Requirements Engineering Education for Senior Engineers: Course Design and its Evaluation

Takako Nakatani
Graduate School of Business Sciences, University of Tsukuba, and National Institute of Informatics, Tokyo, Japan
nakatani@gssm.otsuka.tsukuba.ac.jp

Toshihiko Tsumaki
National Institute of Informatics, Tokyo, Japan and Nihon Unisys, Ltd.
tatsumaki@nii.ac.jp

Tetsuo Tamai
Graduate School of Arts and Sciences, The University of Tokyo, Tokyo, Japan
tamai@graco.u-tokyo.ac.jp

***Abstract*—Offshore development has become common for software companies. Such companies have started to concentrate their engineers' efforts on the early software development phase. Within this context, it is important to educate senior engineers to master all aspects of requirements engineering (RE). Though it is not easy to master the various aspects of RE all at once, we can expect engineers to study methods on their own if they believe that learning these methods is important for them or for their projects. We have developed a two-and-a-half-day role-playing workshop for engineers that focuses on teaching the importance of RE, the background, rationale, and purpose of the requirements, as well as the actual requirements. The target students of our course are senior engineers with over ten years experience in software development. We first give an overview of RE techniques, and then, introduce the instructional design of our course for senior engineers. We also present the results of an actual workshop, which showed that engineers could learn worldviews of clients' requirements and the importance of the "why" of requirements.**

***Keywords*-requirements engineering; engineers' education; role-playing workshop; requirements engineering map;**

I. Introduction

Offshore development has become common for software companies. Such companies have started to concentrate their engineers' effort on the early software development phase. The software industry is facing a shortage of talented engineers. If software vendors can procure offshore developers, they can shift in-house engineers from the middle and/or later development phases to the early development phase.

One problem is that engineers, including managers, still focus on the middle to later development phases. The importance of requirements engineering (RE) has been recognized since Boehm's book in 1981 [1]. His work showed that correcting requirements errors on large projects is 100 times more expensive in the later development phases than in the requirements phase. However, most engineers do not prioritize RE because they are mainly motivated to develop software, rather than software requirements specifications (SRSs).

There are engineers who produce a thick SRS from a large amount of information obtained as stakeholders' actual requirements, but they can also produce the same volume of SRS from a small set of stakeholders' requirements. This means that engineers can form SRSs independently of the stakeholders' needs. What makes this aspect of engineering culture even worse is that a reviewer cannot distinguish the latter from the former. If the rationale for each requirement is described in the SRS, the reviewers may be able to distinguish one from the other. Therefore, SRSs should contain descriptions that can help reviewers understand the background and/or purpose of the requirements. In some software development companies, SRSs rarely include descriptions about the "why" of requirements and few engineers care about its importance. Engineers and reviewers devote their attention to "what" and "how" to build their software. In order to shift in-house engineers from the middle and/or later development phases to the early development phase, the engineers should learn the importance of "why" of requirements.

Then, how should we educate engineers to have interest in RE and motivate them to write proper SRSs? We have developed an educational course for senior software development engineers. The purpose of the course is to teach them the importance of RE, especially the importance in understanding the background and rationale of requirements, since these aspects enable us to cope with the volatility of requirements.

In Japan, most requirements engineers have approximately ten years development experience. Therefore, the target students of our course are senior engineers with over ten years of experience in software development. The students are selected by their supervisors who want them to be well educated in RE. We assume that such students have programming, design, and team-work skills.

There is a cultural problem in educating senior engineers. Such engineers will make an effort to master a new technique only if it looks useful to them. The purpose of our educational course is to motivate them to apply RE techniques to their job. Therefore, the course has been designed to teach students the usefulness of RE techniques.

In our previous work [2], the course was not adequately evaluated. To design the instruction of the course, we researched RE education styles, the RE map introduced by Tsumaki and Tamai [3], and RE methods. Nakatani et al. described the details of the instructional design [4]. The main purpose of this paper is to discuss the evaluation of this course.

The remainder of this paper is organized as follows. Section 2 introduces related work of RE education, and Section 3 defines the instructional design of the course. We also discuss the hypothesis for the instructional design, the course products, method selection, and style selection of the course in this section. Section 4 introduces the course evaluation. The last section presents our future work and conclusions.

II. Related Work

A number of RE educational styles have been developed, and each style is useful in their own way [5].

The basic style is a lecture-based course. This style is acceptable for an introduction to RE [6], covering all the key issues in RE to introduce how to use RE.

Another style focuses on the practical use of RE. It was developed for undergraduate students to provide them with the specific abilities/skills to face the demands of the professional software engineering market [7]. This 16-hour course is composed of several modules, i.e. software quality, requirements documentation, a requirements elicitation workshop, quality analysis techniques and methods, and tools.

Berenbach et al. have introduced a unified requirements model that integrates features, use cases, and hazard analysis [8]. For educating engineers, teaching multiple methods is more effective than just teaching a single method. Furthermore, the relationships between the methods should be taught to better understand them and plan for updating the students own methods.

Most RE educational courses regard student participation as important. Therefore, game-based [9], [10], role-playing [11], experiment-based [12], investigation-based [13], and/or workshop styles are rewarded with good results. Game-based training may be attractive to most students. However, we do not apply game-based training, because we require students to evaluate RE techniques in a more realistic situation.

Students should understand their clients' points of view. In this sense, a role-playing exercise can be an appropriate style for learning RE [11]. One of the strong points of a role-playing style exercise is that students can experience different organizational contexts. As a result, they can personally know others' worldviews of requirements. Such an experience is important for engineers because their real roles are always as developer, and they have fewer chances to appropriately consider their clients' worldviews in their daily lives. We decided to apply role-playing exercises to help the students better understand their clients' worldviews.

There are several RE courses developed for professional engineers. Siemens has several RE courses for engineers worldwide [14]. The first one, focusing on the foundation of RE, is conducted as a three-day lecture-based educational course. After the course, the students are required to answer questions concerning the frequency and usage of the techniques that they learned during the course. The most frequently used technique was requirements writing. Therefore, we included requirements writing in our course. Simmons [15] recommended that further education through corporate training should include a lecture on the company's culture and specific company requirements, e.g., safety requirements, hardware constraints, and/or tools environment. Simmons also pointed out that the most typical problem for professional engineers is the time limitation. The length of the course was usually less than a week, and on average was only two or three days. Therefore, follow-up mentoring is required to strengthen the adoption of new practices and to help ensure first-use success. Our course is followed by a lecture on the company's specific constraints, tools environment, and culture. The students have to give a presentation about their results and conduct an evaluation several months after the course.

The distinctive feature of RE education for engineers is its viewpoint of usefulness in an actual business environment. It may be easy to teach professional engineers to think in a business context, since most students are involved with business-related issues. However, it is not easy to get them to focus on business goals rather than technical goals. Therefore, RE education cannot bypass business aspects [16].

III. Instructional design of course

A. Hypothesis for instructional design

The course is designed to teach students the importance of the following issues:

- understanding RE activities to grasp the background, purpose, and rationale of requirements,
- grasping the situation and worldviews of stakeholders to understand real-world problems,
- exploring and considering alternative solutions and trade off relations among the solutions to efficiently negotiate,
- identifying the variability and commonality of requirements, as well as the likelihood of requirement volatility to properly design software architecture, and
- defining requirements for achieving goals of stakeholders without meeting unexpected side effects, e.g., causing hazardous, injurious, and/or inefficient environment in the real-world.

Every RE technique addresses and offers solutions with regard to the list above. however, they only contribute a part

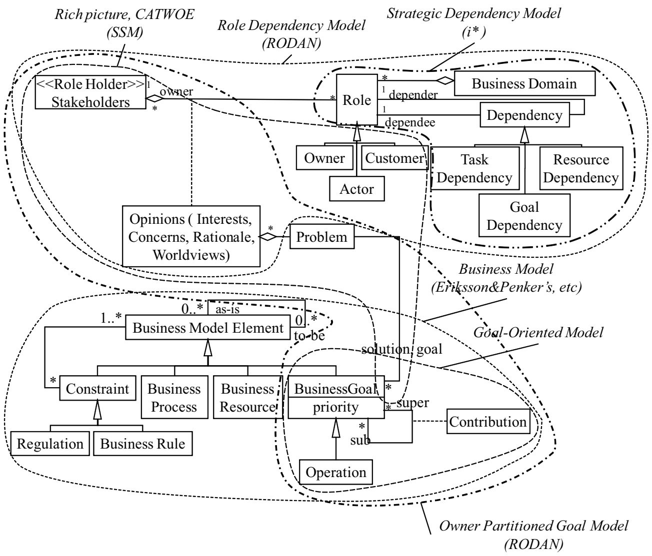

Figure 1. Metamodel of RODAN from product perspective

of the issues. Therefore, we have selected techniques and integrated them to solve all those issues. The techniques for the course should provide models that represent the background of stakeholders' requirements including each stakeholder's worldviews, situation, environment, problems and roles in their organization. To evaluate the outcome of the course undertaken, we define the following hypothesis.

- If an engineer has learned the importance of practicing RE, he/she can focus on the "why" of every requirement.

The remainder of this section presents the instructional design of our course, including the products, methods, educational style and process.

B. Course products

The product of RE is an SRS. RODAN [17], [18], an integrated RE method [19], provides the metamodel shown in Fig.1. The dotted circles represent the integrated methods. The name of the model produced by each method is shown in italics. The metamodel represents the structure of requirements in the context of the real-world from the product perspective of RE activities.

It provides the following information, which an engineer must grasp to elicit requirements.

1) Each stakeholder's worldview of his/her requirements
   - Each stakeholder's roles and responsibilities in the organization
   - Each stakeholder's interests, and opinions
   - Each stakeholder's problems evaluated from his/her worldview
2) The goals of the business organization
   - Achievable goals that can solve the problems
   - Problematic as-is situations and expected to-be situations
   - Solutions, including alternatives to achieving each goal
3) Constraints to verify the selected solutions

According to the metamodel, the software requirements can trace their rationale backward to the problem via solutions that are included in the to-be business model. Thus, the forward tracing path from the stakeholders and their roles in finding solutions via their problems, opinions, and goals is regarded as the rationale of the requirements. The path shows how the software requirements affect the stakeholder's business.

Again, the purpose of our course is to help engineers understand the importance of RE activities and motivate them to learn RE techniques to grasp the stakeholders' situations. Students are expected to produce a model that visualizes the requirements and their rationale.

If they can understand the causes and effects of requirements changes through the model, they will gain the motivation to learn RE and its techniques to prevent requirements changes in their projects. The model is also effective for cooperative work. If the problems of stakeholders can be visualized in a model, the stakeholders can share them. Understanding and sharing others' problematic situations is the starting point for negotiations.

Grasping the "why" of requirements means that engineers should understand the real-world, which is essential to integrate a business analysis and RE [20]. The metamodel shown in Fig.1 is one of our guides for selecting which methods to teach in the educational course. When we selected RE methods, we placed the expected products that the students of our course would produce in the metamodel. Furthermore, the SRS becomes more reliable if engineers can answer the following questions.

1) Who needs the solutions and why? (Personal perspective)
   Requirements viewed from a personal perspective need to be evaluated in the organization.
2) Why should the solutions be highly prioritized? (Organizational perspective)
   Requirements viewed from an organizational perspective require negotiations among the related organizations.

In the next section, we discuss the selected methods for the course.

C. Method selection

Tsumaki and Tamai introduced the RE map [3] shown in Fig.2. They focused on two dimensions in order to characterize the variety of activities of RE techniques: one concerning the elicitation of operational types of activities and the other concerning the target object types of requirements

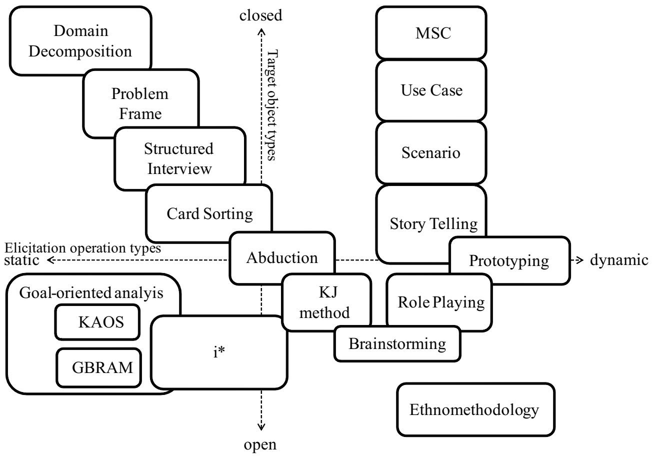

Figure 2. Requirements engineering technology map (RE map) [3]

analysis and/or elicitation. They divided the operational-type dimensions of the RE map into two categories, i.e., static and dynamic, and also divided the object-type dimension into closed and open types.

The first dimension, the operational type, captures how the requirements elicitation and acquisition processes are conducted. A method is positioned within the operational type space by considering how much it requires the imagination or the systematic thinking of the analyst. When an engineer applies a method of *domain decomposition* in order to analyze a domain, he/she can elicit requirements by decomposing the domain systematically. Thus, the method of *domain decomposition* can be regarded as a method with a highly static elicitation operation. DeMarco's structured analysis[21] is one of the methods they apply domain decomposition. On the other hand, developing a prototype requires an engineer's imagination, because the engineer has to elicit a concrete image from ambiguous requirements as a part of the developing system.

The second dimension, the object type, concerns the properties of the target object space to be analyzed, as either relatively closed or open. For example, an engineer who applies a method of *domain decomposition* starts his/her analysis from a concretely defined domain. Therefore, the target object type of such methods is in a closed space. On the other hand, when an engineer develops a prototype, the space of objects elicited by prototyping is sometimes open, and sometimes closed. Thus, the target object type of *prototyping* is placed in the neutral position. *Prototyping* is placed in the center of the right-hand quadrants of the RE map.

According to the characteristics of these dimensions, we can place *Ethnomethodology* into the RE map. The target object type of *Ethnomethodology* is more open, and its elicitation operation type is more dynamic than most methods of RE.

The analysis proceeds from understanding each stakeholder's worldviews, eliciting the goals of the business organization, and checking constraints to verify the selected solutions. Therefore, in the first step of the course, students have to focus on the open type objects and use a dynamic way of thinking. The applicable methods are in the lower-right quadrant of the RE map. In the second step, the students have to think about the solution in order to solve the problematic situation or achieve the business goals. The target object type is still open, however, the way of thinking becomes static on route to lead the solutions. This is the process in which the students can use the static methods, e.g., the goal-oriented methods, in the lower-left quadrant. In the third step, he/she grasps selectable requirements or solutions.

The students analyze the adequacy of the requirements as to whether possible side effects and/or malicious behaviors harm users in the real-world. In these analyses, the target object type becomes closed and the applicable methods anticipate imagination of the analysts [22]. These methods are in the upper-right quadrant of the RE map. Scenario analysis, claim analysis [23], and misuse case analysis [24], [25] are the applicable methods in this step.

After eliciting the requirements, the students must consider the structure of requirements and their adequacy for the software. These activities are static activities, and they treat closed-type objects. For example, methods of *domain decomposition* and *problem frame* are the applicable methods for these activities.

The real requirements elicitation process starts in various situations. In some cases, analysts have to start by understanding a vague, real-world problem. In such a situation, the analyst proceeds with their analysis from the lower-right quadrant to the lower-left quadrant, and then goes to the upper-left quadrant via the upper-right quadrant. In other situations, the analyst can start by verifying the correctness of the provided requirements, starting from the upper-left quadrant.

Our educational course must be designed on the assumption that the target domain is vague. We have to keep in mind that the purpose of this course is to teach the importance of RE.

We selected the following methods based on the RE map and the RODAN metamodel.

- Rich picture:
  This is a tool introduced in Soft Systems Methodology (SSM) by Checkland [26]. In a rich picture, a balloon represents a stakeholders' opinion with regard to a problematic situation. The placing of icons for stakeholders can represent the environment between the stakeholders. In the rich picture, the students are required to draw icons with an expressive face. From the face of the icons, the other students can analyze the stakeholder's subjective emotion. Thus, *rich picture* is

useful for depicting a stakeholder's emotional situation or environment. *Rich picture* treats open-type objects dynamically. Therefore, they can be positioned in the lower-right quadrant of the RE map. Since *Rich picture* uses the interview technique, it can be placed at the upper and left side of *Ethnomethodology* [27] that is closed to the real-world interview and observation.

- CATWOE analysis:
  This is another tool introduced in SSM. *CATWOE* is an abbreviation of *customer*, *actor*, *transformation process, worldviews, owner*, and *environment*. Its implication is "the actor is ordered by the owner to transform the present problematic situation into an accomplishable future situation for his or her customer under the given environment," representing an owner's intention for the future named *worldviews*.
  We divided *CATWOE* analysis into two parts: *CAT+OE* and *W* based on their thinking method. The *CAT* and *OE* are elicited dynamically and are derived from a rich picture. The target object type of *CAT+OE* analysis is more closed than *rich picture* analysis. Its way of thinking is still dynamic, but a little more static than *rich picture* analysis. Therefore, it can be positioned above and to the left of *rich picture* analysis in the lower-right quadrant of the RE map.
  The *W* represents the owner's worldviews. Its analysis starts from the result of *CAT+OE* analysis. The method employs abductive reasoning and so it is placed at the same spot as *abduction*.
- Role dependency modeling:
  This modeling technique extends the strategic dependency model in *i** [28] with three dependency roles. Therefore, it can be placed at the same spot as *i**.
- Owner-partitioned goal analysis:
  This technique extends the goal model of *KAOS* [29] partitioned by the goal owners. Thus, it can be positioned at the same spot as *KAOS* in the lower-left quadrant.
- Theory-W:
  Before analyzing requirements adequacy, engineers and their clients have to negotiate for selecting requirements. *Theory-W* [30] is a method for negotiation on the requirements and constraints of the software development project. In the negotiation, the selectable requirements are discussed in various aspect: e.g. risks, costs and performance of an investment. Therefore, negotiations need a dynamic way of thinking and the target object type is more closed than *CATWOE* and goal-oriented analysis methods. *Theory-W* can be positioned in the upper right quadrant.
- Claim analysis:
  This technique is introduced as a part of the scenario analysis [23] in the upper-right quadrant.
- Misuse case analysis:
  It is positioned as a part of the use case analysis in the upper-right quadrant.
- Object-oriented methods:
  There are various modeling techniques in the Unified Modeling Language (UML) [31]. For example, when we analyze business processes, we can use sequence diagrams. When we analyze business resource structures, we use class diagrams and/or object diagrams [32]. Prior to starting these kinds of analysis, an analyst should have obtained business domain information. In contrast, use case analysis starts from interviews with stakeholders. Therefore, the target object types of object-oriented methods are in a more closed problem space than that of use case analysis. They can be positioned in the upper right quadrant.

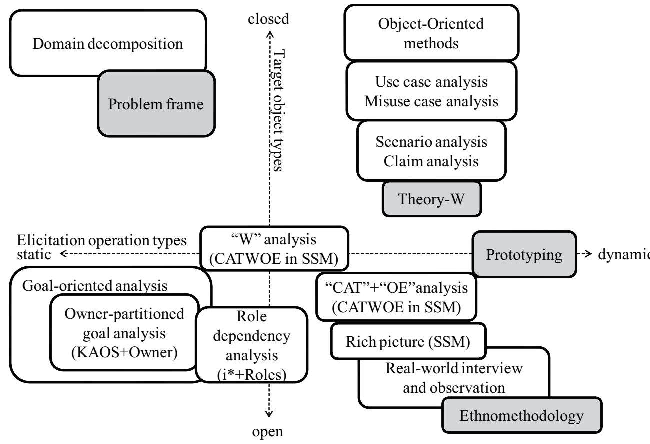

Figure 3. RE map for the educational course

The methods with stepwise refinement are mainly positioned in the upper-left quadrant of the RE map. The target students are familiar with these methods. Therefore, we selected methods from the "open" and/or "dynamic" quadrant. Now, we are ready to define the RE educational map of our course. It is shown in Fig.3. Though the size of each icon is different from the original RE map, this has no representational meaning. Only the relative position of the icon has a meaning. The gray rectangles represent methods that are relatively new and unfamiliar to the students. They are not applied in the exercise but introduced in the lecture.

D. Style selection

As mentioned in the first section, our target students are engineers with over ten years of experience. The students have to recall their clients' worldviews. This is *the* key point when they interview their actual clients. Therefore, all students need to have a chance to experience a client role in the course.

To gain experience of the clients' perspective, role-playing exercises are useful. As we have described in the related work, this is a popular style in RE educational courses. From the role-playing exercises, students can experience various personalities and thus learn other roles' worldviews[11]. We chose a role-playing workshop style in our course to force our students to open their mind to their clients' worldviews. For example, if we organize a class with three teams, each team takes turns at analyzing the same subject. After the students of Team A play a developer role for Team B, they play the client role for Team C. Therefore, once they have interviewed Team B for ten to fifteen minutes, they then switch to the client role to be interviewed by Team C in the following session. The students are required to develop an SRS for their clients, and then, their clients evaluate the SRS.

Each team consists of four to five students, which is a good size for effective discussions and cooperation. It is better if the class consists of more than two teams to prevent the students from interviewing with the interviewed team. Time management is more efficient with an even number of teams.

E. Schedule planning

The duration of the course is limited to eighteen hours (two-and-a-half days). The course is initiated by focusing on personal viewpoints by drawing rich pictures. The course consists of a cycle in each period: lecture, exercise and evaluation.

The first period takes eight hours. In this period, the students are expected to reflect the problematic real-world in their model and answer the first question, "who needs the solutions and why," using their models as evidence. Its aim is to learn the relationships between the stakeholders' viewpoints and their perspectives using a rich picture. The other aim of this period is to extract the organizational goals and requirements using the CATWOE and Role Dependency (RD) analyses.

The second period takes six hours. At the end of the second period, students should be able to answer the questions: "why should the solutions be highly prioritized?" and "why should we directly interview multiple stakeholders?" Its aim is to extract the requirements and their alternatives in order to solve the organizational problems with the Owner Partitioned Goal (OPG) analysis. Another aim of this period is to understand the rationale of the requirements from the CATWOE definitions and the owners' goals. The students are also required to represent the as-is and to-be situations from the structural, dynamic and collaborative perspectives in UML. The as-is models represent the real problematic situations and the to-be models represent the future situations. The other aim is to improve the requirements adequacy by considering the misuse cases and a scenario analysis.

The last period takes four hours and is directed at developing an SRS.

At the end of each period, a presentation session is organized. In this session, the developer team explains the results of their activities to their customer team.

F. Workshop assignment

We have conducted the course 16 times over the past four years. It was normally run by three to six teams. Each team was composed of four to five engineers with more than ten years of software development experience. The following assignment was used in the course.

Workshop assignment::

Develop a software requirements specification for an effective meeting support system.

Though the assignment description is a bit vague, its topic is familiar to every engineer who is employed by a company and has a lot of meetings in their business lives. Both familiarity and vagueness are important for achieving the goals of the course. Familiarity helps students consider the problem as their real problem. Vagueness forces the students to define the problematic situation. For example, concerning the assignment, they have to consider the meanings of the effectiveness and ineffectiveness of their meetings and the support of meetings as well. The course should start with dynamic way of thinking for analyzing open subjects.

IV. Evaluation and Discussion

A. Evaluation method

We must evaluate the following points.

1) Students' satisfaction level of the course.
2) Adequacy of the role-playing workshop style of the course.
3) The hypothesis mentioned in section III-A.
   If an engineer has learned the importance of practicing RE, he/she can focus on the "why" of every requirement.

There are five levels of evaluation, including what Jack Phillips [33] defined as return on investment (ROI) methodology.

- L-1 Reaction and/or satisfaction of the planned action:
  Measurements of participant satisfaction with the training program.
- L-2 Learning:
  Measurements of participants' growth in knowledge, skills and attitudes during the training.
- L-3 Job applications and/or implementation:
  Measurements of participants' changes in their on-the-job behavior and progress using what they have learned from the training.
- L-4 Business impact:
  Measurements of the actual business results achieved as a consequence of applying the knowledge and skills gained from the training.

L-5 Return on investment (ROI):
Compares the program's monetary benefits to the actual cost of the training program.

We evaluated the course by L-1, L-2, and L-3. This paper does not report the evaluation of L-4 and L-5. They are our future work.

In the first trial in 2006, we developed training materials and evaluated the team size as well as the length of the course. The evaluation of the course design has started from the second trial year.

In general, a perfectly objective evaluation of any educational course is difficult. The following are obstacles to a perfect and objective evaluation.

- Students of the course are individually different every time.
- Every course is updated from the previous course in order to improve its effectiveness.

Though we have faced those obstacles in evaluating the instructional design of the course, we were able to overcome them because of the course specific environment.

- The students of the course were different each time, but the students did belong to the same company and were educated in the same way. Therefore, the students' technical ability can be regarded at the same level.
- The style, process and contents of each course followed the instructional design. Thus, we can regard the evaluation results of each course objectively comparable.

In the following sections, we report the results of the evaluation. L-1 evaluation was conducted by a simple questionnaire and by asking for the students' candid comments. For L-2 evaluation, we applied "learned memos (LM)", which are shared post-it notes that list what each student learned. The students made LMs after each presentation session and put them up on the wall. LMs tell us what the students learned and when they learned it. For the evaluation for L-3, we conducted reviews of sample SRSs.

B. Case 1: L-1 Evaluation

The questionnaire poses four questions for evaluating the course effectiveness for their job, its understandability, the teacher's ability and the material quality.

In the target course, most students were directed by their bosses to attend the course for the purpose of implementing the company's strategy of shifting their engineers' duty from the middle development phase to the early development phase. At the evaluation target class, there were twelve students in three groups reporting average scores of 3.70, 3.00, 3.80 and 3.60 range from 1 to 5, respectively. The scores were not so high. The scores might partially dependent on each student's business background. For example, there were some students who did not have any experience in modeling requirements. These students felt that modeling was difficult before understanding the importance of RE. In another example, some students believed that they would have fewer opportunities to come in contact with their clients. These engineers usually start their job based on the SRS provided by other engineers. According to the L-1 evaluation, it looks hard to motivate them to start learning RE. The course may be effective only for engineers who have had a chance to directly communicate with their clients and modeling knowledge and experiences. The acquisition of RE requires organizational cooperation in putting the engineers into the practical RE environment before attending the class.

The course was planned under the assumption that the students have already mastered UML. However, mastering UML is not equal to mastering modeling techniques. In order to improve the effectiveness of our course, we rescheduled the RE related courses in 2008. We scheduled the course after *Ethnomethodology* exercise and a modeling-technique-with-UML course. In the modeling course, the students discussed how to create appropriate models and, learned the real nature of the modeling. There are no correct model, but only adequate one. The *Ethnomethodological* exercises trained students how to analyze open-type objects dynamically. As a result, the average of the metrics became 4.20, 3.67, 4.07 and 3.87 respectively. Thus, we could improve the results for the L-1 evaluation. The RE education needs not only a course design, but also a design of students experience for learning RE.

One of the reasons why the understandability scores were low was revealed in the comment section of the questionnaire. They felt they needed more experience to fully understand the methods. The purpose of the course is to help the student understand the importance of RE and motivate them to start learning the RE methods. Therefore, this comment was what we expected from the students.

C. Case 2: L-2 Evaluation by Learned Memos

The learned memos (LMs) were used to evaluate the effectiveness of the role-playing style of this course. They were divided into two categories: the client role and the engineer role. The target course had 15 students. We collected 67 and 70 LMs, as comments about the client and engineer roles, respectively. Furthermore, there were 28 LMs for method applicability to the students' job and 6 LMs for the new knowledge of writing SRS exercise. In the LMs, the students wrote comments what they had learned from the exercise. Typical comments are given below.

- From the client role
  C-1 There were unproductive questions.
  When a student playing the client role was asked by another student playing an engineer role, "this is our understanding of your intention. Is it OK?" he could not reply "no", even though the model did not completely represent his situation. According to the memo, we could observe that the role-

playing exercise made him start to re-consider the better questions and the way of interviews.

C-2 Some of the requirements in the SRS did not have rationales.

When the requirements were decomposed into detailed functional requirements, their rationales became vague. Then, they had started to re-explore the rationale of the requirements from their exercise products in the final exercise. We hope that such an experience will help students better master RE in their jobs.

- From the engineer role

E-1 Requirements were not the product for an engineer, but for the clients.

One student also mentioned that she had to offer solutions that her clients could accept, rather than the ones she wanted to build. In the class, when the developer team presented their SRS to their client team, the client team asked, “so, how does the system work for our problems?” Analyzing and providing solutions ”for the clients” is one of the keys of RE. The student learned the key.

E-2 There were different problems and solutions for every client.

Another student mentioned that one solution could not solve everyone’s problems. The course instruction is designed according to the RODAN metamodel. The metamodel shows that every requirement depends on the client role. We guided students with the RE methods to the world that the metamodel implies.

The role-playing style is one of the popular educational styles [11]. The aim of role-playing is to provide students a chance to play other roles unfamiliar to them and to reconsider their responsibilities. From the comment C-1, we can understand how the student reconsiders his/her experience as a result of playing the client. To succeed in requirements engineering, every engineer should be able to focus on their clients’ situation and observe the real-world from their clients’ viewpoints. We can conclude that the role-playing style was the effective style of the course for leaning clients’ viewpoints.

D. Case 3: L-3 Evaluation by reviews of sample SRSs (1)

In order to simulate the application of the students’ skills that acquired in the course, we developed two sample SRSs for different software one for business software and the other for embedded software. The sample SRSs were organized according to the IEEE std.830 [34], but the samples lacked the rationale of each requirement.

The students executed the SRS review session before the course (pretest) and after the course (post test) for the different SRSs. When we compare and review the results of each student, the effect of the course can be clarified.

Recall our hypothesis.

- If an engineer has learned the importance of practicing RE, he/she can focus on the “why” of every requirement.

According to the result of the test, we can see how much the students have become engineers who can focus on the “why” aspect of requirements.

The actual duration of the course was approximately eighteen hours. There were nine students. They spent one hour for pretest and post test, respectively. Because of the limitation of the review schedule, they did not point out all the defects of the SRS. The results of the evaluation are shown in Table.I. In the pretest, the students could only point out the ambiguities and incompleteness of the sample SRS. In the post test, they discovered the insufficient descriptions of the rationale for each requirement.

We succeeded in teaching students the importance of the rationale of requirements. The results told us that the students became focusing on the rationale of each requirement after the course. Thus, our hypothesis cannot be rejected.

E. Case 4: L-3 Evaluation by reviews of sample SRSs (2)

We had a chance to execute the course for engineers belonging to a different organization. If our course is effective for all engineers, we could get similar results in the different organizations. The students had studied modeling techniques before our course. Therefore, the result of the case 4 can be comparable with the result of the case 3. The differences between the two are the course schedule and the evaluation duration. The course was held over four-week period, consisting of 3 hours per a week. Since the total time of the course was twelve hours, we could not spend the same time for developing the final SRS and for the post test in the class. We decided to give the students the exercise and the test as their homework. Their homework was due in one month.

According to the results of the pretest and post test, we could not find any differences. No students could point out the lack of rationale for any requirements in both of the tests. Thus, our hypothesis cannot be accepted in this case.

The differences between the two courses were its time concentration and the timing of the post test. The course density may influence the students in focusing on the requirements engineering way of thinking. The daily life of engineers is too concentrated on *how* to build the system. Our proposed course requires the students to devote their attention to the *why* aspect of the system to be developed. Therefore, if the course is sparse, it may be hard for the students to shift their focus from *how* to *why*.

For this reason, our course seems to have failed in bringing about the desired effect within a one month period, as witnessed after the end of the course. If the engineers had had a chance to adopt their experiences to their daily

Table I
RESULT OF EVALUATION FOR THE SRS REVIEWS IN A 2.5 DAYS COURSE (NUMBER OF FOUND DEFECTS OF SRS)

	number of found defects of SRS	Student IDs A	B	C	D	E	F	G	H	I
Pretest	Lack of purpose and rationale	0	0	0	0	0	0	0	0	0
	Ambiguities	0	1	1	0	0	0	0	2	0
	Incompleteness	0	1	4	0	0	1	0	0	0
Post test	Lack of purpose and rationale	5	4	3	2	3	2	5	5	3
	Ambiguities	1	0	1	0	1	0	0	0	0
	Incompleteness	0	0	2	0	1	1	0	0	2

lives, the results might have been different. The course need follow up training.

V. CONCLUSIONS

There are various RE educational courses. Most of them are designed for undergraduate students. Our target students are engineers with over ten years of experience. In the first year trial, we assumed that they had mastered modeling techniques. However, the second year trial told us that our assumption was not applicable to all students. We decided to include an additional week of UML modeling training period prior to our course from the third year on. Due to this addition, we were able to improve the course results. Our course relies heavily on the students' modeling ability.

Although the students' technical experiences were different, the assignment on meetings *were* familiar to every student. This familiarity helped students start the role-playing workshop quickly. Their LMs told us that the role-playing workshop led students to join easily and to have direct experience of being clients.

According to the L-3 evaluation by reviewing the sample SRSs, the students of case 3 could understand how critical it is to understand the importance of the rationale of requirements. After the evaluation of case 4, we could not conclude the result can be gained every time. The evaluation of engineers working in different working environments is our future work. Real L-3 evaluation is required for our course, rather than the simulation. To evaluate our course according to Phillips's L-3, along with other evaluation levels, is also our future work. This will require more cooperation from both students and organizers.

The course introduces several methods, but these methods are not mandatory for all engineers. The metamodel of the products of the course can accept other methods if their products fit the metamodel and have a proper position on the RE map. The students are responsible for selecting adequate methods and/or applying their company's methods by mapping to the RE map. The comments on the questionnaire told us that the course could motivate students to learn RE methods.

Lessons learned in our experience are as follows.

- Methods in the lower-right of the RE map can open the engineers' eyes to the *why* aspect of requirements.
- The focus on the rationale and/or background of requirements improve SRSs quality.

Problems including follow up training still remain to be solved.

ACKNOWLEDGMENTS

We would like to thank all the students that took the course and also offer special thanks to the course staff for helping to improve our course.

REFERENCES

[1] B. W. Boehm, *Software Engineering Economics*. Prentice Hall, 1981.

[2] T. Nakatani, "Improving the engineering mind in eliciting requirements," in *The 2nd International Workshop on Requirements Engineering Education and Training (REET'07)*, 2007, pp. 37–41.

[3] T. Tsumaki and T. Tamai, "A framework for matching requirements elicitation techniques to project characteristics," *Software Process: Improvement and Practice*, vol. 11, no. 5, pp. 505–519, 2006.

[4] T. Nakatani, T. Tsumaki, and T. Tamai, "Instructional design of a requirements engineering education course for professional engineers," *Multimedia Services in Intelligent Environments-Integrated Systems*, 2010.

[5] D. Gause, "Making the case for the soft in software engineering or what does it really take to be prepared for the RE world ahead?" 2005. [Online]. Available: http://lams.epfl.ch/course/esoa2008/old_esoa2007/business-it/esoa2007_week6b_gause.pdf

[6] U. Nikula, "Experiences of embedding training in a basic requirements engineering method," in *The 17th Conference on Software Engineering Education & Training (CSEET'04)*. IEEE, 2004, pp. 104–109.

[7] G. Lami, "Teaching requirements engineering in the small: an under-graduate course experience," in *the 1st International Workshop on Requirements Engineering Education and Training (REET'05)*, 2005. [Online]. Available: http://re.cti.depaul.edu/REET05/P4_Lami.pdf

[8] B. Berenbach and T. Wolf, "A unified requirements model; integrating features, use cases, requirements, requirements analysis and hazard analysis," in *International Conference on Global Software Engineering (ICGSE 2007)*. IEEE, 2007, pp. 197–203.

[9] R. Smith and O. Gotel, "Using a game to introduce lightweight requirements engineering," in *The 15th International Requirements Engineering Conference (RE'07)*. IEEE, 2007, pp. 379–380.

[10] J. Beatty and M. Alexander, "Games-based requirements engineering training: an initial experience report," in *The 16th International Requirements Engineering Conference (RE'08)*. IEEE, 2008, pp. 211–216.

[11] D. Zowghi and S. Paryani, "Teaching requirements engineering through role playing: Lessons learnt," in *The 11th International Requirements Engineering Conference (RE'03)*. IEEE, 2003, pp. 233–241.

[12] G. Regev, D. C. Gause, and A. Wegmann, "Requirements engineering education in the 21st century, an experiential learning approach," in *The 16th International Requirements Engineering Conference (RE'08)*. IEEE, 2008, pp. 85–94.

[13] N. H. Madhavji and J. Miller, "Investigation-based requirements engineering education," in *the 1st International Workshop on Requirements Engineering Education and Training (REET'05)*, 2005. [Online]. Available: http://re.cti.depaul.edu/REET05/P3_Madjhavi.pdf

[14] B. Berenbach and T. Rayment, "The evaluation of a requirements engineering training program at Siemens," in *The 16th International Requirements Engineering Conference (RE'08)*, 2008, pp. 205–210.

[15] E. Simmons, "Reflections on a successful corporate requirements engineering training curriculum," in *the 1st International Workshop on Requirements Engineering Education and Training (REET'05)*, 2005. [Online]. Available: http://re.cti.depaul.edu/REET05/F3_Simmons.pdf

[16] A. Wegmann, "Theory and practice behind the course "designing enterprise-wide IT systems"," *IEEE Transactions on Education*, vol. 47, no. 4, pp. 490–496, 2004.

[17] T. Nakatani and T. Fujino, "Role and owner based business domain analysis," in *Asia Pacific Software Engineering Conference 2005 (APSEC2005)*. IEEE, 2005, pp. 130–137.

[18] T. Nakatani and T. Fujino, "Rodan: A requirements elicitation method for a business domain by focusing on roles," *IPSJ Journal*, vol. 48, no. 8, pp. 2534–2550, 2007, (in Japanese).

[19] S. Brinkkemper, M. Saeki, and F. Harmsen, "Meta-modelling based assembly techniques for situational method engineering," *Information Systems*, vol. 23, no. 7, pp. 489–508, 1998.

[20] M. Haglind, L. Johansson, and M. Rantzer, "Experiences integrating requirements engineering and business analysis," in *The IEEE Third International Conference on Requirements Engineering*. IEEE, 1998, pp. 108–117.

[21] T. DeMarco, *Structured Analysis and System Specification*. Prentice Hall, 1979.

[22] M. Jackson, "Seeing more of the world," *IEEE Software*, vol. 21, no. 6, pp. 83–85, 2004.

[23] J. M. Carroll, *Making Use: Scenario-Based Design of Human-Computer Interactions*. MIT Press, 2000.

[24] I. Alexander, "Misuse cases: Use cases with hostile intent," *IEEE Software*, vol. 20, no. 1, pp. 58–66, 2003.

[25] L. Beus-Dukic and C. Myers, "Use and abuse cases," in *the 1st International Workshop on Requirements Engineering Education and Training (REET'05)*, 2005. [Online]. Available: http://re.cti.depaul.edu/REET05/F4_BeusDeukic.pdf

[26] P. Checkland and J. Scholes, *Soft Systems Methodology in Action*. John Wiley & Sons, 1991.

[27] H. Garfinkel and A. Rawls, *Ethnomethodology's Program*. Rowman & Littlefield Publishers, 2002.

[28] E. S. K. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," in *The 3rd International Symposium on Requirements Engineering (RE'97)*. IEEE, 1997, pp. 226–235.

[29] CEDITI, "A KAOS tutorial," 2007. [Online]. Available: http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf

[30] B. Boehm, P. Bose, E. Horowitz, and M. J. Lee, "Software requirements negotiation and renegotiation aides: A Theory-W based spiral approach," in *The 17th International Conference on Software Engineering (ICSE'95)*. ACM, 1995, pp. 243–253.

[31] Object Management Group, "Unified modeling language." [Online]. Available: http://www.uml.org/

[32] H.-E. Eriksson and M. Penker, *Business Modeling with UML*. John Wiley & Sons, 2000.

[33] J. J. Phillips and R. D. Stone, *How to Measure Training Results – a Practical Guide to Tracking the Six Key Indicators*. McGraw-Hill, 2002.

[34] IEEE Std. 830-1998, *Recommended Practice for Software Requirements Specifications*. IEEE, 1998.