

# Requirements Engineering for Time-to-Market Projects

*Christopher McPhee*  
*Quadrus Development Inc.*  
*Suite 200*  
*3553 – 31 Street NW*  
*Calgary, AB, Canada*  
*+1 403 257-0850*  
*chrism@quadrus.com*

*Armin Eberlein*  
*Department of Elect. & Comp. Engineering*  
*University of Calgary*  
*2500 University Drive NW*  
*Calgary, AB, Canada*  
*+1 403 220-5002*  
*eberlein@enel.ucalgary.ca*

## Abstract

*This paper identifies the essential components of a Requirements Engineering (RE) process and provides alternatives to the Requirements Analyst (RA) when deciding on a particular set of RE methods. Existing methods of choosing RE techniques are discussed, and a new technique is proposed specifically for choosing RE techniques for projects with critical Time-to-Market (TTM) projects.*

*The components of the essential RE process as well as many other insights into the practice of RE were identified by examining the current literature and by analyzing the data from 25 completed surveys concerning the area of RE. The survey data provides insight into the state of the practice in Requirements Engineering.*

## 1. Introduction

For a company to develop a software product efficiently and successfully, it is important for the RE phase to be well-defined [1]. Unlike other industries where the majority of capital is spent on tangible materials, the tangible materials in a software product are a very small part of the overall cost of the project. By far, the costliest part of a software product is the development of the application. Reproduction costs are measurable and are normally insignificant [2]. Increased costs resulting from changes to requirements, however, are often much more difficult to measure quantitatively. As a result, software requirements are often quite volatile [3]. When requirements change, effort is required to modify the product to meet the new requirements, thus increasing the cost and delaying the schedule of the project. If requirements remain volatile for too long, the likelihood of project success decreases dramatically, as modifying software to incorporate new/modified requirements is often complicated and error-prone [4].

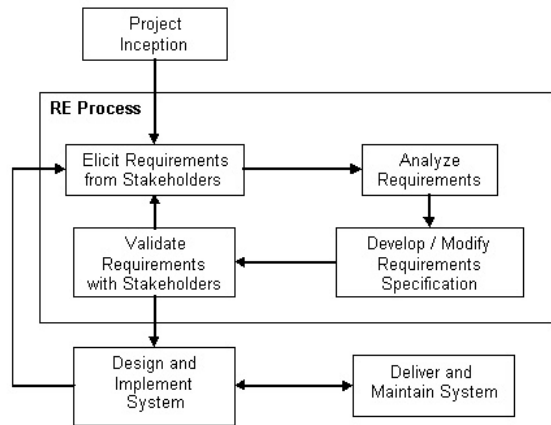
Unfortunately, RE is not particularly easy to do well. Even when requirements experts are employed in the RE phase to help determine the set of requirements, they may miss up to half of the necessary end-user requirements

[5]. To make matters even more difficult, the application of large-scale organizational methodologies to the RE process of TTM projects has been shown to be largely ineffective [6]. Although it is difficult to do well, a sound RE process is important if a company wishes to achieve repeatable project success [7].

The purpose of this research is to provide guidance to software developers in determining the appropriate steps to take in the RE process for a TTM project and the appropriate techniques to facilitate a successful RE project phase. Nikula et al have shown there to be a need for education and knowledge transfer from academia to industry in terms of RE processes and best practices [8]. In addition to discussing the fundamentals of RE for use in a TTM project, this research also describes an RE technique evaluation process which can assist software developers in determining the most appropriate set of RE techniques for use on a TTM project.

## 2. The RE Process

Sommerville & Kotonya [10] state that the RE process consists of 4 main areas, as shown outlined in Figure 1. Although the 4 areas are distinct from one another in their process and purpose, they may occur simultaneously in the requirements stage of a project. For TTM projects, it is desirable that these 4 tasks occur at the same time or at least in very small increments. By being able to perform the tasks together, the requirements engineer saves overall schedule time by requiring less overhead to set-up meetings, contact individuals to clarify issues, etc [9].



**Figure 1 The RE Process**

It is not necessary for project teams to rush through the RE process for TTM projects. The time spent up-front may have significant time savings during the latter stages of the project. In a survey of 16 Finnish companies, one third reported that even with the ever-increasing need to reduce schedule time, they should be spending more time and effort on RE [8].

The RE process consists of four main activities: elicitation, analysis, specification, and validation [10]. Elicitation is the activity of gathering the requirements from stakeholders. After gathering the requirements, they are analyzed to determine areas requiring clarifications, logical groupings, etc. After being analyzed, the requirements are documented and validated with the stakeholder to ensure that the product developed from the requirements will meet the needs of the stakeholder.

The majority of the RE activities should be completed before substantial effort is expended on detailed design and implementation. One artifact of the RE phase is a documented requirements specification that has been validated by the affected stakeholders. The validated requirements specification might consist of a formal requirements document [1, 3], an experimental prototype [11], or a combination of the two. There will likely be stakeholder conflict encountered while developing and validating a requirements specification, but it is much better to start dealing with, and resolving, conflict in the early stages of a project rather than waiting until the end with the hopes that everything will turn out alright [12]. A solid RE process requires plans for addressing and resolving conflict [3].

### 3. Rapid Application Development

Rapid Application Development (RAD) is a broad term used in a number of different contexts. The term itself means different things to different people [9]. In the context of this research, Rapid Application Development

is any development methodology or activity whose overall impetus is to increase speed of application development over that of the traditional development life cycles.

Increased speed of application development can be achieved in three fundamental ways: [9]

1. Perform fewer tasks.
2. Perform tasks more quickly.
3. Perform tasks concurrently.

Points 1 and 2 can lead to a decreased amount of development effort, if development effort is defined in terms of person-hours or person-months. Point 3, however, does not achieve decreased development effort since the overall amount of development effort actually increases as a result of the increased coordination between actors. Point 3 also leads to increased coordination complexity and increases risk in that actors performing concurrent tasks must work from incomplete artifacts and/or must coordinate with other actors performing concurrent activities [9].

### 4. RE Techniques

There are several techniques available to the RA in discovering and analyzing requirements. During the course of this research, 25 techniques were evaluated in terms of their applicability for use in TTM projects. The purpose behind this analysis was to determine if particular RE techniques allow for reduction of time required either for requirements engineering, or for the overall project. It is possible to do very little requirements engineering, thus reducing the time required for requirements engineering to essentially nil. This, however, puts the project at great risk as time may be wasted reworking the product as requirements are discovered [13]. The analysis of the RE techniques assumes that the technique, either by itself or in combination with other techniques, will be used to construct a reasonably complete requirements specification within the initial requirements engineering phase of the project.

Each technique was rated against the three methods of reducing schedule time as discussed in section 3. The methods were rated using the following scale:

1. The technique makes no allowance for the criterion.
2. The technique must be tailored to make allowance for the criterion.
3. The technique explicitly makes allowance for the criterion.

Using this scale, the minimum possible score is 3 (if the technique receives a rating of 1 for each of the 3 criteria) and the maximum possible score is 9 (if the technique receives a rating of 3 for each criterion). The following table shows the score for each technique examined:

**Table 1 RE Technique Scores**

Technique	Score
Joint Application Development [14]	8
Informal Modeling [12]	7
Semi-formal Modeling [15, 16]	7
Scenarios / Use Cases [16,17]	7
Evolutionary Prototyping [10]	7
Requirements Testing [18]	7
Requirements Reuse [19]	7
Requirements Checklists [12]	6
Requirements Reviews [20]	6
Requirements Prioritization [21]	6
Formal Modeling [22]	6
Throw-away Prototyping [4, 10]	6
Interviews [12]	6
Quality Function Deployment [23]	6
Cooperative Requirements Capture [12]	6
Designer as Apprentice [24]	5
Observation and Social Analysis [12]	5
Focus Groups [25]	5
Future Workshops [12]	5
User-Centered Design [10, 12]	5
Viewpoint-Oriented Techniques [10]	5
Requirements Tracing [18]	5
Requirements Change Management [26]	5
Soft Systems Methodology [27]	4
Data Mining [28]	4

## 5. Empirical Research

An on-line survey was constructed to gather information from individuals involved in software development ([http://sern.ucalgary.ca/~cmcphee/RE/RE\\_Questionnaire.html](http://sern.ucalgary.ca/~cmcphee/RE/RE_Questionnaire.html)). The survey covered several different aspects of RE, with the goal of obtaining greater insight into how software developers view the RE phase and what tools or techniques they see as being most

useful. Where appropriate, survey respondents were asked to compare TTM and non-TTM projects to determine where the two types of projects were different in terms of approach to RE.

The following sections discuss the areas of information presented in the survey and the rationale supporting the inclusion of questions in those areas.

### 5.1. Work Experience

The survey was not targeted at any particular group of software developers. Rather, it was anticipated that responses would be gathered from individuals with diverse backgrounds. This section asked questions regarding experience both in terms of years of development experience and regarding project types and industry domains.

The purpose of collecting this data was to determine if there is any correlation between an individual's experience and his views on RE in software development.

The following information was determined from responses in this section:

- Approximately half of the respondents had 10 years of software development experience or more.
- Approximately half of the respondents have spent the majority of their careers working on TTM projects.
- In total, experience level in each of the 5 phases of a software development project lifecycle was almost the same, with Maintenance being the least and Implementation being the most.

### 5.2. Project Success Factors

Project success factors are those factors which have a major influence on the overall success of a project. Specifically, three questions that were asked that have been identified as being important project success indicators [29]:

- How do you know when a project is done?
- How do you know if a project was a success?
- What is the relative importance of budget, schedule, and scope/quality?

The answers to these questions provide insight into how a software developer sees a successful project coming together and what they believe to be the most important factors in software development projects.

The following are the general trends in how respondents determined that a project was done:

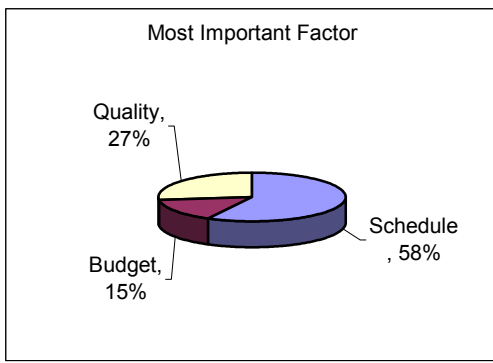
1. User or QA signoff and/or customer acceptance.
2. Client takes ownership.
3. All requirements have been satisfied.

And these are the general trends in how respondents determined that a project was successful:

1. Customer is happy/satisfied (expectations have been met).
2. Estimates have not been grossly exceeded.
3. All requirements have been met.

The top responses both in terms of project completion and project success involved customer sign-off and satisfaction with the final result. This suggests that having testable requirements is absolutely essential. By creating test cases, we are assuring ourselves that we can accurately determine when a project is done.

When asked about the relative priority of Budget, Schedule, and Scope/Quality, the respondents answered as follows:



**Figure 2 Project Priorities**

### 5.3. Attributes of ‘Good’ Requirements

This section listed the seven attributes of good requirements as defined by the IEEE [30]:

- Unambiguous
- Complete
- Verifiable
- Consistent
- Modifiable
- Traceable
- Usable

On a TTM project, it may not be possible, or even desirable, to spend time and effort to ensure that each of the attributes is fulfilled. For instance, one attribute of good requirements is identified as being unambiguous [30]. Truex [31] argues that we should not strive for complete unambiguity, because the environment is always changing and there must be flexibility built into the system requirements.

The responses to questions in this section provide guidance in helping software developers choose suitable RE techniques. If some techniques are appropriate for achieving a particular attribute of a requirement that is deemed to be important for a TTM project, then, all other

things being equal, that technique should be used over another technique that does not contribute to achieving the same attribute.

There was little difference between TTM and Non-TTM projects in terms of what individuals felt to be the importance of each attribute. Attributes that were felt to be slightly more important in TTM projects were ‘Unambiguous’ and ‘Usable’, while ‘Traceable’ was felt to be of greater importance for Non-TTM projects. These differences suggest that the respondents are comfortable with more ambiguity in the requirements for Non-TTM projects than they are on TTM projects. That said, ‘Unambiguous’ was still deemed the most important attribute for TTM projects, and the second most important attribute for non-TTM projects.

### 5.4. RE Techniques

This section provided a list of RE tools and techniques for which the respondents were to indicate both their knowledge level and the perceived usefulness. If a respondent did not have any knowledge of a tool or technique, they were not to specify its perceived usefulness. Respondents were also asked for the perceived usefulness of each tool or technique on TTM versus non-TTM projects.

This section was included in the survey for three reasons:

1. To determine general knowledge of available RE tools and techniques.
2. To determine overall usefulness of technique.
3. To determine whether there was a perceived difference in usefulness of tools and techniques for TTM versus non-TTM projects.

Respondents were asked to rate their familiarity and the perceived usefulness (both for TTM and for non-TTM projects) of several RE techniques on a scale of 1 to 5, with 1 being low familiarity/usefulness and 5 being high familiarity/usefulness. For the usefulness of techniques, only respondents who indicated that they had at least some familiarity were asked as to the effectiveness of the technique.

**Table 2 Top 10 RE Techniques**

Rank	Familiarity	Usefulness (TTM)	Usefulness (Non-TTM)
1	Scenarios / Use Cases	Prioritization	Change Management
2	Semi-Formal Modeling	Change Management	Semi-Formal Modeling
3	Informal Modeling	Scenarios / Use Cases	Reviews

Rank	Familiarity	Usefulness (TTM)	Usefulness (Non-TTM)
4	Change Management	Semi-Formal Modeling	Scenarios / Use Cases
5	Evolutionary Prototyping	Testing	Checklists
6	Interviews	Evolutionary Prototyping	Testing
7	Prioritization	Reuse	Traceability
8	Reviews	Interviews	Viewpoint-Oriented Techniques
9	Throw-away Prototyping	Reviews	Interviews
10	Checklists	Checklists	Designer-as-Apprentice

Although the rankings were not identical, many of the same techniques are included in the top ten for all three categories. Most noticeably, Scenarios / Use Cases, Semi-Formal Modeling, and Change Management are all ranked in the top five. Not surprisingly, Prioritization was thought to be the most useful technique for TTM projects, but did not even register on the top ten techniques for non-TTM projects. Techniques that normally take more effort in terms of the Requirements Analyst's time such as Traceability, Viewpoint-Oriented Techniques, and Designer-as-Apprentice were thought to be useful for non-TTM projects, but not as useful for TTM projects.

The rank correlation coefficient was calculated for three combinations of the RE technique data. A rank correlation coefficient was calculated rather than an absolute correlation coefficient because it is the relative usefulness of a technique that is of interest.

The following table shows the results of this calculation:

**Table 3 Rank Correlation Coefficients for RE Techniques**

Comparison	Rank Correlation Coefficient
Familiarity of Technique versus Usefulness (TTM)	0.729
Familiarity of Technique versus Usefulness (Non-TTM)	0.876
Usefulness (TTM) versus Usefulness Non-TTM	0.679

All three rank correlation coefficients were positive and were greater than the minimum correlation coefficient required for a 99.9% confidence limit. The results lead to the following observations:

1. Technique familiarity is positively correlated with perceived usefulness of a technique, both for TTM and Non-TTM projects.
2. Perceived usefulness of RE techniques for TTM projects is positively correlated with perceived usefulness of techniques for Non-TTM projects.

An additional rank correlation coefficient was calculated to determine the correlation between perceived usefulness of RE techniques for TTM projects versus the appropriateness of RE techniques for TTM projects as calculated by the process described in section 4. The rank correlation coefficient for these two datasets is 0.366 which is less than the minimum rank correlation coefficient for 95% confidence limits for a sample of this size, meaning that the correlation is not statistically significant within 95% confidence limits.

This result leads to two possible conclusions:

1. The technique for determining appropriateness of RE techniques for TTM projects as described in section 4 is not valid; or
2. Since perceived usefulness of RE techniques for TTM projects is highly correlated with technique familiarity, software developers incorrectly perceive some techniques to be more appropriate than others.

Respondents who reported that Schedule was normally their highest priority had, on average, less familiarity with the RE tools and techniques referenced in the survey. This suggests that individuals who mainly work on TTM projects have less general knowledge about the RE tools and techniques available to them. Intuitively, this makes sense as those individuals who are not time-constrained are likely to be more willing and able to research the RE tools and techniques available to them.

### 5.5. Personal Preferences regarding RE

This section asked questions concerning personal preferences regarding the RE phase of a project. Questions related to whether or not an individual liked performing RE and the attributes of good Requirements Analysts.

The answers to these questions can be used to further refine the usefulness of RE tools and techniques. If, for example, a number of individuals reported that one downside of RE is the amount of documentation that must be created and maintained, RE techniques which incorporate the construction of documentation as a by-product of the technique would be considered more desirable than another technique which treats the construction of documentation as a separate activity.

Respondents who indicated that they had explicitly chosen a RE technique in the past were then asked to identify how the technique was chosen. The percentages

indicated in the following tables do not add up to 100% because most respondents who answered this question indicated several reasons, and some respondents did not answer all questions.

**Table 4 Reasons for Choosing RE Techniques**

Reason for Choice of Technique	Percentage of Respondents who Identified the Reason
Previous Experience with Technique	28%
Facilitates Good Communication	24%
Company Standard	16%
Tool Support Available	8%

When asked about the attributes of a good RE technique, all respondents, regardless of whether or not they had chosen a technique before, were asked to respond. The following table shows those responses:

**Table 5 Desirable Qualities of RE Techniques**

Desirable Quality of RE Technique	Percentage of Respondents who Identified the Quality
Easy to Use	32%
Facilitates Good Communication	28%
Facilitates capture of complete set of requirements	24%
Allows for or incorporates traceability	20%
Improves the product quality	8%
Tool support	8%
Incorporates prioritization	4%

Respondents were also asked about what they liked and disliked about RE. The following tables show the responses:

**Table 6 Attractive Aspects of RE**

Attributes of RE that are most liked	Percentage of Respondents who Identified the Attribute
Defining common goals / scope	60%
Gaining domain knowledge	28%
Exploring possibilities	24%
Team building	8%

**Table 7 Unattractive Aspects of RE**

Attributes of RE that are most disliked	Percentage of Respondents who Identified the Attribute
Difficulties in communicating with stakeholders	48%
Documentation	28%
Changing Requirements / Feature Creep	24%

In almost all of the areas regarding personal preferences in RE, difficulty of communication and/or the importance of communication was high-lighted. This is not unexpected, but it underscores the importance of a good line of communication between the Requirements Analyst and the customer. RE techniques, whether they be applied on a TTM or non-TTM project must enhance the analyst/customer communication in order to be successful.

Another important factor in the usefulness of a technique was the respondents' perceived effectiveness and/or the ease of use of the technique. This is especially important for TTM projects where the Requirements Analyst might not have ample time to become familiar with a new technique. Unless the technique is easy to use, the analyst is more likely to go back to a familiar technique.

### 5.6. Importance of RE

This section asked respondents how much effort should be spent in the RE phase of a project in terms of percentage of overall project effort. Respondents were asked to determine percentages for both TTM and non-TTM projects.

Although not contributing directly to assisting a RA in determining an appropriate RE process, the responses could be used to provide software developers with guidance in terms of the amount of effort that should be spent in the RE phase of a project.

Respondents indicated that whether or not they felt that their company spends enough time on RE, effort allocated to RE should be approximately 25% of the total project effort. One slight difference was found when dealing with non-TTM projects. For non-TTM projects, respondents who felt that their company spends enough time in RE actually identified higher RE effort allocation than those who felt their company did not spend enough effort in RE. This suggests that when an individual's company does not spend enough time in RE, he underestimates the appropriate amount RE effort required.

The majority of respondents (two-thirds) did not believe that enough time is spent on RE. In general, respondents recognized that there is a lack of effort spent on hashing out requirements details. Although this is recognized, it is difficult to determine how RE effort should be spent. Obviously, in the waterfall process model, one could specify that the entire RE should be performed (use up all allocated time for RE) before starting on the design and implementation. However, the requirements often need to be amended throughout the design and implementation stages. Therefore, upfront RE effort is best spent on determining the core, unalterable set of requirements with time set-aside at latter points in the project to revisit the requirements and to maintain traceability of requirements to design, implementation and test.

## 6. Summary and Future Research

The RE process can be seen as consisting of four phases that must be addressed iteratively if a requirements specification of reasonable quality is to be produced. As identified by Sommerville [10], those phases are Elicitation, Analysis, Specification, and Validation. By iterating through these phases, the Requirements Analyst gradually discovers the requirements of the system and works to refine the complete set of necessary requirements. Explicitly addressing the separate phases of the RE process, however, is not sufficient to adequately determine the system requirements.

TTM projects are those for which schedule is the driving factor. Although many RE tools and techniques exist, many were developed with the assumption that a stable set of requirements exist, and the Requirements Analyst must simply determine and document them [1]. With the changing nature of today's technology, it may be impossible to determine the complete set of requirements of an organization [31]. The profitability of TTM projects depends on the ability to quickly determine and satisfy the needs of a particular market niche.

A survey was developed and administered to members of the software development community, and the following conclusions were reached following an analysis of the data provided by the respondents:

- There exists a general lack of knowledge regarding available RE techniques.
- RE Technique usefulness is largely determined by familiarity.
- Individuals who worked primarily on schedule-driven projects generally have less familiarity with RE techniques than do individuals who work primarily on budget or scope/quality-driven projects.
- Good communication is a key factor in successful RE.
- It is important that requirements be unambiguous.
- The perceived usefulness of RE techniques for TTM projects is correlated with perceived usefulness for Non-TTM projects but is not correlated with theoretical usefulness of RE techniques on TTM projects.
- RE techniques must be easy to use to be considered by RAs on TTM projects.
- Approximately 25% project time should be spent in RE, although majority of respondents do not feel enough time is allocated to RE in their companies.

There are a number of areas where the research presented in this thesis paper could be expanded:

- Develop a practical guide to Requirements Engineering for software developers to use on time-to-market projects.
- Evaluate appropriate RE techniques for use when one of the other project drivers is most important (Budget or Scope/Quality).
- Evaluate lifecycle methodologies appropriate for TTM projects rather than focusing solely on RE techniques.
- Perform controlled experiments to evaluate the actual effectiveness of the RE techniques that were found to be theoretically suitable for TTM projects.

## 7. References

- [1] R.J. Costello and D. Lui, "Metrics for Requirements Engineering", *J. Systems Software*, 1995, 29:39-63.
- [2] C. Shapiro and H.R. Varian, *Information Rules: a Strategic Guide to Network Economy*, Harvard Business School Press. Wiley. 1998.
- [3] I. Sommerville, P. Sawyer and A. Sommerville, *Requirements Engineering: A Good Practices Guide*, John Wiley & Sons Ltd. 1997.
- [4] F.P. Brooks, *The Mythical Man Month*, Addison-Wesley Longman, Inc, 1995.
- [5] D.L. Cuomo, and C.D. Bowen, "Understanding the usability issues addressed by three user-system interface evaluation techniques", *Interacting with Computers*, Vol. 6, No. 1, pp.86-108. 1994.
- [6] N. Vitalari and G. Dickenson, "Problem Solving for Effective System Analysis: An Experimental Exploration", *Communications of the ACM*, Vol. 26, No. 11, November, 1983 pp. 945-956.
- [7] M. Paulk et al, "Key Practices of the Capability Maturity Model, Version 1.1", *CMU/SEI Technical Report*. 1993.
- [8] U. Nikula, J. Sajaniemi and H. Kalviainen, "A State-of-the-Practice Survey on Requirements Engineering in Small- and

- Medium-Sized Enterprises.” *Telecom Business Research Center Lappeenranta Technical Report*, Lappeenranta, Finland. 2000.
- [9] D.N. Card, “The RAD Fad: Is Timing Really Everything?”, *IEEE Software*, September 1995, pp.20-22. 1995.
- [10] I. Sommerville and G. Kotonya, *Requirements Engineering: Processes and Techniques*. John Wiley & Son Ltd. 1998.
- [11] S. Gasson, “The Reality of User-Centered Design”, *Journal of End User Computing*, Issue 4, 1999, pp. 3-13.
- [12] L.A. Macaulay, *Requirements Engineering*. Springer-Verlag. 1996.
- [13] G. Stark, A. Skillicorn., R. Ameen., “An Examination of the Effects of Requirements Changes on Software Releases”, *Crosstalk: The Journal of Defense Software Engineering*. December. 1998.
- [14] J.H. August, *Joint Application Design: The Group Session Approach to System Design*. Englewood Cliffs, NJ: Yourdon Press. 1991.
- [15] E. Yourdon, *Modern Structured Analysis*, Yourdon Press. 2000.
- [16] G. Booch, I. Jacobson, J. Rumbaugh, *The Unified Modeling Language User Guide*, Addison-Wesley. 1998.
- [21] K.E. Wiegers, “First Things First: Prioritizing Requirements”, *Software Development Magazine*, September. 1999.
- [22] J.A. Goguen, “Formal Methods: Promises and Problems”, *IEEE Software*, Jan/Feb, pp.73-83. 1997.
- [23] Y. (ed) Akao, *Quality Function Deployment. Productivity Press*, Cambridge M.A. 1990.
- [24] H.R. Beyer, K. Holtzblatt, “Apprenticing with the Customer”, *Communications of the ACM* 38(5), pp.45-53. 1995.
- [25] J.F. Templeton, *The Focus Group : A Strategic Guide to Organizing, Conducting and Analyzing the Focus Group Interview*, Probus Publishing Co. 1994.
- [26] L. Lavazza, “Enhancing Requirements and Change Management through Process Modeling and Measurement”, *Proceedings of the 4th International Conference on Requirements Engineering*. 2000.
- [27] P.B. Checkland, J. Scholes, *Soft Systems Methodology in Action*. Chichester: John Wiley and Sons. 1990.
- [17] M. Jarke, X.T. Bui, J.M. Carroll, “Scenario Management: An Interdisciplinary Approach”, *Requirements Engineering Journal*, 1999.
- [18] L.H. Rosenberg, T.F. Hammer, L.L. Huffman, “Requirements, Metrics, and Testing”, *15th Annual Pacific Northwest Software Quality Conference*. 1998.
- [19] W. Lam, J.A. McDermid, A.J. Vickers, “Ten Steps Towards Systematic Requirements Reuse”, *Requirements Engineering Journal*, Volume 2, Number 2, pp.102-113. 1997.
- [20] W. Lam, J.A. McDermid, A.J. Vickers, “Ten Steps Towards Systematic Requirements Reuse”, *Requirements Engineering Journal*, Volume 2, Number 2, pp.102-113. 1997.
- [28] D. Morris, B. Tamm, *Concise Encyclopedia of Software Engineering*, Pergamon Press, 1993.
- [29] F. Hartman, *Don't Park Your Brain at the Door : A Practical Guide to Improving Shareholder Value with Smart Management*. Project Management Institute Publishing, 2000.
- [30] *IEEE Guide to Software Requirements Specifications, IEEE Std 830-1984*. IEEE Inc., 345 East 47th St, New York, NY 10017, USA. 1984.
- [31] D.P. Truex, R. Baskerville and H. Klein, “Growing Systems in Emergent Organizations”, *Communications of the ACM*, Vol. 42, No. 8, August 1999, pp. 117-123.