Requirements Maturation Analysis by Accessibility and Stability

Takako Nakatani*, Toshihiko Tsumaki†, Michio Tsuda‡, Mari Inoki§, Shozo Hori¶,and Keiichi Katamine‖
* *The University of Tsukuba, Tokyo, Japan. Email: nakatani@gssm.otsuka.tsukuba.ac.jp*
†*National Institute of Informatics, Tokyo, Japan. Email: tatsumaki@nii.ac.jp*
‡*Osaka University, Osaka, Japan. Email: m7199801@yahoo.co.jp*
§*Toshiba Solutions Corp. Tokyo, Japan. Email: Inoki.Mari@toshiba-sol.co.jp*
¶*Yaskawa Information Systems Corp., Japan. Email: hori@ysknet.co.jp*
‖*Kyushu Institute of Technology, Fukuoka, Japan. Email: katamine@ci.kyutech.ac.jp*

***Abstract*—The success of any projects can be affected by requirements changes. We define requirements elicitation as the activity of adding, deleting, and modifying requirements. We here refer to the completion of requirements elicitation of a software component as requirements maturation. The requirements of the component will never be changed after 100% maturation. We introduce the requirements maturation efficiency (RME) in order to estimate the requirements maturation period for the component. The RME of the requirements represents how quickly the requirements of a component reach 100% maturation. The goal of this paper is to define a method of estimating the RME of each component. Since requirements analysts (RAs) elicit requirements by accessing requirement sources and stability is one of the characteristics of requirements, we can assume that the RME of a component must be derived from the accessibility of the requirements source and the stability of the requirements. We model accessibility as the number of information flows from the source of the requirements to the RA and model stability with the requirements maturation index (RMI). The results from multiple regression analysis for two cases indicate that RME can be derived by these two factors with a significant level of 5%. We also discuss a method by which to estimate the period for requirements maturation.**

***Keywords*-requirements accessibility; requirements stability; requirements maturation**

I. Introduction

Incomplete, inaccurate, or vague requirements are potential risks in any projects [1], [2]. Requirements are changed in order to correct problems, improve requesters' satisfaction, and adapt the system to the requesters' future environments. There are other factors that cause the requirements changes. IEEE Std. 830-1998 [3] outlines the recommended quality in software requirements specifications (SRSs). Since most SRSs are written in natural language [4], which can result in ambiguous requirements, the requirements are often changed to resolve these ambiguities. Furthermore, new requirements are sometimes discovered by refining the original requirements.

In distributed software development, the geographic distance between engineers working on a project affects the quality of the software [5]. The communication distance between clients and developers may similarly affect the maturation of the requirements. We here refer to the completion of requirements elicitation of a software component as requirements maturation. The requirements of the component will never be changed after maturation. For example, if the clients are too busy to respond to e-mails from the requirements analyst (RA), it may be difficult for their requirements to mature within the expected period. However, if the RA can maintain good communication with his/her clients, he/she can elicit precise and unambiguous requirements in the expected schedule and specify these requirements on time.

Our motivation is to provide a method by which an RA could estimate the period for requirements maturation. The ratio of requirements maturation of every component reaches 100% by the end of the project. Requirements maturation efficiency (RME) is a measure of requirements maturation. It represents how quickly the requirements reach 100% maturation. If all of the requirements have been elicited at the final day of the project, the RME of the requirements must be zero. The project's duration, L, is measured in days or months from its initiation to completion. The maturation period, l, is also measured in days or months from the initiation of the project to requirements maturation. RME is defined by the following equation:

$$RME = 1 - \frac{l}{L} \tag{1}$$

If we can predict the RME of the requirements, we can then estimate the maturation period for the requirements. For example, if RME is close to one, then we can estimate that the requirements will mature during the requirements analysis phase or on a needs basis. When the RME is small, this implies that it is hard to reach maturity and we must control the development process and cope with delayed elicitation or the possibility of requirements changes in the later stages of the project.

The goal of this paper is to define a method of estimating the RME of each software component. Since RAs elicit requirements by accessing requirement sources and stability is one of the characteristics of the requirements, we can assume that the RME of a component must be derived from the accessibility of the requirements source and the stability of the requirements.

In order to clarify the relationship between RME and two factors, i.e., the accessibility of the source of requirements and the stability of requirements, we quantitatively observed the processes for requirements elicitation in real projects and then statistically analyzed the results. The requirements were recorded in formal e-mails, minutes of meetings, defect reports, and issue reports. The records recounted the history of requirements elicitation. We used requirements categories as observation targets in observing the histories of requirements elicitation and counted the number of requirements for each observation target. The word "change" implies that the requirement was elicited later than the first elicitation. The word "volatility" of requirements implies that they were difficult to mature. We do not focus on the changing history of each requirement, but the requirements elicitation history of each software component. With this focal point, even requirements deletion can be regarded as a product of the activities of requirements elicitation. Hence, when we observe the history of requirements elicitation, we do not distinguish between added, deleted, or modified requirements. We count the number of requirements including the number of added, deleted, and modified requirements.

Requirements maturation is observed for each component. We observe the maturation of requirements from physical and/or logical viewpoints. Features and subsystems are examples of physical components. Usability, efficiency, and portability requirements are examples of logical components.

The remainder of this paper is organized as follows. Section 2 introduces related work, and Section 3 describes the research methods we used in our study. We define the requirements categories in order to observe the process for requirements maturation. We also describe the method by which we derive accessibility. Sections 4 presents case studies and discusses the results obtained from these studies and how we estimate the period for requirements maturation. Section 5 concludes the paper.

II. Related work

An RA should identify the stakeholders [6] who are the source of the requirements before eliciting requirements. Stapel et al. [7] referred to the source of the requirements as stores of information. These stores are classified into solid and fluid stores. Books, formal e-mails, and tapes are examples of solid stores that are long-term accessible, repeatable, and comprehensible by third parties. The knowledge of the individuals involved and informal notes are classified as fluid stores. We take into account both solid and fluid stores as stakeholders.

A stakeholder map that depicts the environment of a project effectively determines the distance between the RA and stakeholders. The relationship map that was introduced by Gottesdiener represents the organizational situation by presenting the relationship between developers and clients [8] and provides stakeholders related to the requirements. Alexander and Robertson proposed an onion model as a stakeholder map [9]. The onion model is a project structure that includes developers, clients, and the environment. Core team members are placed at the core of the onion, and they participate in the project on an as-needed basis. Peripheral stakeholders are placed in a ring around the core and they can be functional, political, or financial beneficiaries, or technical or subject matter experts. They recommended that the map for each project should be modeled, after which the RA contacts stakeholders and starts to elicit requirements. Requirements-gathering collaborative networks (RGCNs) provided five collaboration structure patterns among developers [10]. These models are useful for understanding the numbers and/or types of stakeholders to be interviewed. However, the models were not designed to determine accessibility to the sources of requirements. In this paper, we adapt the onion model to establish accessibility to the sources of requirements.

Loconsole et al.[11] defined the term volatility as "the amount of changes to a use case over time." Their results imply that volatility is a characteristics of requirements written in use cases. We should take the stability of requirements into account when analyzing requirements maturation.

Requirements changes can be considered from multiple perspectives. The first is the viewpoint of system development. Fickas and Feather monitored requirements changes and analyzed their causes associated with system maintenance [12]. Sutcliffe et al. discussed requirements changes associated with the contextual uncertainty of the system [13]. In this paper, we consider another perspective from the viewpoint of requirements elicitation.

There are two focuses for requirements changes, i.e., from the process and product viewpoints [14]. Anderson et al. focused on the product viewpoint [15]. They analyzed the history of requirements changes within the context of the development of complex, embedded, real-time, and safety-critical systems with 22 releases. They made five remarks and found a linear relationship between the number of changes occurring in a requirements specification and its size [16]. They made less mention of what caused the delay in requirements maturation, but discussed process in the continuous evolution of requirements. We take into account the causes of delay in requirements maturation in order to estimate the requirements maturation.

Requirements changes can be classified according to their origins, which are related to development environments, stakeholders, development processes, understanding of requirements, and relations between requirements [17]. Nurmuliani et al. [18] analyzed volatility by collecting data on requirements changes. They identified the reasons for and the types and origins of the requirements changes and then classified them. The categories for the origins are not suitable for managing the process for requirements elicitation. Requirements can be classified according to their sources and volatilities. Nakatani et al. [19] defined the taxonomy of requirements. Their taxonomy was categorized into three levels based on the volatility of the requirements. The taxonomy of requirements can be applied to identify the target in observing maturation.

Nurmuliani et al. [18] proposed the measurement of requirements volatility. Their measure does not represent historical information on activities of requirements elicitation. If we focus on the process for requirements maturation, the measure of volatility requires a time variable. Nakatani et al. defined volatility with a time variable, t, as the requirements maturation ratio, RMR [20]. Here, RMR(t) indicates how much the requirements of the observation target had matured over time, t, as follows:

$$RMR(t) = \frac{R(t)}{R_T} * 100 \quad (2)$$

The $R(t)$ in this equation is the cumulative number of requirements for the observation target, i.e., the sum of the cumulative added, deleted, and modified requirements at time t. When a project is started, the requirements maturation ratio $RMR(0)$ of every target is zero, and when the project is completed, the requirements maturation ratio $RMR(end)$ of the target is 100. This measure represents historical information on the activities of requirements elicitation.

Observations of requirements elicitations have revealed that requirements have had a unique reason to be elicited within certain periods of time. The PRINCE model categorized the processes for requirements maturation into four types [20], viz., the E-, M-, L-, and U-types, whose requirements reached maturity in the *e*arly, *m*iddle, *l*ate, and *u*nexpected periods of the project. Although the PRINCE model can be used as a guide to planning the process for requirements elicitation, this model does not provide a way of actually *planning* the process and/or *estimating* the maturation period. We clarify the causes that affect the efficiency of requirements maturation in order to plan the process.

III. Research approach

This section describes the methods we used in our research. We need to estimate RME with known information to plan the process for requirements elicitation. The purpose of our research is to clarify the relationship between RME and factors that affect the process for requirements elicitation. We have explored the relationship in past projects.

We determine observation targets for analyzing the process of requirements elicitation. According to IEEE std. 830-1998, requirements can be ranked by stability. Therefore, we can assume that the stability of requirements can be determined early in the project. In addition, communication difficulties must affect RME. Before we elicit requirements from stakeholders, we have to list stakeholders on a stakeholder map. Therefore, we can analyze difficulties with communication at an early stage of the project by using the stakeholder map. The skill level of an RA is quantified by the skill level of the IT architect in the Skill Standards for IT professionals (ITSS) [21], [22]. We intende to take into account the skills of RAs in the future.

We introduce a method by which we derive RME from the stability of requirements and accessibility to the sources of requirements at an early stage of the project.

A. Category of observation targets

Before quantitatively observing the process for requirements elicitation, we must identify the observation targets. There are two possible types of targets: physical and logical components. Physical components include software components, objects, modules, classes, and use cases, whereas logical components include quality components that are independent of applications. A requirement can belong to multiple components. For example, the requirement "retrieve information of a book from the database of library within three seconds" belongs to a book management component, a functional requirements set, and a performance requirements set. The first category is a physical component, and the second and third categories are logical components.

Logical observation targets in requirements taxonomy [19] are categorized into a structure with three layers. Observation targets in the first layer are organized into two categories depending on their purpose, which can be either strategic (St) or business support (Bz).

- St:
  Strategic requirements are owned by the business representative and are intended to satisfy the business strategy. Strategic requirements are basically stable unless the business environment changes.
- Bz:
  Business requirements support various activities and include requirements for improving productivity in business. Business requirements may be altered by internal evolution within the organization of the end users.

There are functional (F) and non-functional (NF) categories under the first layer. There are interface, control, and entity components [23] in the third layer under the Type-F requirements. There are quality or constraint requirements in accordance with ISO/IEC 25030 [24] under the the type-NF requirements layer. For example, there are reliability (Rel), efficiency (Eff), and constraint requirements, including design constraints (Dc), operational environment constraints (OEc), and project resource constraints (PRc). These components are in the third layer.

When we observe the process for requirements elicitation in each category, we combine the names representing characteristics of requirements. For example, type-BzNF_Dc is related to business support by the non-functional requirements of design constraints. An observer of the history of requirements elicitation can select and integrate these categories as a set of requirements for the observation target in every system.

When we observe the history of requirements elicitation, we do not distinguish between addition, deletion, or modification of requirements. Addition, deletion, and modified requirements are counted as delayed elicitation. They indicate difficulties with requirements maturation. A requirement found in the minutes of a meeting is categorized into a corresponding category and is recorded with the date of the meeting. One sentence sometimes includes several requirements. In such a case, each requirement in the sentence is identified and counted for the corresponding category. One requirement usually has several aspects, i.e., physical and logical aspect. Such a requirement is counted multiple times in order to observe the requirement for each aspect.

B. Stability

Requirements are ranked by stability according to IEEE std. 830-1998. If the requirements of an observation target are sufficiently stable, then they may not be changed after being elicited. Stability is measured by the Requirements Maturity Index (RMI). The RMI was defined by Anderson et al. [15] along with the second implementation of the Software Maturity Index (SMI) [25]:

$$RMI_2 = \frac{(R_T - R_C)}{R_T} \tag{3}$$

where R_T is the total number of requirements elicited, and R_C is the number of requirements that were added, modified, or deleted after the baseline requirements were defined. The change and addition of requirements are tracked in this implementation. We explore the period for requirements maturation. Thus, we do not take into account the traceability of requirements, but rather trace the elicitation history of the requirements set for observation targets. The deletion of a requirement is treated in the same manner as the addition of a requirement. These requirements are the products of requirements elicitation. The total number of current requirements is equal to the number of previous requirements plus the number of current base-line requirements that were added to the previous baseline minus the number of requirements that were deleted from the previous baseline.

The first implementation of the RMI is:

$$RMI_1 = \frac{R_T - (R_a + R_c + R_{del})}{R_T}, \tag{4}$$

where R_a, R_c, and R_{del} correspond to the number of current baseline requirements that have been added, modified, and deleted. Using equation 4, a requirement derived from an existing requirement by decomposition will count as one requirement.

C. Accessibility

Communication between an RA and the requirements source is one of the steps required for requirements elicitation. Accessibility is the ease with which an RA can interact with the source of the requirements, e.g., end users, laws, or technical or application environments.

Accessibility may be impacted by several parameters, such as geographical distance, how busy a stakeholder is, or his/her job priorities. We define requirements source accessibility (RSA) with the following equation in order to facilitate accessibility. It depends on the maximum communication distance between the RA and the source of requirements.

$$RSA = \frac{1}{Max(NIF)} \tag{5}$$

The NIF in this equation is the number of information flows from the source of the requirements to the RA. If the source of the requirements is core stakeholders, its NIF is one, because the RA can directly communicate with the core stakeholders. The RSA of peripheral stakeholders equals 1/2 and that of the project environment equals 1/3. When an RA elicits a requirement from a peripheral stakeholder via a core stakeholder, NIF is two and thus the RSA is 1/2.

We apply Stapel et al.'s FLOW notation [7] as a stakeholder map and visualize the communication distance between the RA and the requirements source. The distance does not imply geographical distance but rather difficulties and/or troublesome communications.

The next section introduces two case studies in the history of requirements elicitation and discusses a method by which we derive RME from RMI and RSA.

IV. Case Study

We conducted two case studies. The first was an education support system developed for improving classroom lectures for 200 or more attendees within non-computer classrooms [26]. The second was an order management system for restaurants [27].

A. Case 1: p-HInT

- Overview

  The first case called p-HInT system involved an education support system developed by Hannan University in Matsubara, Osaka, Japan to improve classroom lectures with 200 or more students in non-computer classrooms [26]. The students in the p-HInT classroom used gaming equipment or mobile phones with Wi-Fi interfaces, and the system helped the lecturer rapidly evaluate the understanding of the students. The network provided one-to-one-like communication with the lecturer and his/her students.

  We monitored the project for 90 weeks, which was how long the project lasted. Systems were released three times during the development period, and the client finalized a contract with the developer around week 29. The project proceeded with good cooperation by the core stakeholders. They frequently communicated via e-mails and their face-to-face communications were documented on videotape and in the minutes of meetings.

The project manager on the client's side was a representative of the lecturers. She interviewed students and lecturers about new ideas, complaints, and the situation in their classes. The project was not influenced by any changes in the social environment. One difficulty with the system was that neither the client nor the developers knew what the Wi-Fi performance of the gaming equipment was. In order to solve the problem, the core stakeholders needed to learn about the efficiency of network communication from the lecturers and students in the classroom in which the prototype system was installed.

Hannan University provided all the documents used in the present study, including the SRSs, issue reports, minutes of meetings, and e-mails. We quantitatively analyzed these documents, and we interviewed the core stakeholders for the qualitative analysis.

- Observation

The history of requirements elicitation for the p-HInT project is shown in Figure 1. It shows the evolutionary process for the requirements of logical components.

The x-axis represents the duration of the project, and the y-axis represents the requirements maturation ratio (RMR). The denominator of RMR is the number of requirements at the end of the development of their third version. The three vertical lines represent the start periods for the three development cycles. Here, mtrP represents the 100% maturation period for each version. The RME are derived for all logical components with their mtrP and durations of development.

The stakeholder map for the project is outlined in Figure 2. Fluid information from the classrooms was given by the students to the lecturers' representative, who then discussed the requirements with the RA. The NIF of the requirements on Wi-Fi efficiency in this situation was three with four nodes, i.e., the actual classroom, the lecturers or students, the lecturers' representative, and the RA. Numerous requirements were introduced by the lecturers' representative who was the project manager and one of the core stakeholders. There was one information flow from the lecturers' representative to the RA.

The correlated efficiency between stability and accessibility was 0.298. We could not conclude that these two factors were dependent variables.

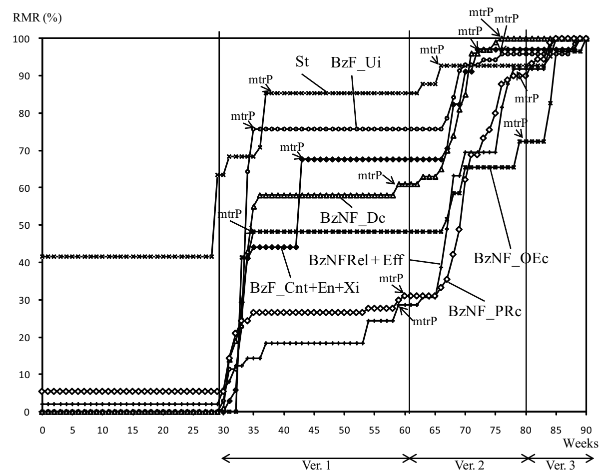

Figure 1. History of requirements maturation ratio of p-HInT.

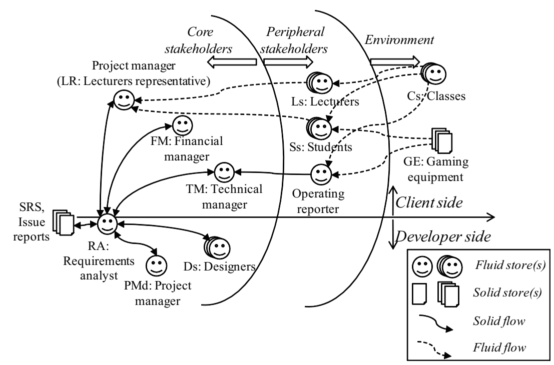

Figure 2. Stakeholder map of p-HInT.

B. Case 2: RESORT

- Overview

The *RESORT* system was developed as a product for restaurants. It received orders from staff members through hand-held terminals and guests through table terminals. Table terminals were placed on the tables of guests and they provided a menu, recipes, and information about alcoholic beverages and entertainment by means of a wireless network from the application server. The previous system was developed several years ago, and the sponsor wanted to upgrade it in order to improve its installation load and extensibility. The new system had to be able to connect to various order entry system (OES) components. The duration of the project was five months. The data we analyzed was collected over 117 days from the external design phase until the end of the project.

- Observation

The stakeholder map for the project is outlined in Figure 3. Three companies cooperated in developing the table terminals, the application server, and the target system. They communicated via a message DB

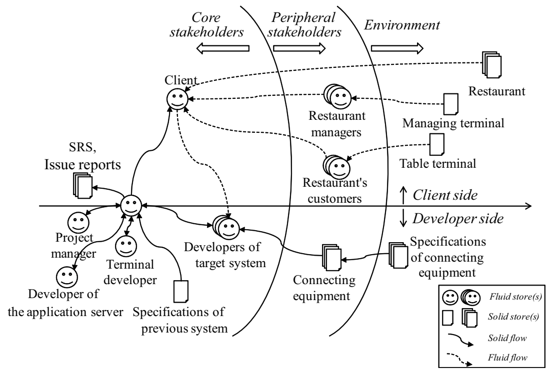

Figure 3. Stakeholder map of RESORT.

provided by the client. Issue reports, the minutes of meetings, and specifications were stored in the message DB. The specifications of the previous system were also provided by the client. The specifications of the external systems were not provided because these were owned by competing companies.

There were no experts on the restaurant management system in the developers' team including the RA. The client provided knowledge to the developers to solve the problem. Hence, there were two information flows from the restaurant to the RA (via the client).

The correlated efficiency between stability and accessibility was 0.567. We could not conclude that these two factors were dependent variables. Therefore, we could apply them to multiple regression analysis as two parameters.

C. Results and discussion

- Multiple regression analysis

  We propose a three-dimensional space to analyze the process for requirements elicitation with requirements accessibility, stability, and the abilities of an RA. The present paper focuses on requirements accessibility and stability. The accessibility, RSA, is derived from the number of information flows from the sources of requirements to the RA. The stability of requirements is a characteristic of requirements and is calculated with Eq. (3).

  The RME is a measure of requirements maturation and represents how quickly the requirements reach 100% maturation. The correlation coefficient between the stability and accessibility of the first and the second case were 0.297 for the former and 0.567 for the latter. Thus, we could treat these two factors as independent variables. In order to obtain the equation of the RME with these two factors, we applied multiple regression analysis to these cases. The data for multiple regression analysis have been summarized in Table I, and RME was calculated with Eq. (1). The RSA in Table I was derived from the number of information flows from the sources of requirements to the RA in Figures 2 and 3. RMI was calculated by using the duration of development and the point of requirements maturation in the history of requirements elicitation.

  According to multiple regression analysis with a significance level of 5% (p=0.002), RME can be predicted by using two factors with the following equation:

  $$RME = -0.31 + 0.27 * RMI_2 + 0.75 * RSA$$

  .

  There may be other parameters involved as mentioned in the definition of accessibility. For example, some developers have complained about their stakeholders being too busy. If stakeholders are too busy to answer the developer's questions, then accessibility decreases. The priority of development may also have an impact on the accessibility. More precise measurements of accessibility will be defined in future works.

  Since the formula needs more evaluation with other cases, we can now interpret the equation as follows. If the source of the requirements is sufficiently close to the RA and the requirements set is sufficiently stable, RME is then predicted to be 0.71. This is the maximum value of RME. However, if the source of the requirements is too far to be elicited by the RA and the requirements set is unstable, then RME is predicted to be -0.31. The requirements with negative RME implies that they will not mature by the end of the project, if the RA does not deal with the problems of accessibility and stability.

  We assumed that the RA of a project would draw a stakeholder map at an early stage of the project, so that it would be possible to define the accessibility of the requirements source at an early stage of the project. Thus, the problem of how to estimate stability, which is measured by RMI_1, emerges. We will be able to estimate the RMI_1 of all requirements categories by continuously observing the process of requirements elicitation in past and future projects. For example, we know that the requirements for the user interface and/or presentation are unstable and have small RMI_1. Based on other observations [28], the RMI_1 of portability is smaller than that of reliability, and the RMI_1 of reliability is smaller than that of maintainability. This is understandable, because, from the point of view of users, portability is more important than the reliability of the system. The ability of maintain the system is a concern for developers. Observing projects provides a great deal of knowledge on stable and unstable requirements.
- RME and maturation types

  RME can take a value from -0.31 to 0.71. Thus far, we have been able to divide the duration of projects

Table I
STABILITY AND ACCESSIBILITY TO REQUIREMENTS CALCULATED FOR TWO CASES.

#Case	Ver.	Observation target	Source of requirements	Accessibility RSA=1/(NIF)	Stability =RMI_1	RME
1	1	St	Lecturers' representative	1.000	1.000	0.727
		BzF_Ctl+En+Xi	Lecturers' representative	1.000	1.000	0.515
		BzF_Ui	Lecturers, Students	1.000	1.000	0.788
		BzNF_OEc	Technical manager	1.000	1.000	0.788
		BzNF_Dc	Technical manager, Designers, Classes	0.333	0.951	0.061
		BzNF_PRc	Technical manager, Financial manager, Gaming equipment	0.333	0.857	0.030
		BzNF_Rel+Eff	Lecturers' representative, Lecturers, Students	0.333	0.643	0.061
	2	St	Lecturers' representative	1.000	1.000	0.750
		BzF_Ctl+En+Xi	Lecturers' representative, Lecturers, Students	0.500	1.000	0.450
		BzF_Ui	Lecturers, Students	0.500	1.000	0.250
		BzNF_OEc	Technical manager, Financial manager, Classes	0.333	0.714	0.100
		BzNF_Dc	Technical manager, Designers, Classes	0.333	1.000	0.250
		BzNF_PRc	Technical manager, Financial manager, Gaming equipment	0.333	0.642	0.150
		BzNF_Rel+Eff	Gaming equipment, Classes	0.333	0.645	0.150
	3	St	Lecturers' representative	1.000	1.000	0.545
		BzF_Ctl+En+Xi	Lecturers' representative, Lecturers, Students	0.500	0.000	0.182
		BzF_Ui	Lecturers, Students	0.500	0.000	0.182
		BzNF_OEc	Technical manager, Financial manager, Classes	0.333	0.875	0.091
		BzNF_Dc	-	-	-	-
		BzNF_PRc	Technical manager, Financial manager, Classes	0.333	1.000	0.545
		BzNF_Rel+Eff	Gaming equipment, Classes	0.333	1.000	0.545
2		TTC	Client, Restaurant managers, Restaurant customers	0.500	0.426	0.025
		MT	Client, Restaurant managers	0.500	0.820	0.025
		IS	Client	1.000	1.000	0.717
		TTM	Client	1.000	1.000	0.833
		CS	Developers of application server	1.000	1.000	0.725
		OESM	Specifications of the external equipment	0.333	0.577	0.017
		OESC	Specifications of the external equipment	0.333	0.851	0.000
		DB	Specifications of the previous system	1.000	0.667	0.608

into E, M, and L stages in the PRINCE model [20]. If the RME of a requirements set is smaller than 0.24 which is one third of 0.71, the type of maturation of the requirements set is estimated to be an L-type. The L-type indicates that the requirements will not be able to be elicited until the late stage of the project. Similarly, if the RME of a requirements set is greater than 0.47, which is two thirds of 0.71, then the type of maturation of the requirements set is estimated to be an E-type, and if the RME of a requirements set is in the rest, the type of maturation of the requirements set is estimated to be an M-type. The project should be able to cope with low accessibility for cases in which projects cannot accept L-type maturation.

V. CONCLUSION

We analyzed the relationship between the requirements maturation efficiency (RME) and accessibility to the sources of requirements and stability. As a result of multiple regression analysis, RME was able to be predicted with the accessibility and stability of requirements.

We proposed a method by which the process foe requirements elicitation could be planned. The results obtained from the present research demonstrate that if the source of requirements is close to an RA and they are stable, then they can be elicited in the early stage of a project and they will be E-type maturation. Furthermore, if the source of the requirements is far from an RA and the requirements are volatile, then the requirements cannot be elicited in the early stage of the project and they will be M or L-type maturation. The equation revealed that the number of information flows from the requirements source to the RA was the only factor in controlling requirements maturation, because the stability of requirements is one of their characteristics.

In order to plan the process for requirements elicitation, a stakeholder map should be drawn and sources for all requirements sets should be identified. We should explore the requirements elicitation by using other projects and lead the RMI_1 of requirements categories for more precise predictions.

ACKNOWLEDGMENTS

The present study was supported in part by the Joint Forum for Strategic Software Research (SSR) and by KAKENHI (21500029). The authors would like to thank all parties who participated in this project. In particular, we would like to thank Professor Noriko Hanakawa and Mr. Masaki Obana of Hannan University for kindly providing the data on requirements.

References

[1] D. Zowghi and N. Nurmuliani, "A study of the impact of requirements volatility on software project performance," *Asia-Pacific Software Engineering Conference*, pp. 3–11, 2002.

[2] Y. K. Malaiya and J. Denton, "Requirements volatility and defect density," *International Symposium on Software Reliability Engineering*, pp. 285–299, 1999.

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

[4] D. M. Berry and E. Kamsties, "Chapter 2: Ambiguity in requirements specification," in *Perspectives on Software Requirements*, J. C. S. D. P. Leite and J. J. H. Doorn, Eds., 2004, pp. 7–44.

[5] M. Cataldo and S. Nambiar, "On the relationship between process maturity and geographic distribution: an empirical analysis of their impact on software quality," in *The 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering*, ser. ESEC/FSE '09. ACM, 2009, pp. 101–110.

[6] S. Robertson and J. Robertson, *Requirements-Led Project Management*. Addison-Wesley, 2005.

[7] K. Stapel, E. Knauss, and K. Schneider, "Using flow to improve communication of requirements in globally distributed software projects," *Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills*, pp. 5–14, 2009.

[8] E. Gottesdiener, *Requirements by Collaboration*. Pearson Education, 2002.

[9] I. Alexander and S. Robertson, "Understanding project sociology by modelling stakeholders," *IEEE Software*, vol. 21, no. 1, pp. 23–27, 2004.

[10] P. Laurent and J. Cleland-Huang, "Requirements-gathering collaborative networks in distributed software projects," *Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills*, pp. 26–30, 2009.

[11] A. Loconsole and J. Borstler, "An industrial case study on requirements volatility measures," in *Asia-Pacific Software Engineering Conference (APSEC'05)*. IEEE, 2005, pp. 249–256.

[12] S. Fickas and M. S. Feather, "Requirements monitoring in dynamic environments," in *The 3rd IEEE International Conference on Requirements Engineering*. IEEE, 1995, pp. 140–147.

[13] A. Sutcliffe, S. Feckas, and M. M. Sohlberg, "Personal and contextual requirements engineering," in *The 13th International Requirements Engineering Conference (RE'05)*. IEEE, 2005, pp. 19–30.

[14] S. Anderson and M. Felici, "Requirements evolution: From process to product oriented management," in *Proceedings of the 3rd International Conference on Product Focused Software Process Improvement (Profes 2001), LNCS 2188*. Springer-Verlag, 2001, pp. 27–41.

[15] ——, "Controlling requirements evolution: An avionics case study," in *SAFECOMP 2000 (LNCS 1943)*. Springer-Verlag, 2000, pp. 361–370.

[16] ——, "Quantitative aspects of requirements evolution," in *Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment (COMPSAC '02)*. IEEE, 2002, pp. 27–32.

[17] M. R. Strens and R. C. Sugden, "Change analysis: A step towards meeting the challenge of changing requirements," in *IEEE Symposium and Workshop on Engineering of Computer Based Systems (ECBS'96)*, 1996, pp. 278–183.

[18] N. Nurmuliani, D. Zowghi, and S. Fowell, "Analysis of requirements volatility during software development life cycle," in *The 2004 Australian Software Engineering Conference (ASWEC'04)*, 2004, pp. 28–37.

[19] T. Nakatani, N. Kondo, M. Inoki, M. Tsuda, and K. Katamine, "The taxonomy of requirements for conducting elicitation," in *Information Systems 2011 (IS2011)*, 2011, pp. 161–168.

[20] T. Nakatani, S. Hori, M. Tsuda, M. Inoki, K. Katamine, and M. Hashimoto, "Towards a strategic requirements elicitation: A proposal of the PRINCE model," in *The 4th International Conference on Software and Data Technologies (ICSOFT 2009)*. INSTICC, 2009.

[21] IPA: Information-technology Promotion Agency, "The skill standards for it professionals V3 2008 English edition," 2008. [Online]. Available: http://www.ipa.go.jp/english/humandev/data/ITSS_V3_2008Eng.zip

[22] ——, "ITSS (Skill Standards for IT Professionals) overview and worldwide promotion," 2010. [Online]. Available: http://events.grace-center.jp/symposium/2010/sites/default/files/ITSSOverviewandWorldwidePromotion_20100316.pdf

[23] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, *Object-Oriented Software Engineering*. Addison-Wesley, 1992.

[24] ISO/IEC 25030:2007, "Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements," 2007.

[25] IEEE Std 982.1, "IEEE standard dictionary of measures to produce reliable software," 1988.

[26] N. Hanakawa and M. Obana, "Mobile game terminal based interactive education environment for large-scale lectures," in *The Eighth IASTED International Conference on Web-based Education (WBE2010)*, March 2010.

[27] T. Nakatani, S. Hori, N. Ubayashi, K. Katamine, and M. Hashimoto, "A case study: Requirements elicitation processes throughout a project," in *The 16th International Requirements Engineering Conference (RE'08)*. IEEE, 2008, pp. 241–246.

[28] The PRINCE Project, "The PRINCE model: approach to the integrated requirements process (in Japanese)," Joint Forum for Strategic Software Research, 2010.