be divided into two categories:

- Attributes that describe the ability of a technique (attributes 1–28): The higher the value of an attribute, the more suitable is the technique for addressing this attribute.
- Attributes that describe economic factors (attribute 29–31): The higher the value of an attribute, the higher the cost of using the technique.

Table 2 A proposed classification schema for RE techniques and their assessment

Categories	No.	Attributes of the techniques	Exploratory prototypes (throw-away prototype)	JAD	Functional decomposition	State charts (also known as state diagrams)	AHP	XP techniques
Elicitation	1	Ability to facilitate communication	0.8	1	0	0	0.6	1
	2	Ability to help understand social issues	0.2	1	0	0	0	0.4
	3	Ability to help get domain knowledge	0.4	0.6	0	0	0	0.4
	4	Ability to help get implicit knowledge	0.2	0.2	0	0	0	0
	5	Ability to help identify stakeholders	0	1	0	0	0	0.4
	6	Ability to help identify non-functional requirements	0	0.8	0	0	0	0
	7	Ability to help identify viewpoints	0	1	0	0	0	0
Analysis and negotiation	8	Ability to help model and understand requirements (both general and domain specific requirements)	1	0	0.8	1	0	0.6
	9	Ability to analyze and model requirements with understandable notations	0	0	1	0.8	0	0.8
	10	Ability to help analyze non-functional requirements	0	0	0.2	0	0	0
	11	Ability to facilitate negotiation with customers	0.8	0	0.4	0.4	0.6	1
	12	Ability to help prioritize requirements	0	0	0.2	0	1	1
	13	Ability to help identify accessibility of the system	0.8	0	0.6	0.6	0	0.2
	14	Ability to help model interface requirements	1	0	0.2	0.6	0	0.8
	15	Ability to help identify reusable requirements and support requirements reuse	0	0	0	0	0	0
Documentation and notation	16	Ability to represent requirements (expressibility)	1	0	0.8	1	0	0.2
	17	Ability to help requirements verification	0	0	0.2	0.8	0	0.4
	18	Completeness of the semantics of the notation	0	0	0.6	0.6	0	0
	19	Ability to help write unambiguous and precise requirements by using the notation	0	0	0.6	0.8	0	0.2
	20	Ability to help write complete requirements	0	0	0.6	0.6	0	0.2
	21	Ability to help with requirements management	0	0	0.6	0	0	0.4
	22	Ability to help design highly modular systems	0	0	0.6	0	0	0
	23	Implementability (executability) of the notation used	0	0	0	0	0	0
Verification and validation	24	Ability to help identify ambiguous requirements	0	0	0	0.6	0	0.6
	25	Ability to help identify interactions (inconsistency, conflict)	0	0	0	0.2	0	0.2
	26	Ability to help identify incomplete requirements	0	0	0	0	0	0
Other aspects	27	Ability to support COTS-based RE processes	0	0	0	0	1	0
	28	Maturity of supporting tool	0.8	0.4	0.8	0.6	1	0.4
	29	Learning curve (Introduction cost)	0.2	0.6	0.4	0.6	0.6	0.4
	30	Application cost	1	0.6	0.2	0.6	0.4	0.2
	31	Complexity of technique	0.2	0.2	0.2	0.4	0.4	0.2

This classification is essential for the calculation of the abilities and cost of RE techniques. Details of the usage of the schema will be discussed in Sect. 3.

It has to be pointed out that most existing techniques only address some of the attributes listed in Table 2. This is the reason why requirements engineers often combine several

RE techniques in order to adequately address the different attributes they are concerned about.

The currently proposed list of the RE techniques attributes worked very well in our research since it provides a foundation for RE techniques assessment and analysis, and was very helpful when it came to selecting the best combination of RE techniques for a certain project [33]. However, application of MRETS to more industrial projects and the inclusion of additional RE techniques will likely result in further refinement of the list of RE attributes.

2.1.2 Techniques evaluations

The following procedure was used to assess RE techniques:

- Assessment of each technique against each attribute in the schema. This step, in turn, is composed of the following sub-steps:
    - RE researchers assess the techniques. Three researchers with 5–10 years experience in RE were involved in the research to assess the RE techniques. In situations where the researchers differred in their assessment, results were discussed and if necessary an aggregation method called arithmetic means was used to resolve the disagreement [44, 45].
    - Industry experts assess the techniques. The final assessment results of the researchers were given to two industry experts who reviewed and validated the results. Both of the two industry experts have more than 10 years of working experience and they were involved in more than 32 and 27 software projects, respectively. They disagreed in about 11 percent (155 out of 1,426) of the assessment results with the three researchers. Disagreements between researchers and industry experts were resolved through discussion or if no consensus could be reached, through the weighted average aggregation method [44]. A higher weight was given to the industry experts to ensure industrial relevance of our work.

It is worth to mention that only 46 of the 56 initially identified techniques were assessed as the remaining 10 techniques are currently not adequately documented. The assessment of more techniques is part of our further research.

- The assessment results were normalized. Since all attributes use the same measurement scale, the normalization of the assessment results is done using the following formula:

$$Z_i^{(j)} = \frac{r_i^{(j)} - 1}{M_i - 1}, \quad \text{where } M_i = 6 \quad \text{for all } i,$$
$$r^{(j)} \in \{1, \ldots, 6\}, \quad j = 1, \ldots, 6, \quad i = 1, \ldots, 31$$

In this formula, 6 represents the number of the ordinal measurements, 31 is the number of the attributes defined. Sample data from the overall dataset is shown in Table 2. This table contains the normalized results of the assessment of six techniques. For example, the ability to facilitate communcation of JAD techniques is assessed as "very high". The normalized value for "very high" is 1, i.e., the entry in that column is 1. This data set is the foundation for the further analysis of RE techniques.

2.1.3 Techniques clustering

Clustering has been used extensively as a data analysis technique in various domains, such as medical data analysis, data mining, and market analysis [46]. Cluster analysis organizes data by abstracting the underlying structure either as a group of individuals or as a hierarchy of groups [47]. This means that clustering techniques allow objects with similar attributes to be organized into groups.

There are a number of clustering methods, such as K-Means, Hierarchy Clustering and Fuzzy Clustering. Different methods are applicable to different situations. Due to the uncertain nature of empirical and experiential data, the Fuzzy Clustering method is used in this research to analyze the similarity between RE techniques.

The basic principle of the Fuzzy Clustering algorithm is to partition n objects into p clusters by minimizing the following cost function [48,49]:

$$\text{Cost} = \sum_{i=1}^{p} \sum_{j=1}^{n} u_{ij}^2 d_{ij}^2 \qquad \text{(C-1)}$$

where

$d_{ij} = \|W_j X_j - m_i\|$ is the distance between each object X_j and the cluster centroid m_i, $j = 1, \ldots, n$; n is the number of objects, $i = 1, \ldots, p$; p is the number of clusters.
m_i is a vector representing the centroid of cluster i.
u_{ij} is the degree of membership of object j in cluster i.
W_j is the weight for each object. $W_j = [1, \ldots, 5]$

The cost function is minimal if

$$m_i = \frac{\sum_{j=1}^{n} u_{ij}^2 W_j X_j}{\sum_{j=1}^{n} u_{ij}^2} \quad \text{and} \quad u_{ij} = \frac{1}{\sum_{k=1}^{p} \left(\frac{d_{ij}}{d_{kj}}\right)^2} \qquad \text{(C-2)}$$

In order to examine the overall feasibility of RE techniques clustering and to identify the optimum number of technique clusters, 46 RE techniques were clustered using 9 different settings as shown in the Table 3. The aim is to use that number of clusters that ensures that the cost function is minimized

Table 3 Clustering setting and performance

Setting / Performance using the setting	Number of clusters=4		Number of clusters=5		Number of clusters=6		Number of clusters=7		Number of clusters=8		Number of clusters=9		Number of clusters=10		Number of clusters=11		Number of clusters=12	
	All weights are 1 (W_i=1)	Various weights	All weights are 1 (W_i=1)	Various weights	All weights are 1 (W_i=1)	Various weights	All weights are 1 (W_i=1)	Various weights	All weights are 1 (W_i=1)	Various weights	All weights are 1 (W_i=1)	Various weights	All weights are 1 (W_i=1)	Various weights	All weights are 1 (W_i=1)	Various weights	All weights are 1 (W_i=1)	Various weights
Value of the cost function	27.12	12.96	25.43	10.65	21.78	6.18	18.81	5.11	17.55	3.60	17.92	3.64	18.12	5.50	19.55	5.87	19.38	6.81
Clustering effects	Basic	Basic	Basic	Basic	Better	Better	Better	Better	Best	Best	Best	Best	Better	Better	Better	Better	Better	Better

Notes: 1. The values of the cost function are calculated based on formula (C-1)

2. In the row containing the Clustering effects

"Basic" indicates that:

1) the value of the cost function is very high compared to the other clustering results.
2) the result of the clustering gives only a rough indication of the high level cluster to which a RE technique belongs.
3) the techniques in each cluster do not have very similar characteristics.

"Better" indicates that the result of the clustering provides better classification results compared to the "Basic" result, yet not as good as the "Best" results defined below.

"Best" indicates that

1) the value of the cost function is amongst the lowest values of all the other clustering results.
2) the result of the clustering provides a fine-grained classification of the RE techniques.
3) The techniques in each cluster exhibit very similar characteristics as can be verified by RE experts.

3. The "weight" refers to the weight of each attribute of the techniques; "various weights" indicates that the attributes were assigned different weights based on the characteristics of the project.

4. It has to be noted that the observation of the result is independent of the assigned weights.

and the techniques in each cluster have similar characteristics. This number can be used as a recommendation when applying MRETS.

The results of the nine different clustering trials can be seen in Table 3. The number of clusters goes from 4 to 12, and in each setting, two calculations were performed. The first one assumed the same weight for all attributes ($W_i = 1$); the second calculation assumed different weights for the attributes. For example, assume that for a certain project the requirements engineers are very interested in the following set of technique attributes: Ability to facilitate communication, Ability to help identify viewpoints, Ability to facilitate negotiation with customers, Ability to help prioritize requirements, Ability to help model interface requirements. Therefore, these attributes have a weight of 5 (highest); other attributes are given a weight of 1 (lowest). We then examine the clustering results to see which of the nine different settings provides the lowest cost function and the most cohesive clustering.

Based on the results of the calculations shown in Table 3, the following observations can be made:

- The higher the number of clusters, the more fine-grained is the classification of the techniques. Further analysis shows that the classification into 8 or 9 clusters maximizes the cohesiveness of the techniques in each cluster and minimizes the cost function. Two examples with the total number of clusters being 4 and 8 are shown in Tables 4 and 5, respectively. As can be seen, the techniques in each cluster in Table 5 have a higher similarity than those in Table 4. For example, cluster 2 in Table 4 contains requirements elicitation techniques. These techniques are further decomposed into clusters 4, 6 and 8 in Table 5 with cluster 4 containing group session techniques for requirements elicitation, cluster 6 containing knowledge elicitation techniques, and cluster 8 containing requirements elicitation techniques that consider the social context. Therefore, the boundary between techniques in different clusters is clearer using 8 clusters (Table 5) rather than 4 clusters (Table 4).
- Clustering also has its challenges. Some techniques were classified into an unsuitable cluster (e.g. in Table 5, Document Mining is a requirements elicitation technique; yet it was included in cluster 7 where most techniques are requirements verification techniques). Such misplacements might be due to the following reasons:
  - Some techniques might not be well-understood.
  - Some techniques might not be used appropriately in practice. An interesting discussion related to the misuse of RE techniques can be found in [25].
  - The Fuzzy Clustering algorithm is limited in its capabilities as it is based on a mathematical model and therefore is in some respect inflexible and rigid. Further research is required to improve the Fuzzy Clustering algorithm and adapt it to the specific purpose of our research so that it will work more reliably.
  - Some techniques simply do not fit well into any of the clusters. For instance, Document Mining does not fit well into any cluster. Its highest membership value (0.23) is in cluster 7, which is still very low compared to other techniques.
- The benefits of clustering are:
  - Clustering provides a mechanism to group RE techniques according to their characteristics. Similar techniques are grouped into the same cluster thus providing a foundation for the analysis of the similarity of various techniques. It can therefore be used for the selection of RE techniques.
  - The analysis results spark discussion about RE techniques. Further analysis based on clustering could provide information for the development of guidelines for the use of RE techniques.

Based on the analysis of the clustering result of the RE techniques in our experiments discussed above, we found that some RE techniques complement each other. Although they have very different objectives they can be used together and the shortcomings of one technique are augmented by the features of another technique. We also found that some RE techniques are functionally comparable to each other as they have very similar functions and outcome [14,33]. We therefore developed two concepts to describe the relationship of techniques: functionally comparable techniques and functionally complementary techniques. These two concepts are described in Table 6.

2.2 Analysis of software projects

The second task focused on identifying attributes of software projects and their relevance for the selection of RE techniques based on empirical research as well as literature [14,41,50]. The task included:

1. Identification of the attributes of software projects.
2. Identification of the techniques that have been used in previous software projects or could be good candidates for a certain type of software project. This allows the definition of rules that guide the selection of RE techniques.

The results of these two tasks are fundamental to the proposed methodology and will be summarized in the following subsection. The interested reader can refer to [14,41,50] for more detailed results.

Section 2.1 focused on the analysis of RE techniques. This section now defines attributes of software projects and their relevance for the selection of RE techniques. The aim is to

Table 4 Clustering result with four clusters

Clusters	Membership value of each technique within each cluster	Name of technique
Cluster 1	0.49609	OOA
	0.48749	ERDs (Entity relationship diagram)
	0.48272	Functional decomposition
	0.44514	Decision tables
	0.4204	Fault tree analysis
	0.4179	Structured natural language specification
	0.41291	Representation modeling
Cluster 2	0.8065	Interview
	0.77552	Contextual inquiry
	0.67396	Brain storming and idea reduction
	0.65252	Future workshops
	0.64511	Focus group
	0.63023	JAD
	0.62024	Designer as apprentice
	0.60554	SSM
	0.54303	Ethnography (observation)
Cluster 3	0.67276	State charts (or state diagrams)
	0.67239	State machine
	0.57052	Petri-nets
	0.54265	Real-time structured analysis
	0.4736	Structured analysis (SA)
	0.44644	Z
	0.44644	LOTOS
Cluster 4	0.42933	SDL
	0.44642	Viewpoint-based verification and validation
	0.38335	Requirements checklists
	0.37768	Formal requirements inspection
	0.37768	Requirements testing
	0.37132	Utility test
	0.36391	Card sorting
	0.35924	Repertory grids

derive rules that show which RE techniques are suitable for a given project.

2.2.1 Attributes of software projects

We defined 21 software project attributes. The project attributes play a key role when it comes to project characterization and selection of RE techniques. Each of these attributes has been defined in detail in [41]. In order to shorten the length of the paper, only the most important attributes are briefly described here:

Project size: This attribute is defined as the size of project (X) in terms of number of requirements. The requirements refer to atomic requirements which are defined as indivisible, "well-formed" requirements [51]. Possible values for this attribute are: very small ($X < 100$), small ($100 \leq X < 500$), medium ($500 \leq X < 1,000$), big ($1,000 \leq X < 4,000$), and very big ($X \geq 4,000$ requirements).

Requirements volatility: This attribute is defined as the percentage of requirements that change throughout the development of project (Y). The attribute can have the following values: very low ($Y < 1\%$), low ($1\% \leq Y < 10\%$), medium ($10\% \leq Y < 30\%$), high ($30\% \leq Y < 50\%$), and very high ($Y \geq 50\%$).

Project category: This attribute defines the type of project. Possible values are: communication, embedded, semi-detached and organic. Some of these values are borrowed from the COCOMO model [1].

Degree of safety criticality: This attribute is defined as the degree of safety required by the system about the loss of

Table 5 Clustering result with eight clusters

Clusters	Membership value of each technique within each cluster	Name of technique
	0.97674	Z
	0.97674	SDL
Cluster 1	0.95298	LOTOS
	0.53364	Petri-nets
	0.73222	Fault tree analysis
	0.68106	Functional decomposition
Cluster 2	0.52614	Decision tables
	0.40634	Entity relationship diagram (ERDs)
	0.40236	Scenario approach
	0.89187	Evolutionary prototypes
	0.78803	Exploratory prototypes (throw-away prototype)
Cluster 3	0.54042	Representation modeling
	0.21519	Extreme programming (XP)
	0.69676	JAD
	0.63824	Focus group
Cluster 4	0.54191	Future workshops
	0.48897	SQFD
	0.84888	Real time structured analysis
Cluster 5	0.84551	State machine
	0.84341	State charts (also known as state diagrams)
	0.70652	Repertory grid
	0.64585	Laddering
Cluster 6	0.44322	Brain storming and idea reduction
	0.34511	Designer as apprentice
	0.32611	Interview
	0.96205	Requirements checklists
	0.95975	Formal requirements inspection
Cluster 7	0.95975	Requirements testing
	0.23787	Document mining
Cluster 8	0.38085	Ethnography (observation)
	0.3171	Contextual inquiry

human life or property. The values are defined as follows: very low, low, medium, high, and very high.

Similarly, we define other attributes: *Product Type, Team Size, Project Complexity, Time Constraints, Organization and Customer Relationship* [12], *Acquaintance with the Domain, Product Quality Criteria, Cost Constraints, Knowledge of RE of the Teams, Degree of Knowledge of Requirements, Availability of a Skilled Facilitator, Stakeholder Heterogeneity, Degree of Innovation of the Project, Customer Availability, Degree of the Importance of Reusability, Degree of the Importance of Eliciting Implicit Knowledge, Degree of Outsourcing*. The definitions of these attributes can be found in [41]. All these attributes provide means to describe the characteristics of a software project. Based on the values of these project attributes and the techniques attributes, the requirements engineers can understand the most desirable techniques' abilities required by the software project in order to address the specific issues of the software project. This will, in turn, help to select the most suitable RE models and techniques for the given project.

2.2.2 The derivation of rules

In order to facilitate RE process development, a set of rules stating what process models and techniques are the most suitable for a certain type of project were developed based

Table 6 Formal definition of "functionally comparable" and "functionally complementary"

	Functionally comparable techniques	Functionally complementary techniques
Definition	Two techniques t and t' are defined as functionally comparable if and only if t and t' are in the same cluster and the differences of their attributes' values are within a specified range, i.e. $\lvert t(j) - t'(j)\rvert \leq \varepsilon$ for all $j = 1, \ldots, 31$, where ε is a project dependent value, normally $\varepsilon \leq 0.4$. The functionally comparable techniques of t can be written as a function F(t), F: $\mathbf{T} \rightarrow \mathbf{P}(\mathbf{T})$, $F(t) = \{t' \mid t' \in \mathbf{T} \wedge \lvert t(j) - t'(j)\rvert \leq \varepsilon\}$ for $j = 1, \ldots, 31$	Two techniques t and t' are defined as functionally complementary if and only if t and t' are not in the same cluster and $0.4 < \left\lvert \sum_{k=1}^{31} (t(k) - t'(k)) \right\rvert \leq \varepsilon$. Our research determined that $\varepsilon \leq 0.8$ is a suitable value for ε [15,55]. The functionally complementary techniques of technique t can be written as a function C(t), where C: $\mathbf{T} \rightarrow \mathbf{P}(\mathbf{T})$, $C(t) = \{t' \mid t' \in \mathbf{T} < \left\lvert \sum_{k=1}^{31} (t(k) - t'(k)) \right\rvert \geq \varepsilon\}$
Semantics	The semantics of the functionally comparable techniques states that the techniques are functionally similar about the attributes defined in our research [15,55]. If two RE techniques are functionally comparable, the major functions of these two techniques are very similar. The condition "in the same cluster" ensures the two techniques are functionally similar.	The semantics of the functionally complementary techniques states that the techniques are functionally not the same, but complementary to each other. If two RE techniques are complementary, the advantage of one technique is the weakness of the other. The condition "not in the same cluster" ensures the two techniques are not functionally similar, but complementary.
Example	For example, Z and SDL are *functionally comparable techniques*as both techniques can support formal requirements documentation and verification. This is also supported mathematically, as $\text{Max}\lvert Z(j) - \text{SDL}(j)\rvert = 0.4$ and $0.4 \leq \varepsilon = 0.4$ for all $j = 1, \ldots, 31$	For example, Ethnography (observation) and Interview are complementary techniques, because observation is good at eliciting implicit knowledge and the overall behavior of the system, but not good at identifying stakeholders, future requirements and the data of the system. Interviewing, on the other hand, is good at doing these. Therefore, these two techniques are functionally complementary.
Property	Functionally comparable techniques satisfy the commutative law, but do not satisfy the transitive law. For example, if t_i is a functionally comparable technique to t_k, then t_k is also a functionally comparable technique to t_i; however, if t_i is a functionally comparable technique to t_j, and if t_j is a functionally comparable technique to t_k, then it is not necessarily true that t_i is a functionally comparable technique to t_k.	Complementary techniques satisfy the commutative law, but not the transitive law. For example, if t_i is a functionally complementary technique to t_k, then t_k is also a functionally complementary technique to t_i; however, if t_i is a functionally complementary technique to t_j, and if t_j is a functionally complementary technique to t_k, then it is not necessarily true that t_i is a functionally complementary technique to t_k.

Meaning of the notations:
T be the set of all existing RE techniques t; **P**(**T**) denotes the power set of **T**
t, t' and t_i denote individual RE techniques, currently, $i = 1, \ldots, 46$; $A_1, A_2, \ldots,$ An are the attributes of t, $A_i \in \mathcal{A}$, $i = 1, \ldots, 31$
$\mathcal{A}$ represents the set of all currently identified attributes of RE techniques; $\mathcal{A}$ is defined as a real number in the range of [0, 0.2, 0.4, 0.6, 0.8, 1] or $\mathcal{A} = [0, 0.2, 0.4, 0.6, 0.8, 1]$
$a_{i,1}, a_{i,2}, \ldots, a_{i,m}$ are the values of the attributes for techniques ti, $t_i = \langle a_i, 1, a_i, 2, \ldots, a_i, m\rangle$; $t_i(j) = a_{i,j}$; $i = 1, \ldots, 46$, $j = 1, \ldots, 31$. In this case **T** can be written as $\mathbf{T} = A_1 \times A_2 \times \ldots, \times A_m$

on a survey of five practitioners who have worked on more than 20 software projects in the past 14 years, as well as three experts in academia who have both research and industrial experience for more than 10 years. Some rules were directly derived from past projects; others were derived from the predictive judgment received from different experts. For example, if expert A recommends a set of techniques $T_i = \{t_1, t_2\}$ for project Pr_i and expert B recommends a set of techniques $T_j = \{t_1, t_3\}$ for the same project, then the final recommendation would be either $T_I \cup T_j$ if t_2 and t_3 are not mutually exclusive, or $T_i \cap T_j$ if t_2 and t_3 are mutually exclusive. Helmer and Rescher [52] state that in imprecise science where no accepted measurements are available the incorporation of expert opinion into the investigation structure of the subject area is acceptable and can also assure the validity of the result just as in other research. Based on this assertion, we argue that the process used in this research for the derivation of the recommendation rules from experts is acceptable.

All the derived rules are part of the RE process knowledge base (REPKB) and are used to help select the most suitable RE techniques for a given project. These so-called recommendation rules provide the initial recommendation given to the requirements engineers in the overall technique selection process. For example, when requirements engineers are faced with a new project, the initial recommendation of RE techniques, denoted as T_{IR}, can be retrieved based on the rules in REPKB and presented to the user with the help of a case-based reasoning mechanism.

Table 7 Examples of a recommendation rules

Rule No.	Attributes of Project and Product (Condition attributes)		Recommended Technique Sets (Decision attributes)
1	Project Size	Small	**Elicitation:** 1. XP (Customer Online) 2. Evolutionary Prototyping 3. Representation Modeling 4. Focus Groups **Analysis and Negotiation:** 1. OOA 2. AHP 3. UML **Documentation:** 1 UML 2. Structured Natural Language Specification **Verification and Validation:** 1. XP (Customer Online Inspection) 2. Formal Requirements Inspection
	Team Size	10	
	Requirements Volatility	High	
	Project Category	Organic	
	Degree of Safety Criticality	Low	
	Time Constraint	High	
	Product Type	New	
	…	…	
2	Project Size	Medium	**Elicitation:** 1. Interviewing 2. Focus Groups 3. Ethnography (Observations) **Analysis and Negotiation:** 1. SQFD 2. Viewpoint-Based Analysis 3. Scenario Approach **Documentation :** 1. SDL 2.Viewpoints-Based Specification **Verification and Validation:** 1. Formal Requirements Inspection 2. Scenario Approach 3. Requirements Testing
	Team Size	40	
	Requirements Volatility	Low	
	Project Category	Embedded	
	Degree of Safety Criticality	High	
	Time Constraint	Low	
	Product Type	New	
	…	…	
3	Project Size	Big	**Elicitation:** 1. Interviewing 2. JAD 3. Ethnography (Observations) **Analysis and Negotiation:** 1. Scenario Approach. 2. SQFD 3. OOA 4. ERD 5. Goal-Based Analysis. **Documentation :** 1. UML 2. SDL 3. Structured Natural Language Specification **Verification and Validation:** 1. Formal Requirements Inspection 2. Requirements Testing 3. Scenario Approach 4. Requirements Checklist
	Team Size	100	
	Requirements Volatility	Low	
	Project Category	Semi-Detached	
	Degree of Safety Criticality	High	
	Time Constraint	Low	
	Product Type	New	
	…	…	
……	…	…	…

The basic format of the techniques recommendation rules is illustrated in Table 7. The recommendations are based on the attributes of the given project. The techniques recommendation rules can be represented as $r_i^R\left(C_i^R, T_{IR(i)}\right)$, where C_i^R is a vector which includes a group of values of the attributes from a given project and $T_{IR(i)}$ is a vector which includes the recommended techniques based on C_i. For example, the first rule in Table 7 can be represented as $r_1^R\left(C_1^R, T_{IR(1)}\right)$; C_1^R ={Project Size = Small, Team Size = 10, Requirements Volatility = High, Project Category = Organic, Degree of Safety Criticality = Low, Time Constraints = High, Product Type = New}; and $T_{IR(1)}$ ={{ Customer Online, Prototyping, Concept Map, Focus Groups}, {OO Analysis, AHP, UML}, {UML-Based Specification, Structured Natural Language Specification}, {Customer Online Inspection, Formal Inspection}}. Currently, 36 rules have been developed in our research [33].

The initial recommendation is based on a case-based reasoning mechanism that uses previous experiences, which however may not be totally suitable for the given project. In this case, further refinement of the selection is needed. Requirements engineers will look for other functionally similar or complementary techniques to the techniques in T_{IR} which have a lower cost and complexity but can still meet the constraints of the given project.

3 The methodology

3.1 Notations

The following additional notations will be used in the remainder of the paper:

Let T_{RS} be a techniques recommendation space. A techniques recommendation space is defined as a set of techniques from which the final combination of recommended techniques can be derived. It is composed of:

- The initially recommended set of techniques T_{IR} for a given project. This recommendation is based on experience from previous projects.
- A set of prospective techniques recommended by the requirements engineers denoted as T_{ER}. The purpose of including these techniques in the recommendation space is to allow the requirements engineers more freedom in the decision making process.
- All the techniques that are functionally comparable or complementary to any of the techniques in $T_{IR} \cup T_{ER}$. The aim of including functionally comparable techniques in T_{RS} is to ensure that techniques with similar functionality, varying costs, and varying complexity are considered and the most suitable one is selected. The aim of including functionally complementary techniques in T_{RS} is to ensure that the overall ability of the recommended techniques is the highest and the combination of the RE techniques is adequately optimized.

Mathematically, the techniques recommendation space can be represented as:

$$\mathrm{T_{RS}} = T_{IR} \cup T_{ER} \cup \left\{t' | t' \in \mathrm{C}(t) \wedge t \in (T_{IR} \cup T_{ER})\right\} \\ \cup \left\{t'' | t'' \in \mathrm{F}(t) \wedge t \in (T_{IR} \cup T_{ER})\right\}.$$

where

$\mathrm{C}(t)$ and $\mathrm{F}(t)$ represent functionally complementary techniques and functionally comparable techniques to technique t, respectively.

Let $T_1, \ldots, T_n$ be the alternatively recommended techniques within $\mathrm{T_{RS}}$, i.e. $T_i \subset \mathrm{T_{RS}}, i = 1, \ldots, n$.

One of the major tasks of the approach is the selection of a T_i from $\mathrm{T_{RS}}$, such that T_i best fits the characteristics of the given project and maximizes the assistance to elicit, analyze, document, manage, verify and validate requirements.

T_i contains techniques of all four phases of the RE process, i.e. $T_i = T_e \cup T_a \cup T_d \cup T_v$ where T_e, T_a, T_d, T_v are requirements elicitation, requirements analysis and negotiation, requirements documentation, and requirements verification and validation techniques, respectively.

3.2 Objective function

The definition of the objective function that is used as criteria for RE techniques selection is of great importance. The objective of the selection process is to find that combination of techniques that has the maximum overall ability and minimal cost for a given project. The objective function focuses on the quality of requirements specifications, the complexity and cost of RE techniques. The techniques selected from the $\mathrm{T_{RS}}$ must meet the criteria set by the objective function [14,33]. The objective function is formally defined as follows:

$$\mathrm{F_C} : \mathbf{P}(\mathbf{T}) \rightarrow \mathrm{Real}$$

$$\mathrm{F_C}(T_i) = \sum_{t \in T_i} \mathrm{Ability}_t \quad \text{(O3.1)}$$

$$\mathrm{Ability}_t = \sum_{i=1}^{28} t(i) - 2 * (B * t(29) + t(30) + t(31)) \quad \text{(O3.2)}$$

$$T_C = \mathrm{Max}\ \mathrm{F_C}(T_i) \quad \text{for all } T_i \subset \mathrm{T_{RS}}, \quad \text{(O3.3)}$$

where

$\mathrm{F_C}$ represents the objective function.

Each T_i is a set of techniques and T_i is one of the technique combinations in $\mathrm{T_{RS}}$; each T_i shall include requirements elicitation (T_e), requirements analysis & negotiation (T_a), requirements documentation (T_d), and requirements verification & validation techniques (T_v); i.e. $T_i = T_e \cup T_a \cup T_d \cup T_v$;

$\mathrm{F_C}(T_i)$ represents the value of the objective function when the recommended RE techniques are T_i;

$\mathrm{Ability}_t$ indicates the normalized numerical value of the overall abilities of techniques t;

T_C denotes the recommended solution for the given project with the maximum value of $\mathrm{F_C}(T_i)$ among all T_is;

$t(1)$ to $t(31)$ represent the normalized numerical values of the RE technique attributes 1 through 31 (see Table 2); B is a coefficient that represents the requirements engineers' knowledge in the following way:

- B = 1, if requirements engineers do not know the technique at all, i.e., the introduction cost attribute of the technique needs to be taken into consideration.
- B = 0, if the requirements engineers have extensive knowledge about the technique, i.e., the introduction cost attribute does not need to be considered.
- Intermediate values are possible to allow for partial knowledge of a technique.

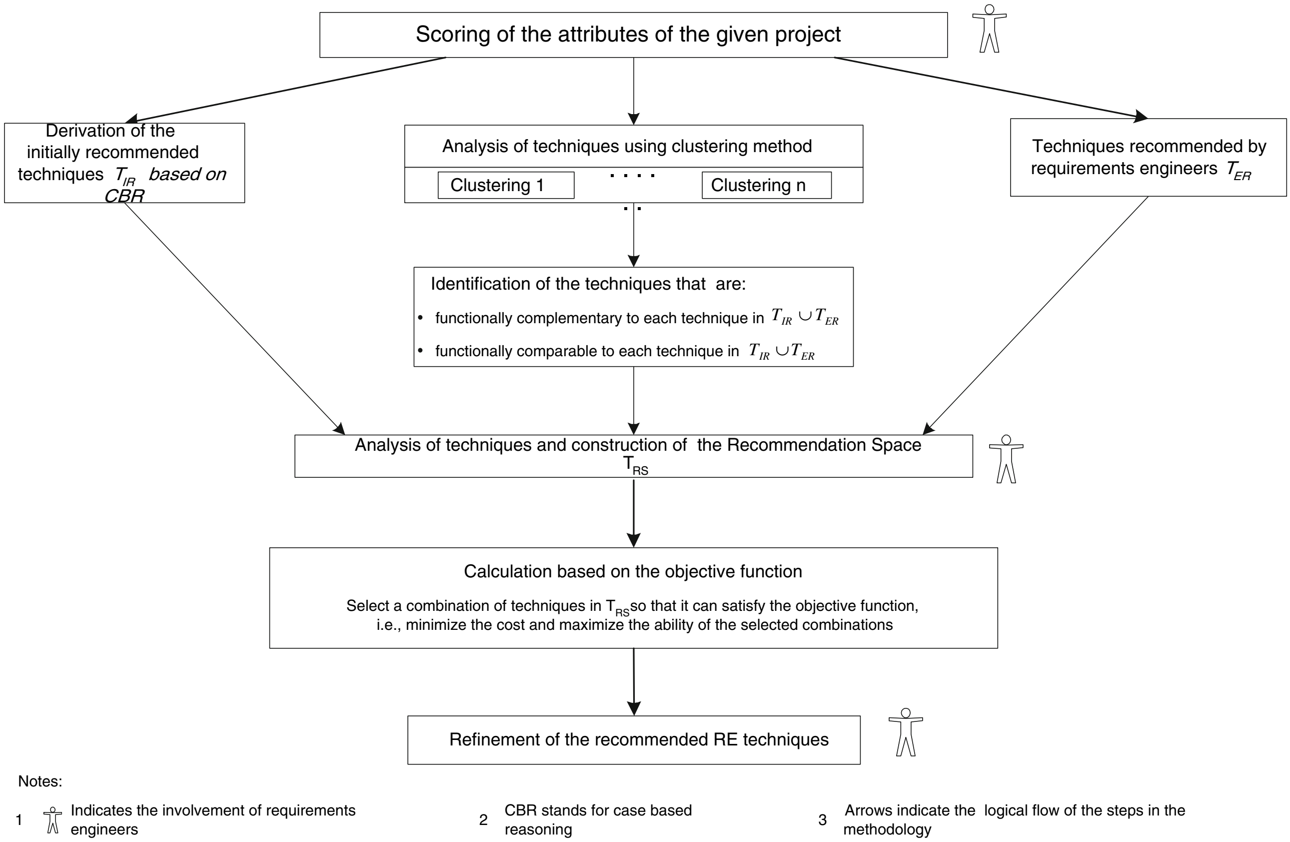

Fig. 1 High-level illustration of MRETS

The numerical factor 2 in the formula 03.2 is an experience factor derived in our case studies during the research which ensures that the cost factor is adequately weighted.
It is worth mentioning that the assignment of the techniques in T_{RS} to T_i can either be done manually by requirements engineers or automatically by computers based on the rules implemented in a support tool. The rule-based tool is still to be developed and the explanation of these rules is beyond the scope of this paper.

3.3 The process of MRETS

The overall approach for the selection of RE techniques is shown in Fig. 1. The approach can be summarized in the following five steps:

Step 1. Scoring of the attributes of the given project.
In this step, the requirements engineers score the attributes of the given project. The scores can be an initial estimate based on the experience of the requirements engineers if detailed information about the project is not yet available.

Step 2. Derivation of the initially recommended RE techniques.
In this step, a set of RE techniques (T_{IR}) is derived based on scored attributes and the rules in REPKB (see Table 7) using a case-based reasoning mechanism. This step does not need the involvement of an expert if computer support is available. The initial recommendation is based on experience from previous projects. It only provides general guidance as to what techniques might be suitable candidates for the new project. However, the situation of the new project at hand will not be exactly the same as the previous projects; therefore, further analysis, refinement and modification of the initial recommendation is likely needed.

Step 3. Analysis of RE techniques using the clustering method.
The major task in this step is to analyze all the techniques in REPKB using the clustering mechanism. This step includes the following sub-steps:
(a) Selection of a set of technique attributes which are considered important for the given project. This step needs the involvement of the requirements engineers. For example, the requirements engineers might be very interested in the following set of attributes: Ability

to facilitate communication, Ability to help identify viewpoints, Ability to facilitate negotiation with customers. Thus, the clustering will be based only on these attributes instead of all 31 attributes. The automation of this step is subject to further research.

(b) Technique clustering based on the technique attributes selected in a).

Step 4. Analysis of the techniques and construction of the recommendation space T_{RS}.

In this step, the requirements engineers construct the techniques recommendation space including potentially suitable techniques for the project. The content of T_{RS} is discussed in Sect. 3.1. This step includes the following sub-steps:

(a) Analysis of the techniques in T_{IR}, to ensure that all the techniques are compatible with the new project. Incompatible techniques are removed from T_{IR}. Requirements engineers can also compare different techniques based on certain attributes such as "Application cost", "Ability to help model and understand requirements".

(b) Selection of a set of prospective techniques from the requirements engineers' perspective denoted as T_{ER}. This makes sure that the requirements engineers' expertise is included in the decision making process. T_{ER} could be an empty set.

(c) Identification of all techniques, which are functionally comparable and functionally complementary to all the techniques in $T_{IR} \cup T_{ER}$, by using the results of the technique clustering of the previous step. The functionally comparable and functionally complementary techniques help requirements engineers find more suitable techniques for the new project.

(d) Combination of the techniques identified in steps a), b) and c) to construct the techniques recommendation space T_{RS}. The requirements engineers are involved in this step.

Step 5. Calculation based on the objective function.

In this step, the requirements engineers select various combinations of RE techniques from T_{RS} based on the calculation of the objective function. This step includes the following sub-steps:

(a) Selection of the various combinations of techniques T_i within T_{RS}.

(b) Calculation of the overall ability of the technique combination T_i selected by requirements engineers based on the values of their attributes using the objective function. This provides a means to evaluate the overall ability and the cost of the candidate techniques so that the most suitable techniques can be selected.

Step 6. Refinement of the recommended techniques selection.

In this step, the final recommendation T_C is adjusted according to the experience of the requirements engineers. Checking the consistency ensures that the recommended techniques are not mutually exclusive, i.e. the usage of one technique t does not violate the basic principles of another technique t'. For example, XP requirements verification and Z requirements verification are mutually exclusive techniques since requirements verification using Z violates basic principles of XP. Some techniques might also be removed based on further evaluation of their complexity and characteristics of the given project. Appropriate requirements management techniques will also be included into the final recommendation.

It has to be mentioned that RE management techniques selection is currently still done manually in an unsystematic manner based on expert's knowledge and familiarity with RE management tools. Automating this process is subject to future research.

Detailed guidelines for each step are defined and stored in REPKB as well as the knowledge about RE techniques and the guidelines for their usage to provide effective support for techniques selection.

4 A case study

Three case studies were conducted throughout the course of this research. The following describes one case study that shows the application of the MRETS approach to an industrial project in company X (the name of the company is withheld for reasons of confidentiality).

Company X is a medium-sized software organization that worked on a Port Scheduling System (PSS). The PSS project took about one and half years and had many non-functional requirements. With the approval of the organization's management, the requirements engineers agreed to apply the MRETS approach to help select a set of suitable RE techniques for the PSS project. The authors were heavily involved in the entire development process in order to provide necessary guidance and help.

The case study was designed according to the guidelines and methodology proposed by Kitchenham et al. [53]:

1. Definition of the hypothesis
2. Selection of the pilot project
3. Selection of a suitable method for comparison and criteria for the validation
4. Consideration of the effects of confounding factors
5. Planning of the case study
6. Conducting and monitoring the case study against the plan
7. Analysis of results and generation of report

The hypothesis of the case study is: "Using a combination of RE techniques selected with MRETS rather than using ad hoc practices has a positive impact on the overall quality of the software requirements specification".

The next two subsections describe steps (6) and (7) in more detail. A detailed description of steps (2) to (5) for the case study can be found in [33].

4.1 Conducting the case study

In this step, the requirements engineers followed the process of the MRETS methodology described in Sect. 3 to derive a set of RE techniques for the project PSS. A summary of the process is given below:

Step 1. Scoring of the attributes of the given project (see Table 8).
After the initial analysis of the problem, the requirements engineers scored the attributes of the PSS project based on its description and perception by requirements engineers. Other attributes that are either unknown or less important are omitted in Table 8.

Step 2. Derivation of the initially recommended RE techniques.
In this step, case-based reasoning was conducted in the REPKB. The result of the reasoning was a project case with project attributes most similar to those of the PSS. Furthermore, the case-based reasoning provided a set of recommended techniques, denoted as T_{IR}, for the new project. In our case, T_{IR} is shown in the third column of Table 9.

Step 3. Analysis of RE techniques using the clustering method.
In this step, the following tasks were carried out:

- Based on the attributes identified for the given project, the requirements engineers chose those technique attributes (see Table 10) that were essential for the selection of suitable RE techniques.
- The requirements engineers assigned weights to each attribute based on the features of the project (see second column of Table 10).
- Based on our previous research [14,33], we recommended 9 clusters, i.e., $P = 9$.
- The requirements engineers conducted technique clustering based on the Fuzzy Clustering algorithm. The outcome of this clustering is presented in Table 11.

Step 4. Analysis of the techniques and construction of the techniques recommendation space T_{RS}.
After the analysis of the techniques in T_{IR}, requirements engineers decided to include all the techniques of T_{IR} into T_{RS}. No additional techniques were suggested by the requirements engineers for inclusion in T_{RS}. Thus, the techniques recommendation space included the following two parts:

- All the techniques in T_{IR}.
- All the techniques that are functionally comparable and functionally complementary to each technique in T_{IR}.

Based on the clustering in the last step, both "Focus group" and "Ethnography" are considered as functionally complementary techniques to "Interview"; and "OO analysis" and "Goal-based analysis" are considered as functionally comparable techniques to "Scenario-based analysis", and "Viewpoint-based analysis techniques", respectively. These relationships are presented in Table 9. The inclusion of functionally comparable techniques and functionally complementary techniques in each cluster provides means to explore additional techniques that might be suitable for the new project.

Step 5. Calculation based on the objective function.

After analyzing the techniques in T_{RS}, requirements engineers selected the most promising combinations of techniques (see Table 12). The calculation of the most suitable technique combination was made using the objective function (see Sect. 3.2). Based on formulas O3.1 and O3.2, the cost of each technique and the final score of each technique in the recommendation space is shown in Table 13.

In most cases, the techniques combination with the highest score will be selected as the final set of techniques unless there are other factors that have to be considered. In this case, further refinement of the techniques selection is required.

The final overall scores for all combinations of RE techniques as determined by the objective function are shown in Table 14. The technique combination 4 (T_4) has the highest score and is therefore the final recommendation based on formula O3.3.

Step 6. Refinement of the recommended techniques selection.

Table 8 Project definition

Project description	The project is to develop a "Port Scheduling System". The objective of the system is to schedule a container terminal with a throughput of a maximum of 1 million TEU (20 foot equivalent unit) each year. The terminal must also be able to handle smaller cargo. The project requires a highly interactive interface and a high degree of automation of the control functions and the scheduling of the port.
Project attributes	Project size: medium ($\geq$ 500 and $<$ 1, 000 requirements)
	Project complexity: Medium
	Requirements volatility: Low
	Organization and customer relationship: SCR (SCR stands for responding to a Specific Customer Request)
	Project category: Semi-detached
	Team size (number of people in the project): 30
	Degree of knowledge of requirements: Medium
Product attributes	Degree of safety criticality: High
	Product quality criteria: High
	Product type: New

Table 9 Techniques recommendation space

	Initial recommendation (T_{IR})		Functionally Comparable Techniques	Functionally Complementary Techniques
RE techniques	T_e:	Interview, JAD		Focus group (with respect to interview), Ethnography (with respect to interview)
	T_a:	Viewpoint-based analysis, Scenario-based analysis (use cases), AHP	Goal-based analysis (with respect to viewpoint-based analysis), OO analysis (with respect to scenario-based analysis)	
	T_d:	Viewpoint definition, Structured natural language specification	UML	
	T_v:	Viewpoint validation, Formal requirements inspection		

1. T_e: stands for the initially recommended Elicitation techniques. T_a: stands for the initially recommended analysis and negotiation techniques. T_d: stands for the initially recommended documentation techniques. T_v: stands for the initially recommended verification and validation techniques
2. Viewpoint-based analysis is a technique which can integrate OO modeling mechanisms into a RE process

The third column of Table 15 lists the techniques contained in the recommended combination. This final recommendation was once more reviewed by the requirements engineers. Based on the characteristics of the project, the requirements engineers realized that the prioritization of requirements in this project is not that difficult and can be done informally; therefore, AHP was removed during this final review. Additionally, no exclusive techniques were identified in the consistency check. Furthermore, requirements change management techniques were included in the final recommendation.

The final decision made by the requirements engineers is listed in the fourth column of Table 15. This decision includes an in-house requirements management tool called "DocManager" since the company had already had previous experience using this tool. Even though DocManager is less powerful than other commercially available requirements management tools (e.g., DOORS [54]), it provided the necessary functionality at much lower cost.

4.2 Results analysis

4.2.1 Quantitative analysis

The recommended techniques were used in the RE process for the PSS project by the requirements engineers in company X. The data collected during the case study in the PSS project which used our MRETS approach, was compared with a previous project of an Intelligent Industrial Waste-Water Treatment System (IWTS) that did not use our MRETS approach. Both projects exhibit very similar project attributes (see Sect. 2.2.1). Table 16 and Fig. 2 compare the two projects

Table 10 Technique attributes selected by requirements engineers

Selected technique attributes	Weight of the attributes
Ability to help get domain knowledge	4
Ability to help identify stakeholders	4
Ability to help identify non-functional requirements	4
Ability to help model and understand requirements	5
Ability to help analyze non-functional requirements	5
Ability to help model interface requirements	5
Ability to help requirements verification	4
Ability to help get implicit knowledge	4
Ability to help write unambiguous and precise requirements by using the notations	4
Ability to help write complete requirements	5
Ability to help with requirements management	4
Ability to help identify interactions (ambiguous, inconsistency, conflict)	3
Maturity of supporting tools	5

Table 11 Result of the clustering

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
Interview	Evolutionary prototypes	Viewpoint-based documentation	Ethnography	Specification and description language (SDL)	Structured analysis (SA)	OOA	Requirements checklists	Future workshops
Contextual inquiry	Exploratory prototypes (throw-away prototype)	Structured natural language specification	Designer as apprentice	Z	Real-time structured analysis	Repre-sentation modeling	Formal requirements inspection	Focus group
Brain-storming and idea reduction	eXtreme programming (XP)	eXtreme programming (XP)	Document mining (document inspection)	LOTOS	Problem frame oriented analysis	UML	Requirements testing	JAD
Viewpoints-based elicitation			AHP	State charts (also known as state diagrams)	Decision tables	Scenario–based approach	Utility test	SQFD
eXtreme Programming (XP)			Card sorting	State machine	Functional decompo-sition	ERDs	Goal-oriented verification and validation	SSM
			Repertory grids	Petri-nets	Fault tree analysis		Viewpoint-based verification and validation	
			Laddering	Goal-oriented analysis	Viewpoint-based analysis			

It is worth mentioning that Extreme Programming (XP), considered as a technique in this research, appears in three clusters. The reason for this is that the membership of XP is almost the same within three of the clusters. This reveals the fact that XP partially addresses requirements elicitation (using the "customer online method"), requirements analysis (using the "prototyping method"), and documentation (using the "user story card"). Even though XP involves some requirements validation, it does not score high enough to be included in cluster 8 where the requirement verification techniques are located

Table 12 Technique combinations

No.	Technique combination	T_e	T_a	T_d	T_v
1	T_1	Interview, Focus Group, Ethnography,	Viewpoint-based analysis, AHP	Structured natural language specification	Viewpoint validation
2	T_2	Interview, Focus Group, Ethnography,	OO Analysis, AHP	Viewpoint-based definition	Formal requirements specification
3	T_3	Interview, Focus Group, Ethnography,	Scenario-based analysis, AHP	Structured natural language specification	Formal requirements inspection
4	T_4	Interview, Focus Group, Ethnography	Viewpoint-based analysis, AHP	Viewpoint-based definition	Formal requirements inspection
5	T_5	Interview, Focus Group, Ethnography,	Goal-oriented analysis, AHP	Viewpoint-based definition	Formal requirements inspection
6	T_6	Interview, Ethnography,	Scenario-based analysis, AHP	Viewpoint-based definition	Viewpoint validation
7	T_7	JAD, Interview	Viewpoint-based analysis, AHP	Viewpoint-based definition	Formal requirements inspection
8	T_8	JAD, Interview	Scenario-based analysis	Structured natural language specification	Viewpoint validation

Table 13 Scoring results

Category	Name of the techniques in T_{RS}	Abilities of each technique	Cost of each technique	Final score for each technique based on equation O3.1)	Notes
Elicitation	Interview	5.2	0.6	4.0	B = 0
	Focus Group	5.8	1.2	3.4	B = 0
	JAD	5.6	1.8	2.0	B = 1
	Ethnography	4.8	1.4	2.0	B = 1
Analysis	Viewpoint-based analysis	4.4	0.7	3.0	B = 0
	Scenario-based analysis (use case)	4.8	1.2	2.4	B = 1
	OO analysis	3.6	1.2	2.0	B = 0
	AHP	1.6	1.2	−0.8	B = 1
	Goal-oriented analysis	4.2	2.2	−0.2	B = 1
Documentation	Viewpoint-based definition	6	1	4.0	B = 0
	Structured natural language specification	3.4	0.2	2.6	B = 0
	UML	6.6	1.8	3.0	B = 0.4
Verification and validation	Viewpoint-based validation	2.8	2.2	−1.6	B = 1
	Formal requirements inspection	2	1	0	B = 0

carried out by the same team except that two junior developers were not part of the PSS project. As can be seen, the two projects are similar in the number of people involved and the duration. However, even though the PSS project has about 25% more requirements than the IWTS project, it required less development time. Furthermore, the PSS project was only 8.33% over time, while the PSS project was 31.25% over time in terms of person-months. The likely reason for this is that the recommended RE techniques helped discover and correct more requirements earlier on than was the case in the IWTS project, which experienced very late requirements changes. As can be seen from Table 13, the only training cost was related to the Ethnography technique ($B = 1$) since the requirements engineers were already familiar with the other techniques. The comparison of the two projects shows that the advantages of using the recommended RE techniques outweigh the costs associated with the training and the time spent on applying additional RE techniques. Furthermore, the skills gained from the training were not only applied to this one project, but developers continue to use their newly gained knowledge in future projects.

4.2.2 *Qualitative analysis*

In addition to the quantitative analysis (presented in the last sub-section), a questionnaire was conducted among all the developers, requirements engineers as well as managers who were involved in the PSS project [33]. The feedback from the developers about the suitability of the recommended techniques in the project was very positive. The company was able to develop a much better requirements specification with more precise definitions, clearer structure and traceability compared to previous software projects. But most importantly, requirements ambiguity and conflicts were greatly reduced [33]. The requirements engineers and project manager emphasized that the high quality of the requirements specification had a positive impact on the software project compared to the project IWTS project. The data collected in the case study show that requirements volatility was lower compared to similar projects carried out previously by the company. A key success indicator is that no major requirement that would have had a significant impact on the overall system structure or on major functionality (see Table 16 for the definition of "major requirements") was added or deleted after the completion of the RE phase.

4.2.3 *Observations and discussion*

The following observations were made based on qualitative and quantitative data collected throughout this research:

- The developers did not see any need to change the recommended final set of techniques, and most noticeably, both the qualitative and quantitative data collected from the survey that has been done subsequently suggested that recommended techniques made a positive contribution to the success of the software project [33].
- The recommended techniques are not applied sequentially. Table 17 illustrates the usage of the recommended techniques in the project and shows that techniques were not used in a fixed order but iteratively or in parallel depending on the situation. This shows that the use of the recommended techniques in practice cannot be totally pre-programmed and is subject to the situation and the judgment of requirements engineers.
- Not all features of a technique have to be used. Only those features that were necessary based on the situation were used. For example, only the following two activities of the technique Ethnography were used: "An in depth study of one or more situations" and "The study of action in a social and cultural context". The partial use of RE techniques has already been discussed in [55]. Ethnography is a relatively unknown technique and not used regularly in software projects. However, it was strongly recommended in this project due to the fact that the project team was not familiar with the problem domain of the project. The use of Ethnography in this project led to the discovery of essential port management and scheduling functionalities which would have otherwise been overlooked.
- Requirements engineering is not the sole duty of requirements engineers. The involvement of developers and senior management in the RE process under the leadership of requirements engineers has definitely had a positive impact on the project. This conclusion is consistent with results reported in [56].

Table 14 Overall ability scores for each technique combination

No.	Technique combination	Sum of ability scores for all the techniques in each techniques combination based on (O3.2)
1	T_1	12.6
2	T_2	14.6
3	T_3	13.6
4	T_4	15.6
5	T_5	12.4
6	T_6	10.0
7	T_7	12.2
8	T_8	9.4

This case study illustrates the help MRETS provided during RE process development. We acknowledge that comparing the data from the two projects cannot be used to proof that our methodology will always improve project success. There are many other factors that have an impact on project success and can reduce the validity of the study:

- Management commitment: The management of the two projects had slightly different levels of commitment to the RE process. Management of the PSS project wanted to use a well-developed RE process model right from

Table 15 Recommendation and final selection of techniques

Categories	Initial recommendation	Final recommendation	Final decision
Elicitation	Interview, JAD	Focus Group, Interview, Ethnography	Focus Group, Interview, Ethnography
Analysis & Negotiation	Viewpoint-based analysis, Scenario-based analysis (Use Cases), AHP	Viewpoint-based analysis, AHP	Viewpoint-based analysis
Documentation	Viewpoint-based definition, Structured natural language specification	Viewpoint-based definition	Viewpoint-based definition
Verification and validation	Viewpoint validation, Formal requirements inspection	Formal inspection	Formal requirements inspection
Management			Requirements management supported by an in-house requirements management tool called "DocManager"

Table 16 Data collected from the current project and a previous project

Measured Data / Project Name		Port Scheduling System (PSS)	Intelligent Industrial Waste-Water Treatment System (IWTS)
RE techniques used		Focus Group, Interview, Ethnography, Viewpoint-Based Analysis, Viewpoint-Based Definition, Formal Requirements Inspection, Requirements management supported by an in-house requirements management tool called "DocManager"	Informal Focus Group, Interview, OO Modelling, informal documentation, and informal requirements verification and validation
Total number of (atomic) requirements in the final requirements specification		882	702
Number of analysts involved (play the role of requirements engineers as well)		4	4
Number of developers involved		49	51
Number of original requirements		348	307
Requirements elicited using the recommended RE techniques	Absolute	496	266
	% of the total number of requirements	56.2%	37. 9 %
Number of requirements deleted during requirements verification and validation		43	0**
Number of requirements modified during requirements verification and validation		164	0**
Number of requirements added during requirements verification and validation		12	0**
Number of requirements discovered during the design stage	Absolute	55	138
	% of the total number of requirements	6. 2%	17. 5 %
Number of requirements deleted during the design stage		0*	74
Number of major requirements changed after start of design	Absolute	0	7
	% of the total number of requirements	0	1%
Number of requirements discovered during the testing stage	Absolute	14	65
	% of the total number of requirements	1. 6 %	9. 3 %
Number of requirements changed after design began	Absolute	69	277
	% over the total number of requirements	7. 8 %	39.5 %
Project duration	Planned	18 months	16 months
	Actually spent	19.5 months	21 months
Effort in Person-months	Planned	882	816
	Actually spent	955.5	1071
Cost overrun in terms of the Effort in Person-months	Number	73. 5	255
	% over the total effort of the project	8.3%	31.3%

Notes:
1. 0* indicates that "No requirement was deleted"
2.. 0** indicates that "No requirements verification and validation techniques were used"
3. major requirement is defined as the requirement which has major impact to the overall system structure and major functionalities

the beginning. However, based on our information about both projects, the difference in commitment was not large enough to have a major impact on the result.

- Learning effects and training: Learning effects play an important role if the projects are in the same domain. However, since the two projects are in two different domains, learning effects were considered minimal. Additionally, the amount of training provided for the PSS project was limited. Therefore, these factors are not considered as a major reason for the success of the PSS project.
- Available project data: The data collected from the sister project IWTS was limited which did not allow us to compare the two projects as thoroughly as we would have liked. The difficulty in getting sufficient data to verify the proposed theory or arguments in software engineering research has already been identified by Glass [57]. This is also the case in this research. However, it is worth

mentioning that even though more techniques were used in the PSS project than in the IWTS project, this does not necessarily means that the more techniques are used, the better the result will be. In fact, using more RE techniques might even delay the software project and possibly even lead to project failure. Examples of project failures due to inappropriate use of RE techniques can be found in [25]. The key point is to use those techniques that satisfy the constraints of the project and contribute to the overall quality of the requirements at the same time.
- Other factors. These factors are related to the personal attitudes and experiences of people during the application of the MRETS methodology and might also have influenced the techniques selection. Additionally, factors such as experience of project manager and staff, proper planning, ownership, also play a role in the success of software projects [58]

Nevertheless, the data collected is supported by both the qualitative and quantitative feedback of requirements engineers and developers involved. All of the requirements engineers and developers agreed that the two projects are comparable with regards to the defined controlled variables of the project. The case study supports the fundamental assumption made by the RE community that getting high-quality requirements early on will reduce rework and overall development cost. Furthermore, developers stressed that the high-quality of the requirements made tracing requirements to their sources possible, understanding the requirements easier, and conflict resolution more effective compared to previous projects. Additionally, the company is committed to collaborate with us in future and to apply our approach for RE techniques selection in upcoming projects.

5 Conclusions and future work

In this paper, we proposed an approach called MRETS for the selection of RE techniques using clustering and decision support mechanisms. The feasibility of the MRETS approach was demonstrated in a case study which we compared with data from a previous project. The research has so far resulted in the following:

1. It provides an extensive review of significant research efforts in the RE domain over the last 2 decades [7,33]. The large number of RE techniques provides us with numerous alternatives to select appropriate RE techniques for a given project.
2. The research thoroughly investigated the advantages and disadvantages of different RE techniques and their applicability in practice [7,14,33].
3. The clustering method and decision support mechanism help find more suitable, less complex and less expensive techniques for a specific task.
4. The limitations in our research are inevitably influenced by the scope of the RE techniques included and the subjective nature of the data collected. In brief, they include the following aspects:
    - The limited number of the RE techniques. Currently, only 46 RE techniques are included in the REPKB. More techniques will be added to REPKB in future.
    - The completeness and accuracy of the techniques evaluation schema. The 31 technique attributes do not cover all aspects of RE techniques. They will likely require further refinement.
    - Our understanding of RE techniques. The evaluation of some techniques can be controversial. This can have an impact on the clustering.
    - The sources of the data. The data is derived from RE experts and available literature on RE techniques. A completely objective assessment of the effectiveness of RE techniques is impossible.
    - The lack of availability of detailed descriptions of some techniques. This has an impact on the accuracy of the final result of the clustering.
    - The classification schema and rating methods used for the RE techniques analysis still have limitations and will be improved in our future research.
    - The limited number of recommendation rules in the REPKB and the limited number of software project types covered in the recommendation rules also impact the decision making process of RE technique selection.
    - The links between the attributes of the software project and the attributes of the RE techniques are still relatively weak and some decisions still have to be made by requirements engineers. This is due to the large diversity of software projects and the multidimensional nature of RE techniques, which make the establishment of a direct link between the two kinds of attributes very challenging. However, we argue that it is at the moment only possible to develop this kind of logical link between the attributes of the software project and the techniques themselves. Development of direct links between attributes of the software project and attributes of RE techniques is still subject to future research.

Nevertheless, we claim that this research provides significant progress in the analysis and classification of RE techniques for helping requirements engineers select the most appropriate RE techniques for a project. The major contribution of this research can be summarized as follows:

- First, it proposed a techniques evaluation schema which includes 31 attributes that need to be considered when selecting RE techniques.

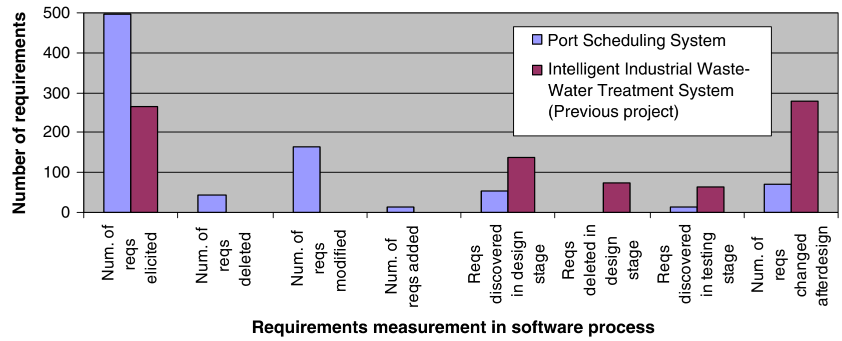

Fig. 2 Comparison of requirements change between the PPS and IWTS projects

Table 17 Techniques used in the project

Techniques used in the project	Major objectives
Unstructured interview	Identify additional information about the scope, social context and stakeholders of the project.
Formal (structured) interview	Identify stakeholders and elicit the essential requirements from key stakeholders. Focus on those stakeholders that are not available to attend the focus group meeting. The elicitation results include both functional and non-functional requirements.
Focus Group	Elicitation of requirements from different stakeholders and their viewpoints.
Viewpoint-based analysis	Modelling requirements to improve understanding.
Interview	Elicitation of requirements from those stakeholders who did not participate in the scheduled Focus Group meeting. This ensures that all key stakeholders can express their views about the system.
Ethnography	To elicit the implicit requirements and functional requirements.
Viewpoint-based analysis	Modelling requirements to improve understanding.
Viewpoint-based definition	Define and document the requirements.
Interview	Requirements negotiation, verification and validation.
Inspection	Verify and validate the requirements.
. . .	. . .

- Second, using clustering and the decision support mechanisms, MRETS provides help for the selection of the most suitable techniques for a given project.
- Third, this approach proposed a way that links the attributes of a software project to the attributes of RE techniques which ensures that selected RE techniques are well-suited to the project. Currently, these links are still done manually, yet further research will result in automating the linking.

The cases studies carried out during this research indicate the usefulness and applicability of the proposed approach. By helping requirements engineers select suitable RE techniques for a given project, MRETS contributes to the overall quality of the requirements specification, which, in turn, contributes to the overall quality of the software product. Therefore, we argue that this research made not only a contribution to RE, but also to the software engineering domain in general.

Our future work will focus on the refinement of the approach and the development of tools to support MRETS.

Acknowledgments The authors would like to thank the editors and the three anonymous reviewers for their very detailed and valuable comments that helped improve this paper.

References

1. Boehm, B.W.: Software Engineering Economics, Prentice-Hall, Englewood Cliffs (1981)
2. Macaulay, L.A.: Requirements for requirements engineering techniques. In: IEEE Proceedings of ICRE, pp. 157–164 (1996)
3. Sommerville, I., Sawyer, P.: Requirements Engineering: A Good Practice Guide, Wiley, New York (1997)
4. Herlea, D., Jonker, C.M., Treur, J., Wijngaards, N.J.E.: A case study in RE: a personal internet agent, Technical Report, Vrije Universiteit Amsterdam, Department of Artificial Intelligence (1998)
5. Thanasankit, T., Corbitt, B.: Towards understanding managing RE—a case study of a Thai Software House. In: Proceedings of 10th Australasian Conference on Information Systems (1999)
6. Emam, K.E., Birk, A.: Validating the ISO\IEC 15504 measure of software requirements analysis process capability. IEEE Trans. Softw. Eng. **26**(6), 541–566 (2000)
7. Jiang, L., Eberlein, A.: Requirements engineering: a review and a proposal. In: Proceedings of the Third ASERC Workshop on Quantitative and Software Engineering, 17–18 February 2003. Banff, Alberta, Canada (2003)

8. Glass, R.L., Vessey, I.: Contemporary application-domain taxonomies. IEEE Softw. **12**(4), 63–76 (1995)
9. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: Finkelstein, A. (ed.) The Future of Software Engineering. ACM Press, New York (2000)
10. Sutcliffe, A.: A Technique combination approach to requirements engineering. In: 3rd IEEE International Symposium on Requirements Engineering (RE'97), 05–08 January 1997, Annapolis, Maryland, USA (1997)
11. Mannio, M., Nikula, U.: Requirements elicitation using a combination of prototypes and scenarios, Technical Report. Telecom Business Research Center, Lappeenranta, Lappeenranta University of Technology, P.O. BOX 20, FIN-53851 Lappeenranta, Finland, ISBN 951-764-528-7 (2001)
12. Macaulay, L.A.: Requirements engineering, Applied Computing. Springer, New York (1996)
13. Davis, A.M.: Software Requirements, Objects, Functions and States. Prentice-Hall, Upper Saddle River (1993)
14. Jiang, L., Eberlein, A., Far, B.H., Majid, M.: An Innovative requirement engineering technique selection model, Technical Reports, 30 April 2004. Department of Electrical and Engineering, University of Calgary, http://www.enel.ucalgary.ca/∼ jiangl/research/indexOfResearch.htm (2004)
15. Kumar, K., Welke, R.J.: Methodology engineering: a proposal for situation specific methodology construction. In: Cotterman, W., Senn, J. (eds.) Challenges and Strategies for Research in Systems Development, pp. 257–266. Wiley, Chichester (1992)
16. Brinkkemper, S.: Method engineering: engineering the information systems development methods and tools. Inf. Softw. Technol. **37**(11) (1995)
17. Brinkkemper, S., Lyytinen, K., Welke, R.: Method Engineering, Principles of Method Construction and Tool Support, Chapman & Hall, London (1996)
18. Rossi, M., Tolvanen, J.P.: Using reusable frameworks in development of a method support environment. In: Proceedings of the WITS 1995, vol. 95, no. 15, pp. 240–249. RTWH, Aachen (1995)
19. Lyytinen, K., Welke, R.: Metamodeling and method engineering, Special issue on metamodeling and method engineering. Inf. Syst. Int. J. **24**(3), (1999)
20. Tolvanen, J.P.: Incremental method engineering with modeling tools: theoretical principles and empirical evidence, PhD thesis, Computer Science, University of Jyväskylä (1998)
21. Saeki, M.: Software specification & design methods and method engineering. Int. J. Softw. Eng. Knowl. Eng. (1994)
22. Maiden, N., Rugg, G.: ACRE: slecting methods for rRequirements acquisition. Softw. Eng. J. **11**(3), 183–192 (1996)
23. Hickey, A.M., Davis, A.M.: Requirements elicitation and elicitation technique selection: a model for two knowledge-intensive software development processes. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003., 6–9 January 2003, pp. 96–105 (2003)
24. Hickey, A.M., Davis, A.M.: Elicitation technique Selection: how do experts do it? In: Proceedings of the 11th IEEE International Requirements Engineering Conference (2003)
25. Bickerton, J.M., Siddiqi, J.: The classification of requirements engineering methods. In: Proceedings of IEEE International Symposium on Requirements Engineering, 4–6 January 1993, pp. 182–186. San Diego, CA ,USA (1993)
26. Kotonya, G., Sommerville, I.: RE, Processes and Techniques, John Wiley & Sons Ltd, New York (1998)
27. Davis, G.B.: Strategies for information requirements determination. IBM Syst. J. **21**(1), 4–31 (1982)
28. Browne, G.J., Ramesh, V.: Improving information requirements determination: a cognitive perspective. Inf. Manage. **39**, 625–645 (2002)
29. Lobo, L.O., Arthur, J.D.: An objectives-driven process for selecting methods to support requirements engineering activities. In: 29th Annual IEEE\NASA Software Engineering Workshop (2005)
30. Lauesen, S.: Software Requirements: Styles and Techniques, Addison Wesley, New York (2002)
31. Jiang, L., Eberlein, A., Far, B.H.: Evaluating the requirements engineering process using major concerns, Technical Reports, 30 August 2003, Department of Electrical and Engineering, University of Calgary. http://www.enel.ucalgary.ca/∼jiangl/research/indexOfResearch.htm (2003)
32. Glass, R.L.: Matching methodology to problem domain. Commun. ACM **47**(5), 19–21 (2004)
33. Jiang, L.: A framework for requirements engineering process development, PhD thesis, University of Calgary (2005)
34. Keyes, J.: How software is developed undergoing basic changes. Softw. Magaz. **12**(1), 38–40, 43–47, 55–56 (1992)
35. Wieringa, R.J.: Requirements Engineering: Frameworks for Understanding, Wiley, New York (1996)
36. Bray, I.K.: An Introduction to Requirements Engineering, Addison Wesley, New York (2002)
37. Jiang, L., Eberlein, A., Far, B.H.: A methodology for RE process development. In: ECBS 11th IEEE International Conference on the Engineering of Computer-Based Systems, Brno, Czech Republic, 2004
38. Nielsen, P.A.: Reflections on development methods for information systems: a set of distinctions between methods. Off. Technol. People **5**(2), 81–144 (1989)
39. Agell, I.O., Straub, B.H.: Though this be madness, yet there is method. Int. J. Strat. Inf. Syst. **2**(1), 5–14 (1993)
40. Extreme Chaos (2001) The Standish Group International, Inc.
41. Jiang, L., Eberlein, A., Far, B.H.: A RE process Template for Aiding RE process development, Technical Reports, 30 August 2003. Department of Electrical and Engineering, University of Calgary. http://www.enel.ucalgary. ca/∼jiangl/research/indexOfResearch. htm(2003)
42. Booch, G.: Object-oriented design with applications, Benjamin\Cummings, New York (1994)
43. Jiang, L., Eberlein, A., Far, B.H.: Evaluating the requirements engineering process using major concerns. In: Proceedings of the IASTED International Conference on Software Engineering (SE2004) as part of the Twenty-Second IASTED International Multi-Conference on Applied Informatics, Innsbruck, Austria, 17–19 February 2004, pp. 30–35
44. Mesiar, R., Calvo, T., Mayor, G.: , Aggregation Operators: New Trends and Applications, Physica-Verlag, New York (2002)
45. Jiang, L., Eberlein, A.: Clustering requirements engineering techniques. In: IASTED International Conference on Software Engineering and Applications (SEA 2006), Dallas, USA (2006)
46. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surveys (CSUR), **31**(3), 264–323 (1999)
47. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1998)
48. Chiu, S.: Fuzzy model identification based on cluster estimation. J. Intell. Fuzzy Syst. **2**(3) (1994)
49. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York (1981)
50. Pfeiffer, S.: Requirements Engineering for Dynamic Markets, MSc Thesis. Department of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta, 2003
51. Salzer, H.: ATRs (Atomic Requirements) Used Throughout Development Lifecycle. In: 12th International Software Quality Week (1999)
52. Helmer, O., Rescher, N.: On the Epistemology of the Inexact Science, Management Science, 1959, vol. 6 Issue 1, p25, 28p

53. Kitchenham, B., Plckard, L., Pfleeger, S.L.: Case studies for method and tool evaluation. IEEE Softw. 52–62 (1995)
54. Telelogic DOORS (2002) http://www.telelogic.com/products/doorsers/doors/ index.cfm, 2002
55. Fitzgerald, B.: An investigation of the use of system development methodologies in practice. In: Fourth European conference on information systems, pp 143–161 (1996)
56. The Chaos Report (1995) The Standish Group International, Inc., 1995 http://www.standishgroup.com/sample_research/index.php
57. Glass, R.L.: The software-research crisis, Software, IEEE, **11**(6), 42 – 47 (1999)
58. A Recipe for Success (1999) The Standish Group International

Author's Biography

Li Jiang got his Bachelor Degree of Software Engineering from Liaoning University of Petroleum and Chemistry in China, and a PhD in Software Engineering from University of Calgary, Canada. He is currently a lecturer at the University of Adelaide, in Australia. Prior to coming to University of Adelaide, he was a senior lecturer in University of New Brunswick, Canada. Before he came to Canada, Li Jiang was an associate professor at the Liaodong Institute, in China. Li Jiang worked as software engineer, analyst, and architect in number of companies in China before he became an academician. His research interests focus on the intelligent support for the improvement of software engineering practice in general; improvement of the requirements engineering practices in particular. He has been visiting scientist at Shanghai Jiaotong University and University of Calgary in 1996 and 2000, respectively.

Armin Eberlein received the Dipl-Ing (FH) degree in Telecommunications Engineering at the Mannheim University of Applied Sciences in Germany, the M.Sc. degree in Communications Systems, and the PhD degree in Software Engineering from the University of Wales, Swansea, UK. He is currently an Associate Professor and the Head of the Computer Engineering Department at the American University of Sharjah in the United Arab Emirates. Previously, he was an Associate Professor at the University of Calgary, Canada, where he was the Director of the Software Engineering program and one of the Co-Directors of the Alberta Software Engineering Research Consortium (ASERC). His research interests focus on the improvement of requirements engineering practices and techniques. He worked previously as a hardware and software developer at Siemens in Munich, Germany, and has consulted for various companies in Germany, the UK, and Canada.

Behrouz Far received B.Sc. and M.Sc. degrees in Electrical Engineering from Tehran University, Iran, in 1983 and 1986 respectively, and PhD from Chiba University, Japan, in 1990. Before moving to the University of Calgary, where he is an Associate Professor at the Department of Electrical and Computer Engineering, he was an Associate Professor at the Department of Information and Computer Sciences, Saitama University, Japan. His research interest is in engineering of intelligent, distributed and heterogeneous networked software systems, specifically in design and implementation of agent-based software systems and support tools and techniques for groupware systems.

Abdolmajid Mousavi received his Master of Science degree in electrical and computer engineering from university of Tehran, Iran, in 1997, and enrolled in the PhD program of the department of electrical and computer engineering at the University of Calgary, in 2003. From 1997 to 2003, he was jointly the manager of the computer networks division of Behineh Pardazi Co., in Iran, a lecturer at the University of Lorestan, Iran, and a consultant for medical instrumentation standards in the National Institute of Industrial Research and Standardization of Iran. Currently, he is doing his PhD with a major in software engineering at the University of Calgary. His main research interest is application of artificial intelligence and formal methods in software engineering.