A *F*RAMEWORK FOR *R*EQUIREMENTS *E*NGINEERING P*R*OCESS D*E*VELOPMENT (FRERE)

Li Jiang
School of Computer Science, The University of Adelaide, Australia
ljiang@cs.adelaide,edu.au

Armin Eberlein
Computer Engineering Department, American University of Sharjah, UAE
eberlein@ucalgary.ca

Abstract:

Adoption of the most suitable Requirements Engineering (RE) process and selection of the most appropriate RE techniques for a given project are challenging issues for industry. In order to alleviate the problems, a Framework for RE pRocess dEvelopment (FRERE) was developed based on our research in the last six years. The overall architecture of the framework and a brief case study are presented in the paper. The framework addresses several core issues of the RE process: engineering issues, domain issues, characteristics of the software project, and issues of conformity to software process standards. FRERE consists of three parts: (1) The RE Process Knowledge Base (REPKB) which contains RE process knowledge, (2) Methodologies which provide guidance to develop a customized software process model in an iterative way; and (3) Assessment Models which provide information about the suitability of the newly developed RE process model for the software project under development.

1 Introduction

There is consensus that RE is a critical process within the Software Engineering (SE) lifecycle [1-3]. Research provides empirical evidence of the benefits of RE or provides statistical results showing that improving the RE process will likely lead to improvements in software quality and productivity in large and medium-sized software organizations [1-4]. Most people agree that following a well-defined RE process and using appropriate RE techniques has a positive impact on software quality. Currently, numerous RE process models are available and each one uses various RE techniques to address different issues of software development. In our previous research, we identified 26 RE process models and 56 techniques that are well-defined and documented [5]. However, we found that there is still a big gap between RE theory and practice, and good RE process models and techniques are not adequately used in software project development [5-7]. There are a number of reasons that contribute to this, such as technical maturity of the organization, organizational culture, application domain, etc. In our research, we found that the lack of adequate guidance and help for selecting the best RE practices, process models and techniques for a specific software project is one of the main reasons for the lack of RE practices in real projects.

The need for selecting adequate practices, process models, and techniques for a specific software project has been addressed by many researchers [4-11]. Furthermore, Glass stresses that industry needs guidelines on how and when to use the various techniques in projects of different domains [8]. In order to address the issues discussed above, a framework of RE process development, called FRERE, was developed based on our analysis of the state-of-the art research from various areas of RE and SE, and survey results collected from experts in academia and industry. The theory and practice of software process engineering [12-13] is another foundation for building the FRERE framework. The major objective of the framework is to provide help to engineers to develop and customize RE process models supported by the FRERE tool.

The constructive research approach [14-15] was adopted during the development of the FRERE framework. Ideally, the framework can be used for all types of software projects; however, it fits best to projects that use a complete RE process. The FRERE framework is presented in a condensed manner in this paper. A detailed discussion of each specific component can be found in [11].

The rest of this paper is organized as follows: Sets of attributes of techniques, process models, and software projects are presented in Section 2. The FRERE framework is presented in Section 3. A case study is summarized in Section 4. Conclusions and future work are discussed in Section 5.

2 Attributes of RE Processes, Techniques, and Software Projects

2.1 Attributes of RE Processes and Techniques

Having a good understanding of existing RE process models and RE techniques is essential for developing the most suitable RE process model for a software project. To achieve this objective, it is important to define a set of

1530-0803/08 $25.00 © 2008 IEEE
DOI 10.1109/ASWEC.2008.26

IEEE computer society

attributes which can best characterize RE techniques and process models. Based on our analysis of RE process models and techniques, 31 attributes were defined to characterize techniques and processes. Examples of these attributes are listed in Table 1. These attributes help to classify and analyze RE techniques and process models in detail and to differentiate between them.

Table 1 Examples of attributes of techniques and processes

Aspects	Examples of attributes of the Techniques and Process
Technical perspectives	Ability to facilitate communication
	Ability to help understand social issues
	Ability to help get domain knowledge
	Ability to help get implicit knowledge
	Ability to analyze and model requirements with understandable notations
	Ability to help analyze non-functional requirements
	Ability to facilitate negotiation with customers
	Ability to help prioritize requirements
Cost perspective	Learning curve (Introduction cost)
	Application cost
	Complexity of techniques

2.2 Attributes of Software Projects

In order to ensure the applicability of the newly developed RE process model for a software project, it is vital that the characteristics of the software project are considered [16]. Furthermore, several researchers argue that software projects with different characteristics require different process models and techniques [7; 18-19]. Therefore, one of the essential tasks in this research is to identify a core set of software project attributes which can best represent the characteristics of software projects. Based on previous research [7; 17; 20-21], 21 project attributes were defined in the FRERE framework. All these attributes and their definitions can be found in [11]. Some examples of attributes are given in the following:

(1) Project Size: This attribute is defined as the size of the project (X) in terms of number of requirements. Possible values are: Very Big (X>=4000 requirements), Big (4000=>X>=2500), Medium (2500=>X>=1000), Small (1000=>X>100), Very Small (X<100).

(2) Requirements Volatility: This attribute is defined as the percent of requirements that are likely to change throughout project development (Y). The attribute can have the following values: Very High (Y>=50%), High (50%>Y>=30%), Medium (30%>Y>=10%), Low (10%>Y>=1%), and Very Low (Y<1%).

(3) Project Category: This attribute defines the type of project. Possible values are: Communication, Embedded, Semi-embedded and Dynamic. Some of these values are borrowed from the COCOMO model [1].

Similarly, we define other attributes: Product Type, Team Size, Project Complexity, Time Constraints, Degree of Safety Criticality, Organization and Customer Relationship, 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. It is worth mentioning that these attributes were defined in such a way that they can be logically linked to the characteristics of RE process models and techniques. For example, Time Constraints of a software project is relevant to Training cost and Application cost of RE techniques. The issue of Product complexity in a software project requires RE techniques with higher ability to elicit, model, document, and verify the requirements of the project.

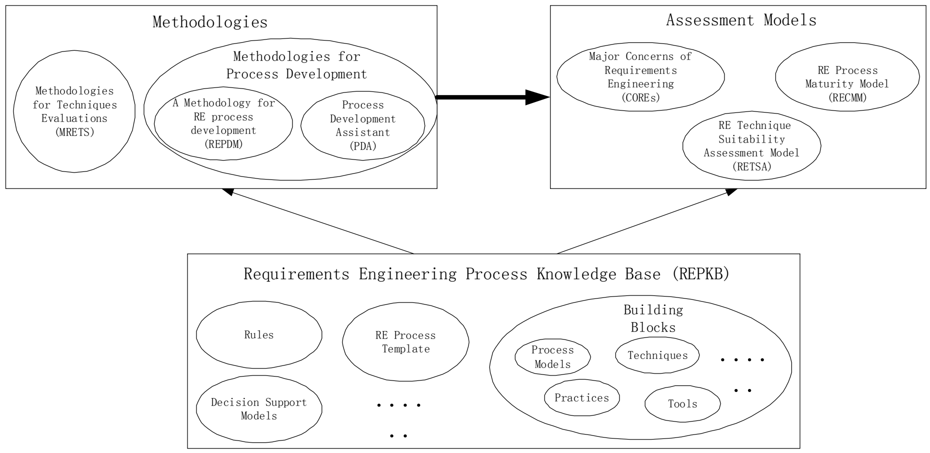
3. FRERE Framework

A high-level abstraction of the FRERE framework is shown in Figure 1. FRERE consists of three parts: the RE Process Knowledge Base (REPKB), Methodologies[1], and Assessment Models. Each part can be further decomposed into components which provide different kinds of support for RE process development.

There are several reasons for designing such a framework:

- Detailed knowledge of RE process models and techniques is of valuable help for decision making during RE process development. In the last three decades, numerous RE processes and techniques have been developed. Knowledge about them has to be analyzed, documented, and stored in a structured manner so that it can be used when developing RE process models. Additionally, such a repository of RE knowledge can also be used as a tool to educate software engineers and students in SE.
- RE process development is a complicated process. It is important to develop a meta-process model that can be instantiated when an RE process model is to be developed for a given project. This meta-model provides guidelines on how to develop an RE process model and how to select appropriate building blocks. In a similar manner, an independent methodology for technique selection is also required to aid RE technique selection.
- Once an RE process model is developed, assessing its suitability for its anticipated project is necessary. Additionally, it is important to know the goodness of the process model with regards to RE principles, best practices and processes. The RE process assessment models address these issues.

[1] It is worth mentioning that the terms “methodology”, “method” and “technique” is used in several different ways in literature [11]. Some researchers argue that “technique” is the only proper word to describe existing methods and methodologies in the software domain. On the other hand we use “methodology as a reserved word in this research that refers to a sequence of systematic steps that aid RE process development.



Notes: (1) The line "→" indicates "support" relationship (2) The notation "➡" indicates "use " relationship

Figure 1 Components of the FRERE Framework

All these components are indispensable for supporting RE process development. Making best use of each component and combining them into a cohesive framework is one of the major characteristics of this research. In the following sub-sections, we will present some essential components of the FRERE framework.

3.1 RE Process Knowledge Base (REPKB)

The REPKB is an essential part of the FRERE framework. It was developed based on a detailed analysis of the literature about RE processes, techniques, and good practices. In order to ensure that a newly developed RE process model fits best for a software project, the REPKB is organized into the following major components: RE Process Building Blocks, Rules, Cases, and Templates. Several of these components are discussed below.

3.1.1 RE Process Building Blocks

An RE Process Building Block (see Fig. 2) is a mechanism to organize knowledge about RE processes in the framework. Each building block is dedicated to a certain part of RE process knowledge. There are 13 building blocks in REPKB which provide knowledge units that are used during the development of new RE processes [11].
The building blocks 1 to 4, 7, 9 and 11 have the same structure and each one consists of following parts:

- A set of RE practices (or activities). For instance, the requirements elicitation building block is composed of knowledge about requirements elicitation and contains a set of activities and guidelines for conducting requirements elicitation. The assignment of a certain practice to a building block is based on its most common usage. The objective of assigning an activity to only one building block is to manage the RE process knowledge more effectively.
- Attributes of the activities, such as the Advantages, Complexity, and Condition for the usage. This information provides help for the selection and tailoring of the activity.

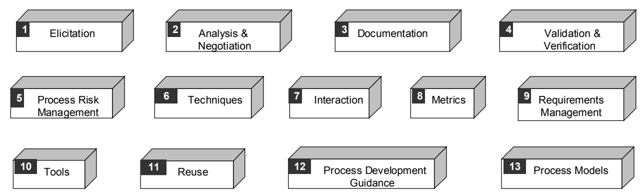


Figure 2 RE Process Building Blocks in REPKB

As an example, the structure of the Elicitation Building Block is shown in Fig. 3

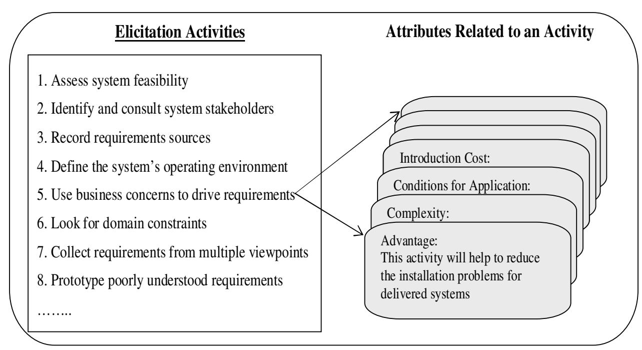


Figure 3 The Elicitation Building Block

The remaining building blocks have slightly different structures. Building block 5 includes requirements process risk indicators and their definitions. Building block 6 includes information about RE techniques. Building block 8 includes commonly used requirements metrics and their definitions. Building block 10 includes information on RE tools. Building block 12 includes guidelines for building block selection and overall RE process development methodologies. Building block 13 includes process models and guidelines for their usage.
The following factors are considered when constructing a building block in the REPKB:

- The building block and the knowledge structure on which it is based should reflect current RE process research and practice. Ideally, the building block should also be used as necessary component of the RE process template.
- A logical relationship between the building blocks and the attributes of the software project should be established whenever possible. This relationship will ensure the suitability of the newly developed process model for the given software project.
- The building block should contain a cohesive set of RE process knowledge. This factor is important since it has a significant impact on the extension and maintenance of the RE process knowledge.

3.1.2 RE Process Template

An RE process template is defined as a pattern which is applicable for most software projects and serves as a template or reference model for the development of an RE process. Our template was developed based on previous research in requirements processes, especially Kotonya & Sommerville's RE process model [22].

The RE process template includes five distinct phases: Requirements Elicitation, Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation, and Requirements Management. Each phase by itself can be seen as a process in its own right [22] and is composed of a number of activities. Requirements Management is carried out in parallel with the other four phases. It considers planning, monitoring and controlling the changes of requirements in the RE process. An "activity" refers to the smallest element in an RE process model.

It is worth mentioning that the RE processes building blocks and the RE process template are complementary to each other. The building blocks provide a mechanism for the construction of RE processes where the RE process knowledge is classified and organized into separate and relatively independent logic components. RE process building blocks are a good choice when the requirements engineer wants to develop an RE process from existing knowledge units. The "bottom-up" approach is taken when building blocks are used. In contrast, the RE process template is a good candidate when the requirements engineer wants to tailor the RE process template to a given project since RE process templates allow people to look at the RE process knowledge from a holistic perspective. This approach is a "top-down" mechanism. Moreover, the RE process template captures essential RE process logic while leaving out operational details that will be provided later by requirements engineers for the specific software project.

3.1.3 Rules

A set of production rules was developed in the proposed framework. These rules are mainly used to represent information and to reason about specific instances in the RE process knowledge base. There are three types of rules in the REPKB: (1) Assent rules that describe which process models and techniques are recommended for certain software projects based on the characteristics of the software projects, process models and techniques. An example of this type of rules is shown in Figure 4. (2) Dissent rules that describe which process models and techniques should not be used for certain software projects based on their characteristics. (3) Consistency rules which are used to examine the consistency of the techniques selected for the newly developed RE process model.

Rule Type: Assent
Logical Operation: AND
Condition 1: Stakeholder Heterogeneity Of ?X Is High
Condition 2: Degree Of The Importance Of Usability Of ?X Is High
Condition 3: Importance OF The Eliciting Implicit Knowledge Of ?X Is High
Condition 4: Customer Availability Of ?X Is High
Technique 1: Ethnography
Action : Recommend To ?X

Notes: ?X indicates a specific software project

Figure 4 Example of an Assent Rule

All these rules are implemented in the FRERE tool using a frame structure. Currently, 125 rules are defined in the REPKB. Derivation of more rules and storing them into REPKB is subject to further research.

3.1.4 Cases

In order to facilitate RE process development, a set of cases (currently there are 21 cases) stating what process models and techniques were used or can be used for a certain type of project were developed based on past industrial projects as well as a survey of experts from both academia and industry. Additional cases 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 the techniques $T_j= \{t_1, t_3\}$ for the same project, then the recommendation would be either $T_i \cap T_j$ if t_2 and t_3 are mutually exclusive (usage of technique t_2 violates the basic principles of technique t_3), or $T_i \cup T_j$ if t_2 and t_3 are not mutually exclusive. Helmer & Rescher state that in imprecise science where no commonly accepted measurements are available, the incorporation of experts' opinions into the investigated subject area is acceptable and can also assure the validity of the result just as in other research [23]. Based on this assertion, we argue that the process used in this research for the derivation of the cases from experts is acceptable.

Examples of this type of cases are shown in Table 2. More information about the cases can be found in [11].

Table 2 Examples of cases in REPKB

Cases	Condition Attributes								Decision Attributes			
	Attributes of Project								Building Blocks	Process Models	Techniques	Tools
	Size of the Project	Team Size	Require-ments Volati-lity	Project Category	Degree of Safety Criticality	Quality Standard	Product Type	…				
1	Small	10	High	Organic	Low	Medium	New	…	1. Analysis and Negotiation 2. Techniques 3. RE Process Models	1. Xtreme Programming Process 2. Any One of Agile Process.	**Elicitation:** 1. Customer Online 2. Prototyping 3. Concept Map 4. Focus Groups **Analysis and Negotiation:** 1. OO analysis 2. AHP 3. UML 4. SASD **Documentation :** 1 UML-Based 2. Text-Based **Verification and Validation:** 1.Customer Online 2.Technical Review	Any requirements documentation tool
2	Medium	40	Low	Embedded System	High	High	New	…	1. Elicitation 2. Analysis and Negotiation 3. Documentation 4. Verification and Validation 5. Techniques	1. KAOS Process Model 2. Volere Process Model	**Elicitation:** 1. Interviewing 2. Focus Groups 3. JAD 4. Observations **Analysis and Negotiation:** 1. QFD 2. Formal Modeling 3. OO Modeling 4. Goal-Based Analysis 5. Viewpoints Analysis 6.Use case Modeling **Documentation :** 1.Formal Notation 2.Viewpoints-Based 3.Use Case-Based **Verification and Validation:** 1. Technical Review 2. Scenario Approach 3. Verification with Formal Language 4. Requirements Testing	1. Tools such as DOORs, Requisite Pro 2. Tool for verification if formal notations are used
…	…	…	…	…	…	…	…	…	…	…	…	…

It is worth mentioning that all the information derived from the cases can only serve as the initial recommendation space for the RE process model and techniques. Further refinement of the recommended solution is necessary to ensure the final solution is suitable for the given software project.

3.2 RE Process Development Methodology

The methodology part in Fig. 1 includes three methodologies to aid RE process development and techniques selection. We will present the RE Process Development Methodology (REPDM) in detail later in this section. The other two methodologies are summarized in the following:

- Methodology for RE Techniques Selection (MRETS) [24]. MRETS makes use of a clustering method and experiences from requirements engineers to analyze RE techniques. Based on the computation of the objective function, the methodology provides a decision support mechanism to support RE techniques selection for a project based on its characteristics.
- Process Development Assistant methodology (PDA) [25]. PDA provides means for RE process development. It is based on Pohl's 3-dimensional framework for requirements engineering [26], the RATS methodology [27], and vector theory. One of the novelties of PDA is that it goes beyond the limitation of 3-dimensional RE processes by allowing process development in an N-dimensional space in which vector theory was used to calculate the shortest path (assumed to be the optimized sequence of activities) in the (process) space.

The REPDM methodology developed in this research provides stepwise guidance for RE process development [11]. The REPDM consists of a set of meta-processes and guidelines that provide recommendations on how to proceed in RE process development.
The methodology includes the following essential steps:

1) Scoring the project attributes based on the initial estimation of the characteristics of the software project. This estimation can be based on the perception of the software project and/or past experiences.
2) Case-based reasoning to retrieve the recommended building blocks, techniques, process models and tools for the given project from REPKB.
3) Selection of RE process development strategies, building blocks, techniques, process models and tools for the given project based on the recommendations derived in step 2).
4) RE process development. In this step, requirements engineers can choose the following strategies in the development process:
   - Develop a process model using components from different process elements. This is common practice in process engineering, and method engineering [13, 28-29]. Using this option, the requirements engineers are required to make further selections based on the recommendation from Step 2, such as selecting one process model as basis for further process development or selecting a set of building blocks from the recommendation by considering the characteristics of the given project.
   - Develop a process model by tailoring the RE process template [13]. This option is chosen when requirements engineers consider the process template helpful for developing a process model

after evaluating the recommendations given in Step 2 by using the REPKB. The requirements engineers are required to instantiate the RE process template to a level suitable for the given software project and then select or define suitable practices for the new process model to the project.
- Process development from scratch [13]. This option is chosen when no other process model is suitable. A new process model for a given project is developed using PDA and the knowledge from the REPKB.

To facilitate RE process development, seven operations are defined to help RE process development. These operations are Tailoring, Selecting, Adding, Deleting, Splitting, Merging, and Modifying. The operations are well-defined mechanisms that help reuse existing knowledge of the RE process supported by the FRERE tool.

5) Assessment of the newly developed RE process model. In this step, the newly developed RE process model is assessed using the assessment models defined in the framework. The assessment models will be introduced in the next subsection.

3.3 RE Process Assessment Models

There are three RE process assessment models in the framework:

1) The RE Process Maturity Model (REPMM), which is a model similar to the CMM model but aimed at RE. It was proposed by Sommerville & Sawyer [31].
2) The RE Technique Suitability Assessment (RETSA) model. This model provides information about the suitability of a technique for a software project based on the attributes of the project [16]. It was developed based on knowledge about RE techniques contained in literature, and the experience of software developers and experts from both industry and academia using a survey. This model takes the software project attributes and suitability of RE techniques with respect to the characteristics of software projects into consideration. The essential elements in the model are a technique Suitability Assessment Matrix (see Table 3) and a function which is used to calculate the overall suitability of the selected RE techniques.
3) The major Concerns of Requirements Engineering (COREs) assessment model [30]. In this research, a "CORE" is defined as a specific interest or objective of the RE process which needs to be addressed to ensure the development of a well-defined, high-quality requirements specification that is complete, concise, unambiguous and consistent. A set of major COREs was developed based on a survey of experts from industry and academia [30]. For example, "Effective communication", and "Elicitation of non-functional requirements and system constraints" are two major

Table 3. An Example of Technique Suitability Assessment Matrix

Attributes of Project		Unified Modeling Language(UML)	SDL	User Story Card
Project Size	Very big	1	0.5	0
	Big	1	0.75	0.25
	Medium	1	1	0.5
	small	0.75	0.75	0.75
	very small	0.5	0.5	1
Project Complexity	Very High	1	0.5	0
	High	1	0.75	0.25
	Medium	1	1	0.5
	Low	0.75	0.75	0.75
	Very Low	0.5	0.5	1
Requirements Volatility	Very High	0.5	0	1
	High	0.75	0.25	0.75
	Medium	1	0.5	0.5
	Low	1	0.75	0.25
	Very Low	1	1	0

Notes: Specification Description Language (SDL), Unified Modeling Language (UML) and User Story Card are three requirements documentation techniques.

COREs of the RE process. The list of major COREs proposed in this research covers all aspects of RE principles discussed in [22, 31, 32]. Currently, the major COREs-based RE process assessment model includes two methods:

- Overall assessment method which provides an assessment of the overall RE process.
- Category-based assessment method which targets each phase of an RE process and encourages RE process improvement in each phase.

These three models are complementary to each other and can be used to evaluate user-developed RE process models within the context of a specific software project. The CORE model assesses the RE process model from the perspective of best engineering practices [30]; RETSA assesses the process model considering the characteristics of the software project; the REPMM model assesses the RE process model at a lower level of abstraction [31] considering various standards.

4 A Case Study

In this section, we describe a real industrial software project that was carried out using the FRERE framework. We compare this project with a previous project that did not use the FRERE framework.

In summary, the case study followed systematic steps for conducting case studies [33]. Due to the page limit, we will only present the major parts of the case study. Interested readers can refer to [11] for more information.

4.1 Develop a hypothesis and selection of the Pilot Project

The hypothesis for our case study was formally defined as: *Using an RE process model developed with the FRERE framework rather than using ad hoc RE practices has a* *positive impact on the overall quality of the software requirements specification and the software project.*

The hypothesis can be considered true if the following items are met:

- The RE process model developed using the FRERE framework is considered suitable for the given software project in terms of scale and complexity.
- RE practitioners agree that the RE process model developed has a positive influence on the project when compared to ad hoc RE practices.
- When compared with previous projects, an RE process model developed using FRERE results in a higher quality requirements specification.

To verify the hypothesis, we carried out a case study in a software company X (the name of the company is withheld for reasons of confidentiality). This is a company that is CMM level 2 certified and its management is committed to software process improvement. The RE processes in the company had been ad hoc in most previous software projects which is one of the barriers for achieving a higher CMM level. With management support from the company, we were able to select a software project called Port Scheduling System (PSS) as the test bed for the case study. The PSS project was compared with another project, called Intelligent Industrial Waste-Water Treatment System (IWTS), which had similar project characteristics and was previously carried out in company X. One of the features of the IWTS project was that the RE process in this project was ad hoc; therefore, it can serve as a sister project for the PSS project.

4.2 Summary of the Case study

The PSS project was carried out over about one and a half years during which the authors were involved in the software project to provide guidance and help.

The major steps carried out during the case study are summarized below:

(1) Scoring the attributes of the PSS project (see Table 4)

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. The objective was to ensure that the attributes of the software project were considered during RE process development.

(2) Case-based reasoning (CBR)

In this step, CBR was carried out using the REPKB. The result of the reasoning was a project case with project attributes similar to those of the PSS project. The solution part of the retrieved case, denoted as P_{IR}, included a set of building blocks, RE process models, techniques, and tools for the new project. In our case, P_{IR}, as the initial recommendation, is shown in Table 5.

Table 4 Project Definition

Project Description	This project is to develop a "Port Scheduling System". The objective of the system is to schedule a container terminal with a throughput of maximum 1 million TEU (twenty foot equivalent unit) each year. The terminal must also be able to hand the smaller cargos. The project requires a highly interactive interface……
Project Attributes	**Project Size**: Medium (>=700 and <1200 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): 49
	Degree of Knowledge of Requirements: Medium
Product Attributes	**Degree of Safety Criticality**: High
	Quality Standard: High
	Product Type: New

Table 5 Recommendations based on CBR

Building Block	Process Model	Techniques
• Elicitation • Analysis and Negotiation • Documentation • Verification and Validation • Interaction Management • Requirements Management • Techniques • Tools • Process Risk Management • Requirements Metrics and Process Metrics • RE Process Models	VORD Volere	T_e **:** Interview, JAD T_a **:** Viewpoint-Based Analysis, Scenario-Based Analysis (Use Cases), AHP, T_d **:** Viewpoint Definition, Structured Natural Language Specification T_v **:** Viewpoint-Based Validation, Formal Requirements Inspection Tools: RequisitePro Rational Rose

Notes: (1). T_e:, T_a, T_d, T_v stand for the initially recommended Elicitation, Analysis and Negotiation, Documentation, and Verification and Validation techniques respectively.
(2) VORD: Stands for Viewpoints Oriented Requirements Definition, it is a RE process model. Viewpoint-Based Analysis is a technique which can integrate Object-Oriented modelling mechanisms into a RE process [22]. (3) Volere is the name of an RE process model [3]

(3) Selection based on the recommendation

In this step, the requirements engineers decided to adapt the process model recommended in step (2) and augment it with other selected requirements techniques, tools, and activities from the chosen building blocks. The selected process model, techniques, building blocks, and tools are listed in Table 6. It is worth mentioning that, in this step:

- The process model selection is based on the experiences and the knowledge of the requirements engineers supported by REPKB.
- Techniques selection follows the MRETS methodology through a process of clustering techniques, looking for comparable and complementary techniques, combination of the techniques, computation of the objective function, and making the decision as listed in Table 6.

Table 6 Selection based on the recommendation

Building Block	Process Model	Techniques
• Elicitation • Analysis and Negotiation • Documentation • Verification and Validation • Interaction Management • Requirements Management • Techniques • Tools • Process Risk Management • Requirements Metrics and Process Metrics • RE Process Models	VORD	T_E : Focus Group, Interview, Enthnography T_A : Viewpoint-Based Analysis, T_D : Viewpoint-Based Definition T_V : Formal Requirements Inspection Tools: DocManager

Notes: T_E:, T_A, T_D, T_V stand for the final recommended Elicitation, Analysis and Negotiation, Documentation, and Verification and Validation techniques respectively.

- An in-house requirements management tool called "DocManager" was selected and used in the project rather than the recommended tools RequisitePro and Rational Rose since the in-house tool is cheaper, and the company had a lot of experience using this tool even though it is less powerful than the recommended commercially available requirements management tools.

(4) RE Process Development

Since the RE techniques had already been selected, RE process development at this stage mainly refers to the definition and selection of activities from the chosen VORD process model, techniques and the building blocks. The VORD process model served as the basis for RE process development, i.e., the activities of the VORD process model were either directly selected or modified for use in the new RE process model for the PSS project. New activities were defined based on the experiences and discretion of requirements engineers. All activities were defined with the General Activity Definition Template (GADT). A screen shot of the activities definition in the PSS case study is shown in Fig. 5. With the help of REPKB, the PSS RE process was finalized and the activities defined are shown in Fig. 6.

1. Consider the system stakeholders
2. Identify system operators and their roles in the system, as well as those principal individual who might be associated with that class.
3. Identify the viewpoints by referencing to the viewpoints template
4. Hold focus group meeting to elicit requirements from various stakeholders
5. Record the source and rationales of requirements
6. Consider the system constraints and non-functional requirements
7. Prioritizing requirements
8. Modeling viewpoint service behavior (functional requirements modeling)
9. Model user interface requirements as constraints on viewpoint services
10. Identify and analyze the safety related requirements
11. Identify the conflict and resolve the conflict by negotiation with stakeholder
12. Develop test cases for key functional requirements.
13. Identify and analyze the relationship of requirements.
14. Document viewpoint requirements with the template of viewpoints
15. Define terminology in the system level as well as all levels if applicable
16. Document functional requirements from all viewpoints with UML-liked notations
17. Document non-functional requirements from all viewpoints with structured language.
18. Document the test case for key function requirements
19. Document the relationship of requirements.
20. Check requirements documentation by using defined checklist.
21. Check document to identify the unambiguous, conflict and/or not implementable requirements
22. Identify and resolve interactions
23. Verify and validate the safety requirements of the system.
24. Identify and document requirements with high volatility
25. Use requirements baseline techniques to management requirements.
26. Define a set of metrics of requirements and RE process that are relevant to the project
27. Manage and monitor the RE process by using the defined metrics
28. Use tool throughout the whole project

Figure 5 Activities in the PSS RE process model

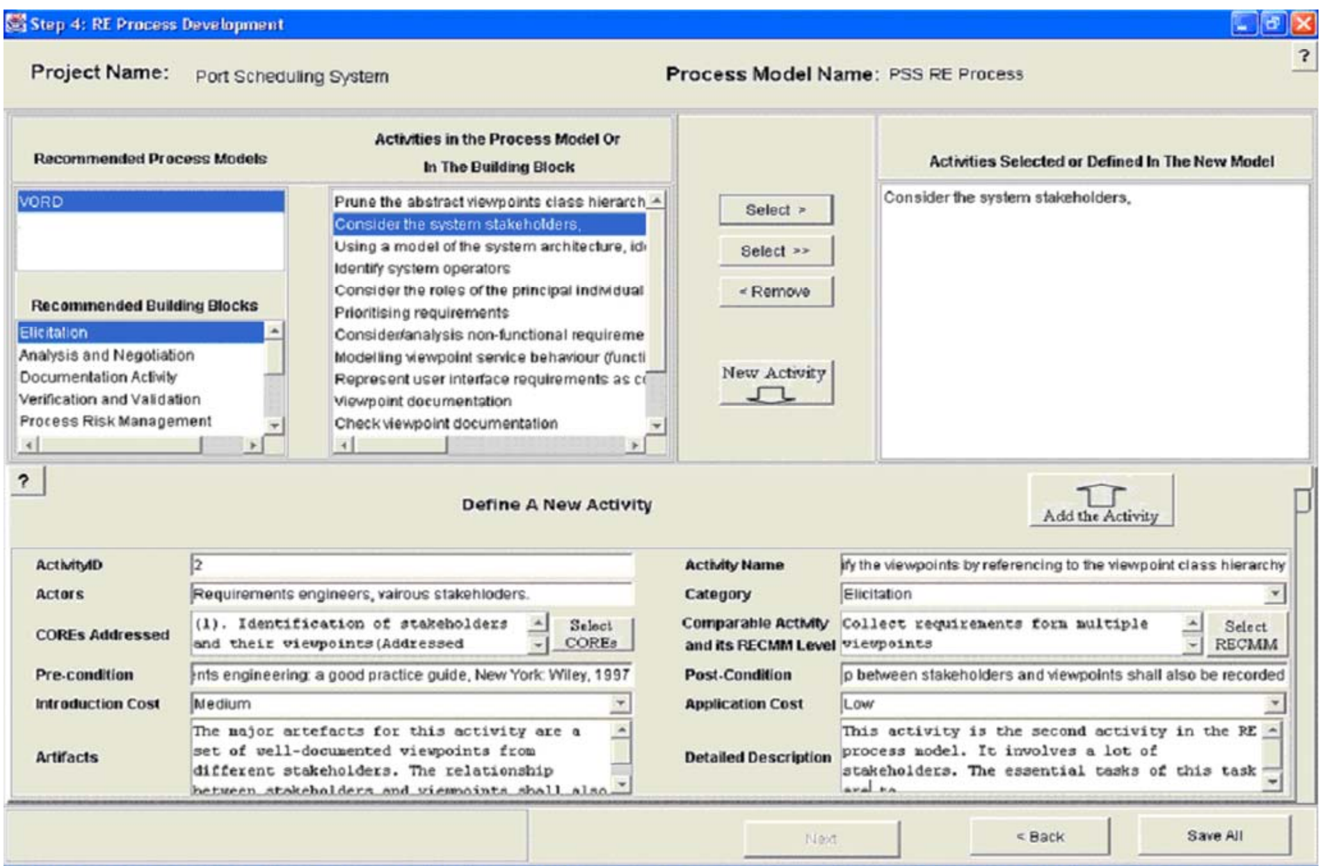


Figure 6: Definition of an activity in the PSS project

Table 7 Data collected from the current project and a previous project

The Recorded Data of the Major Variable	PSS	IWTS
Total number of atomic requirements	882	702
Number of analysts involved	4	4
Number of developers involved	49	51
% of new requirements elicited using recommended RE techniques	56.2%	37.9%
% of requirements changed after start of design	7.8%	39.5%
% of requirements discovered during testing	1.6%	9.3%
Project duration: planned	18 months	16 months
Project duration: actually spent	19.5 months	21 months

4.3 Quantitative and Qualitative Analysis of the Software Project Results

In this subsection, only part of the quantitative and qualitative data is presented and analyzed due to the page limit. Please refer to [11, 24] for more information

(1) Quantitative Analysis

The data collected during the PSS project was compared with the IWTS project that did not use our FRERE framework. Table 7 compares briefly the two projects carried out by the same team except that two junior developers were not part of the PSS project. Even though the PSS project had about 25% more requirements than the IWTS project, it required less development time. Furthermore, the PSS project was only 8.33% overtime, while the IWTS project was 31.25% overtime in terms of person-months. The likely reason for this is that the RE process used in the PSS project helped discover and correct more requirements problem earlier on than was the case in the IWTS project, which experienced very late requirements changes. A key success indicator is that no major requirements that would have had a significant impact on the overall system structure or on major functionality was added or deleted after the completion of the requirements engineering process. The training cost of selected techniques was only related to the Ethnography technique since the requirements engineers involved in the project were already familiar with the other techniques.

(2) Qualitative Analysis

In addition to the quantitative analysis presented in the last section, a questionnaire was conducted among all the developers, requirements engineers as well as managers who were involved in the PSS project. The objective of the survey was to get further feedback about the usage of FRERE. Some of the questions used in the survey are shown in Table 8 together with the data collected from the developers. In summary, the average percentage of "Disagree" and "Strongly disagree" combined is only about 12%; while more than 66% of the respondents chose "Strongly agree" and "Agree". If we also include the number of respondents that selected the answer "Medium", then 88% of the developers agree that the RE Process of the PSS project is beneficial. This information serves as further evidence of the positive effects of the developed RE Process on the PSS project.

Table 8 Qualitative data from survey

Questions for the Survey	SA	A	M	D	SD
(1) The RE techniques can be understood easily	23	19	14	3	1
(2) The frequency of the requirements changes is reduced with comparison to the software project 3F SYSTEM and similar projects done before.	21	23	12	2	2
(3) The requirements are more understandable than before	20	19	17	3	1
(4) The notations used in the specification are acceptable and easily understandable	18	24	13	3	2
(5) The specification is easily traceable	19	22	15	3	1
(6) The RE techniques used in the project were very helpful in reducing the overall delay of the software project	22	19	14	4	1
(7) The overall quality of the documentation is high	19	21	15	3	2

Notes: SA, A, M, D, SD represents the number of people selection of Strongly Agree, Agree, Medium Agree, Disagree, Strongly Disagree respectively

(3) Discussion

The quantitative and qualitative analysis presented above shows that company X was able to develop a much better requirements specification with more precise definitions, clearer structure and traceability using the developed RE process than was the case in the IWTS project which used an ad-hoc RE process.

We acknowledge that comparing the data from the two projects cannot be used as formal proof that our framework will always provide the best solution for a software project. Numerous other factors reduce the validity of the case study such as the differences of the management commitment, learning effects and training, the available project data, and personal attitudes and experiences in the application of the FRERE framework etc. Nevertheless, we argue that based on the case study it is possible to state that:

- The FRERE framework provided significant help during the development of the RE process of the PSS project and aided the selection of RE techniques.
- The RE process model developed with the help of the FRERE framework can be considered as a well-suited RE process model for the PSS project.
- The framework can be very useful to help develop a project-specific RE process model for other software projects within company X.

We acknowledge that the case study does not have a well-understood theoretical basis from which one can draw strong conclusions. However, it is widely believed in the SE domain that real-life case studies are suitable for an industrial evaluation of SE techniques and tools if they are organized and conducted in a sound way [34].

5. Conclusions and Future Work

This paper proposed a framework which is built on our research on RE processes, techniques and process engineering theory over the last six years. The objective of this research is to bridge the gap between RE research and practice by constructing a theoretically sound and practically feasible framework that can help develop a suitable RE process for a particular project. The major characteristics of the FRERE framework are that it addresses several issues of the RE processes: engineering issues, domain issues, characteristics of the software project, and issues of conformity to software process standards. Our case study showed that the RE process model developed with the FRERE framework had a positive impact on the overall success of the software project.

Our future research will focus on the following aspects:

- Further refinement of the FRERE framework.
- Extension of the REPKB such as including more RE techniques, RE process models, software project cases, and elicitation and definition of more rules based on a more detailed analysis of RE techniques and process models.
- Application of the FRERE framework to different software projects to examine the merits and effectiveness of the FRERE framework.
- Complete implementation of the FRERE tool.

References

[1] Boehm BW (1981), SE Economics, Prentice Hall, 1981

[2] Macaulay LA (1996), Requirements Engineering, Applied Computing, Springer, 1996

[3] Robertson S. and Robertson J. (1999) Mastering the Requirements Process, Addison-Wesley

[4] Emam KE, Birk A (2000), Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability, IEEE Trans. on SE, Vol 26, NO. 6, Jun. 2000.

[5] Jiang L and Eberlein A (2003) RE: A Review and A Proposal, Proceedings of the Third ASERC Workshop on Quantitative and SE, Feb. 17-18, 2003, Banff, Alberta, Canada.

[6] Roman G (1985), A Taxonomy of Current Issues in RE. Computer, pp. 14-22, April 1985

[7] Jones C (2000) Software Assessment, Benchmarks, and Best Practices. Addison-Wesley

[8] Glass RL (2004) Matching Methodology to Problem Domain, Communications Of The ACM, Vol. 47, No. 5, pp. 19-21

[9] Damian, D.; Chisan, J (2006) An Empirical Study of the Complex Relationships between RE Processes and Other Processes that Lead to Payoffs in Productivity, Quality, and Risk Management, IEEE Transactions on Software Engineering, Vol. 32, Issue 7, July 2006, pp:433-453

[10] Holt J (1997) A Survey on Current practices in SE, Computing & Control Engineering Journal, Vol. 8, Issue 4, Aug. 1997, pp: 167 - 172

[11] Jiang L (2005), A Framework for RE Process Development, PhD Thesis, University of Calgary, Sep. 2005

[12] Oquendo F., Lecture Notes of Computer Science, 9th European Workshop on Software Process Technology (EWSPT-9) EWSPT 2003, held in Helsinki, Finland in September 2003

[13] Wang Y, King G. (2000), Software Engineering Processes: Principles and Applications, CRC Press, 2000.

[14] Kasanen E., Lukka K. & Siitonen A. (1993) The Constructive Approach in Management Accounting Research, Journal of Management Accounting Research, No. 5, pp.243-264

[15] Harrison K. and Fahy K. (2005). Constructive research: methodology and practice. In Tenenbaum G. and Driscoll M. (Eds.), Methods of research in sports science: quantitative and qualitative approaches, pp. 660-700. Oxford, U.K.: Meyer and Meyer.

[16] Jiang L, Eberlein A (2007) A Case Study For Technique Suitability Assessment Based on Project Attributes, 14th Annual IEEE International Conference on the Engineering of Computer Based Systems, Accepted, 2007 (ECBS)

[17] Pfeiffer S. (2003) RE for Dynamic Markets, MSc. Thesis. Department of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta

[18] Mannion M. and Keepence B. (1998) Using Viewpoints to Define Domain Requirements, IEEE Software, Jan. to Feb. 1998, pp. 95-102

[19] Davis A. M. (1993) Software Requirements, Objects, Functions and States, Prentice Hall

[20] McPhee C. (2000). RE for Projects with Critical Time-To-Market, MSc. thesis, University of Calgary.

[21] Yourdon E. (1997) Death March The Complete Software Developer's Guide to Surviving "Mission Impossible" Projects, Prentice Hall PTR

[22] Kotonya G. and Sommerville I. (1998) RE, Processes and Techniques, John Wiley & Sons Ltd, 1998

[23] Helmer O. and Rescher N. (1959) On The Epistemology of The Inexact Science, Management Science, Oct. 1959, Vol. 6 Issue 1, pp. 25 - 28

[24] L. Jiang, A. Eberlein, B. H. Far, and M. Mousavi, A Methodology for the Selection of Requirements Engineering Techniques, Journal of Software and Systems Modeling. DOI 10.1007/s10270-007-0055-y, 2007

[25] Eberlein A. Jiang L (2007), Description of a Process Development Methodology, Journal of Software Process: Improvement and Practice, Vol. 12, Issue 1, Jan 2007

[26] Pohl K. (1996) Process-Centered RE, John Wiley & Sons

[27] Eberlein A. (1997) Requirements Acquisition and Specification for Telecommunication Services, PhD Thesis, University of Wales, Swansea, UK

[28] Kumar K. and Welke R. J. (1992) Methodology Engineering: A Proposal for Situation Specific Methodology Construction, Challenges and Strategies for Research in Systems Development, Cotterman, W. and J. Senn (eds.), Wiley, Chichester, UK, pp. 257-266.

[29] Brinkkemper S., Lyytinen K. and Welke R. (1996) Method Engineering, Principles of Method Construction and Tool Support, Chapman & Hall

[30] Jiang L., Eberlein A. and Far B. H. (2004), Evaluating the RE Process Using Major Concerns, the Proceeding of the IASTED International Conference on Software Engineering, Feb. 16-19, 2004, Innsbruck, Austria

[31] Sommerville I. and Sawyer P. (1997) Requirements Engineering: A Good Practice Guide, New York: Wiley

[32] Firesmith D. G. (2005) What Are the Principles of RE? RE Online-Discussion-Forum, April 27, 2005.

[33] Kitchenham B., Pickard L. & Pfleeger S. L. (1995) Case Studies for Method and Tool Evaluation, IEEE Software. pp. 52-62

[34] Wohlin C., Runeson P., Ohlsson M. C., Regnell B. and Wesslen A. (2000) Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers