IEICE TRANSACTIONS

on Information and Systems

VOL. E97-D NO. 5
MAY 2014

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY

The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER Special Section on Knowledge-Based Software Engineering

Estimation of the Maturation Type of Requirements from Their Accessibility and Stability

Takako NAKATANI[†a)], ***Member***, **Shozo HORI**[††], ***Nonmember***, **Keiichi KATAMINE**[†††], ***Member***, **Michio TSUDA**[††††], ***and*** **Toshihiko TSUMAKI**[†††††], ***Nonmembers***

SUMMARY The success of any project can be affected by requirements changes. Requirements elicitation is a series of activities of adding, deleting, and modifying requirements. We refer to the completion of requirements elicitation of a software component as requirements maturation. When the requirements of each component have reached the 100% maturation point, no requirement will come to the component. This does not mean that a requirements analyst (RA) will reject the addition of requirements, but simply, that the additional requirements will not come to the project. Our motivation is to provide measurements by which an RA can estimate one of the maturation periods: the early, middle, or late period of the project. We will proceed by introducing the requirements maturation efficiency (RME). The RME of the requirements represents how quickly the requirements of a component reach 100% maturation. Then, we will estimate the requirements maturation period for every component by applying the RME. We assume that the RME is derived from its accessibility from an RA to 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 further, model stability with the requirements maturation index (RMI). According to the multiple regression analysis of a case, we are able to get an equation on RME derived from these two factors with a significant level of 5%. We evaluated the result by comparing it to another case, and then discuss the effectiveness of the measurements.

key words: *requirements accessibility, requirements stability, requirements maturation*

1. Introduction

Though, incomplete, inaccurate, or vague requirements are potential risks in any projects [1], [2], requirements are sometimes changed in order to correct problems, improve requesters' satisfaction, and to adapt the system to the requesters' future environments.

There are several factors that cause 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. New requirements are also discovered by refining the original requirements. Some kinds of requirements are changed as a result of their own characteristics. We designate the characteristic as volatile [5] or stable.

The communication distance between clients and an RA can be one of the factors that affects the process of requirements elicitation. If the clients are too busy to respond to e-mails from the RA, it may be difficult for their requirements to mature within the expected period. However, if the RA can maintain good communication with their clients, he/she may be able to elicit precise and unambiguous requirements within the expected schedule and specify them on time.

If RAs can estimate the period for the maturation of requirements, they will be able to cope with obstacles that delay the requirements maturation. In this paper, we refer to the completion of requirements elicitation of a software component as requirements maturation. In other words, if a set of requirements of a component has matured 100%, further requirements will not be added to the set, will not be modified, and nor will any be deleted. This does not mean that the RA will reject any additional requirements, but simply, the requirements will not come to the project after its maturation. The minutes of meetings are good materials that tell a history of requirements maturation from the initiation of the project, to the closing of the project. From the minutes of meetings, we can accumulate knowledge on the histories of each kind of software component. After the accumulation of such knowledge, our challenge is to estimate the period for requirements maturation.

Our motivation is to provide measurements by which an RA can estimate the period for requirements maturation. We define requirements maturation efficiency (RME) as a measure of requirements maturation. 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 those factors. Thus, we define measurements for accessibility of requirements sources and requirements stability, and clarify the relationship between RME and those two factors. We proceed by applying multiple regression analysis within a case study in order to get an equation for RME, which is derived with a significant level of 5% by these two factors: measurements for accessibility of requirements sources and requirements stability.

Manuscript received June 11, 2013.
Manuscript revised October 30, 2013.
[†]The author is with The University of Tsukuba, Tokyo, 112–0012 Japan.
[††]The author is with Yaskawa Information Systems Corporation, Kawasaki-shi, 215–0004 Japan.
[†††]The author is with Kyushu Institute of Technology, Iizuka-shi, 820–0067 Japan.
[††††]The author is with Osaka University, Suita-shi, 565–0871 Japan.
[†††††]The author is with National Institute of Informatics, Tokyo, 101–8430 Japan.
a) E-mail: nakatani@gssm.otsuka.tsukuba.ac.jp
DOI: 10.1587/transinf.E97.D.1039

Before applying such statistical analysis, the minutes of meetings of actual projects were quantitatively observed in order to understand the history of requirements maturation. When we observed the history, we did not trace the changing history of each requirement, but the requirements elicitation history of each *software component*. With this focal point, even requirements deletion could be regarded as a product of the activities of requirements elicitation. Hence, when we observed the history of requirements elicitation, we only counted the number of requirements, including the number of added, deleted, and modified requirements.

According to our previous study [6], we found that the histories of requirements maturation could be categorized by the characteristics of the software component. We categorize software components into types based on their characteristics: volatility and/or software architectural position.

The remainder of this paper is organized as follows. Section 2 introduces related work, while Sect. 3 describes the research methods with requirements categories that are defined to observe the process for requirements maturation. Section 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.

2. Related Work

An RA should identify the stakeholders 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.

Cataldo and Nambiar [8] clarified that the geographic distance between engineers working on a project affects the quality of the software. Their concept of "distance" can be applied to the requirements elicitation process. We consider the communication distance, in other words, the number of communication flows, as one of the factors that affect the requirements maturation.

A stakeholder map that depicts the environment of a project effectively determines the distance between an RA and stakeholders. The relationship map that was introduced by Gottesdiener represents the organizational situation by presenting the relationship between developers and clients [9] and provides stakeholders related to the requirements. Alexander and Robertson proposed an onion model as a stakeholder map [10]. The onion model is a service structure that includes developers, clients, and the environment. S. Robertson and R. Robertson proposed another stakeholder map [11]. In their map, the core team members are placed in the center of the map, and they participate in the project on an as-needed basis. Peripheral stakeholders are placed in a ring around the core members and can be functional, political, or, financial beneficiaries, or technical or subject matter experts. Requirements-gathering collaborative networks provided five collaboration structure patterns among developers [12]. 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 from the RA to the sources of requirements. In this paper, we adapt the onion model to establish accessibility from the RA to the sources of requirements.

Loconsole et al. [13] defined the term volatility as "the amount of changes to a use case over time." Their results imply that volatility is a characteristic of requirements written in use cases. We take the stability of requirements into account when analyzing requirements maturation.

Requirements changes can be analyzed from multiple perspectives. The first viewpoint is the system development. Fickas and Feather monitored requirements changes and analyzed the causes from these changes in association with system maintenance [14]. Sutcliffe et al. discussed requirements changes associated with the contextual uncertainty of the system [15]. There are other focuses for requirements changes, i.e., from the process and product viewpoints [16]. Anderson et al. focused on the product viewpoint [17]. 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 [18]. They made less mention of what caused the delay in requirements maturation, but discussed the process in the continuous evolution of requirements. In this paper, we focus on the causes that affect the requirements maturation as seen from the product and process viewpoints.

Nurmuliani et al. [5] 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. On the other hand, 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 subject while observing maturation.

Nurmuliani et al. [5] also 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 maturation process with a time variable, t, as the requirements maturation ratio, RMR [20]. Here, $RMR(t)$ indicates how much the requirements of an observation subject had matured over time, t, as follows:

$$RMR(t) = \frac{R(t)}{R_T} * 100 \quad (1)$$

The $R(t)$ in this equation is the cumulative number of requirements for the observation subject, i.e., the sum of the cumulative added, deleted, and modified requirements at time t. R_T stands for the total number of requirements elicited through till the end of the project. When a project is started, the requirements maturation ratio $RMR(0)$ of every subject is zero, and when the project is completed, the requirements maturation ratio $RMR(end)$ of the subject becomes 100. This measure represents historical information on the activities of requirements elicitation. However, they did not mention the way to estimate the $RMR(t)$.

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]: the E-, M-, L-, and U-types, whose requirements reached maturity in the *early*, *middle*, *late*, and *unexpected* 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. In this paper, we clarify the causes that affect the efficiency of requirements maturation and the way to estimate the types of requirements maturation.

A decision tree is proposed to infer those maturation types of an observation subject from the information known in the early stage of the project [21]. However, the decision tree does not provide us any quantitative measurements. We propose measurements to estimate the maturation type of each observation subject.

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 [22]. We do not reconstruct the requirements changes classification, but build analysis space of requirements elicitation with two factors: requirements accessibility, and requirements stability. In the analysis space, requirements accessibility can include factors of stakeholders, and requirements stability can take into account other origins of requirements changes.

3. Research Approach

This section describes requirements categories and measurements that we applied into our research.

3.1 Overview

We need to estimate the RME according to how easy or hard it is to mature the requirements of each component with known information. Thus, the purpose of our research is to clarify the relationship between RME and factors that affect the process for requirements elicitation. The process of our research is as follows.

1. Identify observation subjects in order to analyze the process of requirements elicitation.
2. Determine factors that affect the process of requirements elicitation.
3. Explore the relationships among the factors in past projects.
4. Evaluate the relationship in comparison with another project.

In the following two sections, we discuss the first two processes, and we explore the relationship with the first case and evaluate the result of the third process with comparison to the second case.

3.2 Category of Observation Subjects

The maturation process is observed for each software component. Such software components are the observation subjects. Before quantitatively observing the process for requirements elicitation, we must identify the observation subjects. When every project observes and records the history of requirements elicitation according to the common categories, they will be able to share their knowledge on the requirements maturation processes based on the categories.

There are two possible types of subjects: 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 physical and/or logical 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, while the second and third categories are logical components.

By eliciting requirements, the requirements are put into the set of requirements of a certain category according to their volatility and/or their position in the software architecture. When we analyze the maturation process for the set, the set can be categorized by the following attributes.

- Susceptibility to environmental changes
  If the observation subject is susceptible to environmental changes, the requirements of the subject must be volatile.
- Element in the software architecture
  If the observation subject is in the core of the software architecture, the requirements of the subject must be controlled to be stable or matured in the early stages of the project, while the requirements of the leaf of the architecture can be volatile.
- Requirements source
  Since, requirements are elicited from their source, the requirements maturation depends on their characteristics of their source. In order to understand the characteristics of the history of requirements maturation of each observation subject, it should also be categorized by its requirements source.

In general, the first two attributes do not depend on projects, while the last attribute does. In this section, we introduce categories for the first two attributes. For the last attribute, we apply new measurement in the next section as *accessibility*.

Logical observation subjects in requirements taxonomy [19] are categorized into a structure with three layers that represents the granularity of the subjects. Observation subjects in the first layer are organized into two categories depending on their susceptibility to environmental changes, which can be either strategic (St) or business support (Bz).

The requirements of TypeSt are owned by the business representative and are intended to satisfy the business strategy. They are basically stable unless the business environment changes. However, the business environmental changes cannot be controlled by stakeholders.

The requirements of TypeBz are defined to support various activities and include requirements for improving productivity in business. They may be altered by internal evolution within the organization of the end users. However, such alternations are negotiable among stakeholders. If it is needed, the requirements are controlled through reconciliation and/or rejection.

The differences between these two categories are uncontrollable or controllable. For example, in the later stage of the project, if a stakeholder requires new business requirements that cause rework in the software development, the stakeholders can reject the requirements. On the other hand, when a requirement of TypeSt is elicited and accepted, the stakeholders have to negotiate for triage [23] that may cause changes in other requirements, including deletions and additions. Therefore, categorizing and watching requirements according to these two categories is effective in managing the development of the project.

In the second layer, there are functional (TypeF) and non-functional (TypeNF) categories.

Under the TypeF, there are categories dependent on software architecture styles [24]. For example, if the developed system is applied to an object-oriented organization style, as well as a layered architectural style, we are able to define categories of interfaces, controls, and entities [25] in the third layer. We can define other categories for the different software architectural styles.

There are quality or constraint requirements in accordance with ISO/IEC 25030 [26] under the the TypeNF requirements layer. For example, there are reliability (TypeRel), efficiency (TypeEff), and constraint requirements (TypeC), including design constraints (TypeDc), operational environment constraints (TypeOEc), and project resource constraints (TypePRc).

A requirement found in the minutes of a meeting is categorized and put into a set of requirements corresponding to the category with the date of the meeting. When we observe the history for requirements elicitation in each category, we combine the names representing the characteristics of the requirements. For example, type BzNF_Dc is related to business support by the non-functional requirements of design constraints. An observer can select and integrate these categories as a set of requirements for the observation subject in every system. If a requirement belongs to multiple categories, the requirement is counted multiple times and the requirement may be observed in each category.

3.3 Factors that Affect the Requirements Process

Figure 1 shows a space of factors that affect the process of requirements maturation. The space has two dimensions: source accessibility and requirements stability.

According to IEEE std. 830-1998 [3], requirements are recommended to be ranked according to stability. Therefore, we can assume that the stability of each requirement is determined in the requirements specification.

Before we elicit requirements from stakeholders, we have to list stakeholders on a stakeholder map. Thus, we can analyze difficulties with communication at the early stages of the project by using the stakeholder map.

3.3.1 Stability

Requirements in a software requirements specification are recommended to be ranked according to their stability [3]. If the requirements of an observation subject are sufficiently stable, then they will not be changed after being elicited.

Stability is measured by the Requirements Maturity Index (*RMI*), of which there are two interpretations. The *RMI* was once defined by Anderson et al. [17] along with the second implementation of the Software Maturity Index (*SMI*) [27]. In this implementation, the addition, changes and deletion of each requirement are tracked. Since we explore the past projects and clarify the maturation period for a set of requirements for each observation subject, we do not take into account the traceability of requirements, but rather trace the elicitation history of the requirements set.

We define the *RMI* from the first implementation of *SMI* as follows.

$$RMI = \frac{R_T - (R_a + R_c + R_{del})}{R_T}, \qquad (2)$$

where R_a, R_c, and R_{del} correspond to the number of current baseline requirements that have been added, modified, and deleted. R_T is the total number of requirements from the

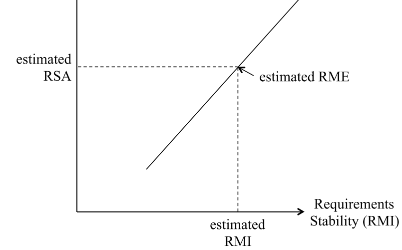

Fig. 1 The two-dimensional space of requirements elicitation.

initiation to the completion of the project. Using Eq. (2), a requirement derived from an existing requirement by decomposition will be also counted as one requirement of the same observation subject. The deletion of a requirement can be treated in the same manner as the addition and modification of a requirement. All of these requirements can be regarded as the products of requirements elicitation.

We apply RMI derived from Eq. (2) as the stability of each requirements set.

3.3.2 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.

We define requirements source accessibility (RSA) with the following equation. It depends on the *maximum* communication distance between the RA and the source of requirements.

$$RSA = \frac{1}{Max(NIF)} \tag{3}$$

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 one of the core stakeholders, its NIF is one, because the RA can directly communicate with the source. The NIF of peripheral stakeholders which surround the core stakeholders equals 2 and that of the project environment placed out side the peripheral stakeholders equals 3. Similarly, if the requirements source is accessed by the RA via n people, there are $n + 1$ communication flows between the requirements source and the RA. In this case, NIF is $n + 1$. Therefore, the RSA of those requirements sources are $1/2$, $1/3$, and $1/(n + 1)$, respectively.

We do not take into account the busyness of each stakeholder and/or personal characteristics, since these factors depend on personal situations and circumstances. They are unknown for an RA and/or a project manager.

In order to derive the number of communication flows, we can apply Stapel et al.'s FLOW notation [7] as a stakeholder map and visualize the communication distance between the RA and the requirements sources. The distance does not imply geographical distance, but rather difficulties and/or troublesome communications.

3.4 Actual Requirements Maturation Efficiency: RME_a

The ratio of requirements maturation of every observation subject reaches 100% by the end of the project, since the requirements elicitation activities have finished by the end of the project. RME_a represents how quickly the set of requirements of an observation subject has actually reached 100% maturation. If all of the requirements have been elicited by the final day of the project, the RME_a of the requirements must be zero. RME_a is observed by the following equation:

$$RME_a = 1 - \frac{l}{L} \tag{4}$$

L is the project's duration, and is measured in days or months from the initiation to completion of the project. The maturation period, l, is also measured in days or months from the initiation of the project to the requirements' maturation.

4. Case Study: Explore the Relationship among the Factors

The this section introduces two case studies in the history of requirements elicitation and discusses a method to derive RME from RMI and RSA.

4.1 Case 1: p-HInT

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 [28].

4.1.1 Overview

The students in the p-HInT classroom used gaming equipment or mobile phones with Wi-Fi interfaces, and the system helped the lecturer evaluate the understanding of the students.

We monitored the project for 90 weeks. Systems were released three times during the development, and the client finalized a contract with the developer in the 28th week. According to the core stakeholders, the project proceeded with good cooperation. The core stakeholders of the client's side were a project manager, financial manager and technical manager. The stakeholders of the developers' side were a project manager and designers. They are shown in the left part of Fig. 2.

They had 26 face-to-face meetings in 60 weeks. The 60 weeks include the development duration. Because of the schedule of the university, the meetings were mainly held during seasonal vacations, and the developer developed the system during the semesters. The meetings were held once a week, and as mentioned above, during the seasonal vacations of the university. The minutes of all meetings were recorded.

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 situations in their classes. 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 had to learn about the efficiency of the network of communication from the lecturers and students in the classroom who used the prototype system was installed.

Hannan University provided us the SRSs and minutes of meetings. We quantitatively analyzed these documents,

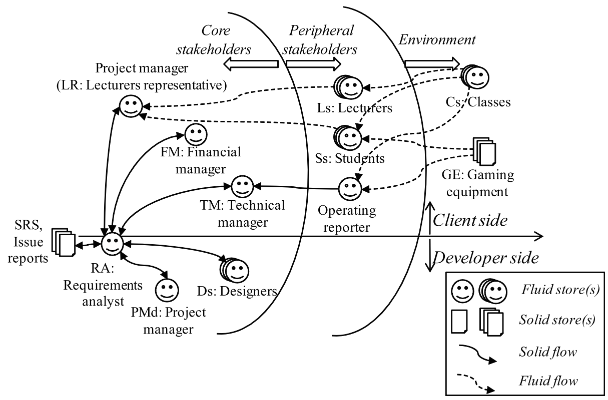

Fig. 2 Stakeholder map of p-HInT.

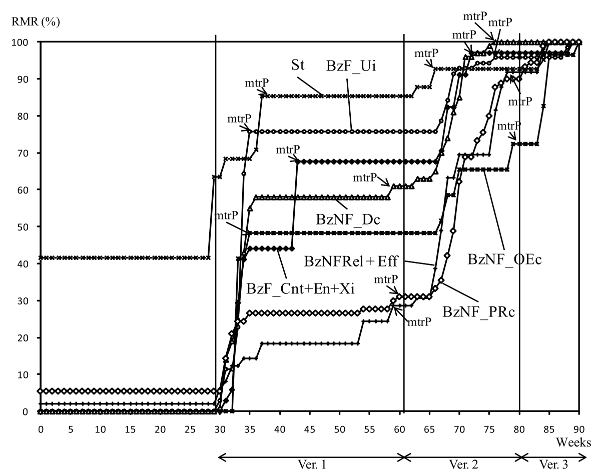

Fig. 3 History of requirements maturation ratio of p-HInT.

and we interviewed the core stakeholders in order to understand the context of the history of requirements maturation.

4.1.2 Observation

The history of requirements elicitation for the p-HInT project is shown in Fig. 3. 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*) shown in Eq. (1). The denominator of *RMR* is the number of requirements at the end of the development of the third version. The three vertical lines represent the start periods for the three development cycles. Here, mtrP represents the 100% maturation period for each component of every version.

We can derive *RSA* from Fig. 2 which outlines a stakeholder map for the project. A lot of requirements were introduced by the lecturers' representative who was the project manager. The situation is represented in the figure as an information flow from the lecturers' representative to the RA. Similarly, the figure shows that information from the classes to the students or lectures was fluid. The lecturers' representative, who then discussed the requirements with the RA, interviewed those students and lecturers. Hence, the NIF of the requirements on Wi-Fi efficiency in this situation was 3 with 4 nodes from the actual classes to the RA via the lecturers or students and the lecturers' representative. We applied two statistical analysis to the case study.

- Correlation analysis
  We evaluated the following null hypothesis by the correlation analysis.
  H0: The correlation coefficient between *RSA* and *RMI* is not 0.
  The correlated coefficient between *RMI* and *RSA* was 0.298 that being greater than 0, but with its probability at 0.189. Thus, we cannot accept the null hypothesis: H0 with the significance level at 5%. Therefore, *RMI* and *RSA* can be treated as independent variables from each other for the multiple regression analysis to derive RME.
- Multiple regression analysis
  The data for multiple regression analysis have been summarized in Table 1, and the *RSA* in Table 1 was derived from the number of information flows from the sources of requirements to the RA in Fig. 2.

RME_a was calculated with Eq. (4). The RMI and RME_a are derived for all logical components with their mtrP and durations of development. According to the minutes of meetings, the development of ver.1 was started from the 28th week, and the length of the term, L, of ver.1, ver.2, and ver.3 were 33, 20 and 11 weeks, respectively. Each term included a requirements analysis phase. We identify the end of the requirements analysis phase at the start point of the flat process of the requirements maturation ratio.

For example, the requirements analysis phase of *St* was finished in the 37th week of the project schedule. After the 37th week, any requirements were not added for *ST*. Thus, the *RMI* of *ST* is 1.000 in ver.1. In the same way, the *RMI*s of *ST* of ver.2 and 3 are 1.000. According to such observation, l of *St* is $37 - 28 = 9$. The RME_a of *St* in ver.1 is calculated as $1 - 9/33 = 0.727$ with Eq. (4).

Similarly, the requirements analysis phase of BzNF_OEc in ver.2 was from the 61st week to the 70th week, and the last requirements were elicited in the 79th week. The cumulative number of the requirements from the 61st week to the 70th week was 5, and the requirements were matured in the 79th week with 2 requirements. Thus, the *RMI* is $5/(5 + 2) = 0.714$. The RME_a of BzNF_OEc is $1 - (79 - 61)/20 = 0.100$.

According to multiple regression analysis, *R*, RME can be predicted by using two factors with the following equation:

$$RME = -0.21 + 0.67 * RSA + 0.23 * RMI \qquad (5)$$

Multiple R-Squared is 72.85%, and the adjusted R-square is 69.65%. This means that 69.65% of the data can be explained by Eq. (5). As its p-value is 0.000, we can accept Eq. (5) with a 5% significance level.

However, three outliers were found as residuals by

Table 1 Stability and accessibility to requirements calculated for Case 1.

#Case	Ver.	Observation subject	Source of requirements	Accessibility $RSA = 1/(NIF)$	Stability $= RMI$	RME_a	
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	*1
		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	*2
		BzNF_Rel+Eff	Gaming equipment, Classes	0.333	1.000	0.545	*3

multiple regression analysis with *R*. They are marked data with "*" in the right most column of Table 1.

- Outliers *1: BzNF_PRc in ver.1
  The requirements of project resource constraints, BzNF_PRc, are requirements of facilities and infrastructures. These requirements were elicited according to the releases of the new version of the system or prototypes. Because of this characteristics of BzNF_RPc, the residual of RME_a became large to be one of the outliers. The RME_a is sensitive to the requirements elicitation in the later stages. If even one requirement is elicited in the last day of the project, the RME_a becomes almost zero. Outliers *1 is the example of such a situation.
- Outliers *2 and *3: BzNF_PRc and BZNF_Rel+Eff in ver.3
  The stability of *2 and *3 was 1.000, while their RMEs were small. Since the data were collected from the actual project, we have to take into account the actual situation in which the RA had to control the requirements elicitation process. For example, if the requirements of an observation subject are stable enough, the RA could decide to elicit the requirements on the basis of need or plan. If the RA decided to elicit such requirements in the later stages of the project, the RME of the observation subject would became smaller. We can accept that the *3 was recorded as the controlled elicitation by the RA.

According to Eq. (5), RME can take a value from −0.21 to 0.69. However, because RME represents the speed of maturation of requirements, its value takes only positive values. Therefore, the value of RME can be greater than 0, while not exceeding 0.69. In order to explain the characteristics of the value of RME, we have taken into account two situations.

1. Situation 1: RSA is 1 and RMI is also 1.
   When the source of requirements is a member of the core stakeholders, the RSA of the requirements is 1. The situation means that the requirements are not modified, added or deleted after the requirements elicitation phase. In this case, the RME of the requirements is 0.69. This value is the maximum value of RME. We consider that the requirements of which RME is 0.69 can mature in the early stages of the project. In other words, the maturation type of the requirements is E-type.
2. Situation 2: RSA is almost 0 and RMI is also close to 0.
   When there is a lot of media between the source of requirements and the RA, the RSA of the requirements becomes close to 0. The situation means that the requirements have been modified, added and/or deleted by the end of the project. In this case, the RME of the requirements is minimum. We consider that the requirements of which RME is 0 cannot mature by the end of the project. In other words, the maturation type of the requirements is L-type.

The RME of requirements of M-type must be larger than that of L-type and less than that of E-type. When we set the thresholds of RME of requirements for E-type, M-type, and L-type, we divided the minimum value of RME, zero, to the maximum value, 0.69, into three potions equally. Thus, the RME of requirements of L-type is greater than zero and not exceeding 0.23, that of M-type is greater than 0.23 and not exceeding 0.46, and the RME of requirements of E-type is greater than 0.46. The thresholds, 0.23 and 0.46, are set within the worse type for projects.

4.2 Case 2: RESORT

The second case is an order management system for restaurants [6].

4.2.1 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.

4.2.2 Observation

The stakeholder map for the project is outlined in Fig. 4. Three companies cooperated in developing the table terminals, the application server, and the observed system. They communicated via a message DB 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 of the restaurant management system on the developers' team, including an RA. The client provided their domain knowledge to the developers to solve this problem. Hence, there were two information flows from the restaurant to the RA via the client.

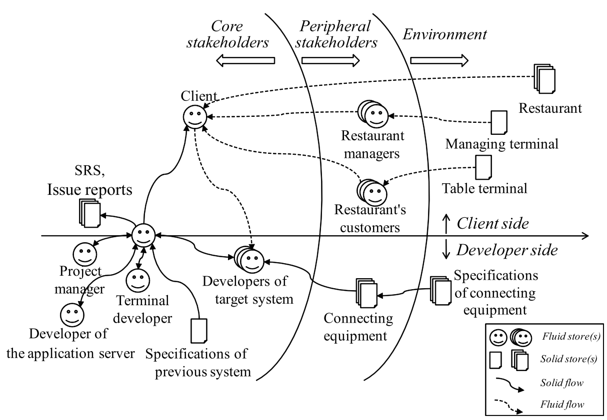

Fig. 4 Stakeholder map of RESORT.

Table 2 summarizes the result of the case. The *RSA* in the table was derived from the number of information flows from the sources of requirements to the RA in Fig. 4, while the RMI was defined base on the interview to the project manager.

According to the interview to the manager, he estimated that the requirements of OESM and OESC would be unstable, since the developers in his team did not have enough knowledge on the application domain. He expected that the size of requirements changes was 20%. Thus, the RMI of these components were 0.8. IS, TTM, and CS were recognized as stable components that communicated with other components developed by the members of the core stakeholders. Furthermore, TTC and MT had human interfaces and communication interfaces. The core team members fixed the specification of the communication interfaces, so the manager took into account the changes in the human interfaces. The manager estimated the size of the requirements changes was less than 5%. Those expected requirements changes depended on the human interfaces. Under the situation, the manager planed to fix these requirements in the design and testing phase with the client's customers: e.g. staffs of a restaurant.

The maturation types, except for TTC and MT, were estimated correctly. According to the interview, the RA left the elicitation of requirements of the exception until later than other requirements elicitation. The action by the RA was allowed by the manager.

The RMI of the TTC and MT were 0.95 that were closed to 1.0 related to that of OESM or OESC. The manager planed to elicit requirements in the later stages of the project, and the requirements would be fixed with their client's customers. It means that the RMI of TTC and MT became 1.0 in the later stages of the project. In a case such as this, if the RA wanted to elicit the requirements, it was possible to do so as soon as possible. Therefore, we can conclude that these exceptions were planned and controlled by the manager, and Eq. (5) cannot be rejected.

4.3 Results and Discussion

In this section we discuss the evaluation of the measurements, as well as the limitation of our method.

Table 2 Stability and accessibility to requirements calculated for Case 2.

Observation subject	Source of requirements	Accessibility $RSA = 1/(NIF)$	Stability RMI	RME	Maturation type Actual (Estimated)
TTC	Client, Restaurant managers, Restaurant customers	0.500	0.95	0.34	L(M)
MT	Client, Restaurant managers	0.500	0.95	0.34	L(M)
IS	Client	1.000	1.0	0.69	E(E)
TTM	Client	1.000	1.0	0.69	E(E)
CS	Developers of application server	1.000	1.0	0.69	E(E)
OESM	Specifications of the external equipment	0.3	0.8	0.20	L(L)
OESC	Specifications of the external equipment	0.3	0.8	0.20	L(L)
DB	Specifications of the previous system	1.0	1.0	0.69	E(E)

- Evaluation of the measurements
  In order to obtain the equation of the RME with two factors: *RSA* and RMI, we applied multiple regression analysis to the first case. 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 *RSA* at an early stage of the project.
  Thus, the problem of how to estimate RMI emerges. There are several ways to estimate RMI. First, we can know the stability of each requirement by referring to a SRS, while it is not clear to measure RMI from the stability defined in the SRS. Secondly, if the manager cannot estimate the RMI of each category, continuously observations of requirements elicitation in the past projects are expected for the estimation. Thirdly, experienced managers are able to estimate the RMI of all requirements categories. In the second case, we applied this method. Though such RMIs are defined by the managers' intuition, in the case 2, it gave us a result that did not contradict to the estimated result. There were two probabilities: one of the possibilities is that both of the intuitively given RMI and Eq. (5) were not accurate, and the other possibility is that both of the results were accurate. Basically, the impact of the *RSA* on the RME is three times bigger than the impact of RMI. Therefore, the accuracy of the RMI for the estimation is less critical than that of *RSA*. We could accept the accuracy of the RMI that the manager of the first case estimate by his intuition. In our future work, we will define the way to derive the RMI from the stability defined in the SRS.
  RME can take a value from 0 to 0.69. If *RSA* is close to zero and RMI is equal to one, RME is 0.02 of which maturation type is estimated to be L-type. In such a case, a project has to improve the low accessibility if the project cannot accept such an L-type maturation. Similarly, if *RSA* is 1 and RMI is almost 0, then RME is 0.46 and its maturation type is estimated to be M-type, not L-type. This can explain the real situation well, since communications between stakeholders and an RA is a general means of eliciting requirements.
- The limitation of RME
  Though the equation of RME was developed by the first case, RME may have another factor, that is, the ability of RA. In the two case studies, we did not take into account the ability of RA, since the problems on the ability of RAs were not reported in the interviews. We will evaluate the equation of RME with the ability of RAs.
  There are limitations within our result. First of all, the equation of RME is applicable to projects in which the RAs have enough skills. We have not analyzed any projects with non-skilled RAs. Secondly, the development organizations had recorded their projects within the minutes of meetings that provided us the processes of their requirements elicitation quantitatively. If there are no records, the projects cannot improve the accuracy of Eq. (5). Thirdly, in order to derive the *RSA* of each subjects in the early stages of the project, we require stakeholder maps and information of stakeholders who are responsible to provide requirements of each subject. If there is no media like the stakeholder map, we cannot derive the *RSA* of each subject.

5. Conclusion

We analyzed the relationship between the RME and two factors: accessibility to the sources of requirements and stability. As a result of multiple regression analysis, RME was able to be estimated and we could get a measurement to derive the maturation type. The limitation of the process provided in this paper was discussed. In our future work, we need to develop the way to derive the RMI from the stability defined in the SRS.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 25330077. We would like to thank Professor Noriko Hanakawa and Mr. Masaki Obana of Hannan University for kindly providing the data on requirements. We would also like to thank Mr. Masao Shimoji and his colleagues who cooperated with our study as interviewees.

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, 1998.
[4] D.M. Berry and E. Kamsties, "Chapter 2: Ambiguity in requirements specification," in Perspectives on Software Requirements, ed. J.C.S.D.P. Leite and J.J.H. Doorn, pp.7–44, Kluwer Academic, Boston, 2004.
[5] N. Nurmuliani, D. Zowghi, and S. Fowell, "Analysis of requirements volatility during software development life cycle," 2004 Australian Software Engineering Conference (ASWEC '04), pp.28–37, 2004.
[6] T. Nakatani, S. Hori, N. Ubayashi, K. Katamine, and M. Hashimoto, "A case study: Requirements elicitation processes throughout a project," 16th International Requirements Engineering Conference (RE '08), pp.241–246, IEEE, 2008.
[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] M. Cataldo and S. Nambiar, "On the relationship between process maturity and geographic distribution: An empirical analysis of their impact on software quality," 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE '09, pp.101–110, 2009.
[9] E. Gottesdiener, Requirements by Collaboration, Pearson Education, 2002.
[10] I. Alexander and S. Robertson, "Understanding project sociology by modelling stakeholders," IEEE Softw., vol.21, no.1, pp.23–27, 2004.

[11] S. Robertson and J. Robertson, Requirements-Led Project Management, Addison-Wesley, 2005.
[12] 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.
[13] A. Loconsole and J. Borstler, "An industrial case study on requirements volatility measures," Asia-Pacific Software Engineering Conference (APSEC '05), pp.249–256, 2005.
[14] S. Fickas and M.S. Feather, "Requirements monitoring in dynamic environments," 3rd IEEE International Conference on Requirements Engineering, pp.140–147, 1995.
[15] A. Sutcliffe, S. Feckas, and M.M. Sohlberg, "Personal and contextual requirements engineering," 13th International Requirements Engineering Conference (RE '05), pp.19–30, 2005.
[16] S. Anderson and M. Felici, "Requirements evolution: From process to product oriented management," Proc. 3rd International Conference on Product Focused Software Process Improvement (Profes 2001), LNCS 2188, pp.27–41, Springer-Verlag, 2001.
[17] S. Anderson and M. Felici, "Controlling requirements evolution: An avionics case study," SAFECOMP 2000 (LNCS 1943), pp.361–370, Springer-Verlag, 2000.
[18] S. Anderson and M. Felici, "Quantitative aspects of requirements evolution," Proc. 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment (COMPSAC '02), pp.27–32, 2002.
[19] T. Nakatani, N. Kondo, M. Inoki, M. Tsuda, and K. Katamine, "The taxonomy of requirements for conducting elicitation," Information Systems 2011 (IS2011), pp.161–168, 2011.
[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," 4th International Conference on Software and Data Technologies (ICSOFT 2009), INSTICC, pp.145–150, 2009.
[21] T. Nakatani, N. Kondo, J. Shirogane, H. Kaiya, S. Hori, and K. Katamine, "Toward the decision tree for inferring requirements maturation types," IEICE Trans. Inf. & Syst., vol.E95-D, no.4, pp.1021–1030, April 2012.
[22] M.R. Strens and R.C. Sugden, "Change analysis: A step towards meeting the challenge of changing requirements," IEEE Symposium and Workshop on Engineering of Computer Based Systems (ECBS '96), pp.278–183, 1996.
[23] A.M. Davis, Just Enough Requirements Management: Where Software Development Meets Marketing, Dorset House, 2005.
[24] M. Shaw and D. Garlan, Software Architecture, Prentice-Hall, 1996.
[25] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, Object-Oriented Software Engineering, Addison-Wesley, 1992.
[26] ISO/IEC 25030:2007, "Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements," 2007.
[27] IEEE Std 982.1, "IEEE standard dictionary of measures to produce reliable software," 1988.
[28] N. Hanakawa and M. Obana, "Mobile game terminal based interactive education environment for large-scale lectures," Eighth IASTED International Conference on Web-based Education (WBE2010), March 2010.

Takako Nakatani received the Ph.D. degree from the University of Tokyo in 1998. She has been a board member of S-Lagoon since 1995, and an associate professor at the University of Tsukuba since 2006. Her interest area is requirements engineering. She is a member of the IEICE ISS, IPSJ, IEEE CS, ACM, and the Society of Project Management.

Shozo Hori received the Ph.D. degree in engineering from Kyushu Institute of Technology, Japan in 2012. He has performed embedded software development and management for 20 years. He has been engaged in process improvement of organizations using ISO9001 or CMM since 1998. His research interests include software engineering, requirements engineering, and project management.

Keiichi Katamine received the Ph.D. degree in engineering from Kyushu Institute of Technology, Japan. He is now an associate professor of computer science and systems engineering at Kyushu Institute of Technology. He is a SEI authorized PSP instructor, and a Goldratt School certified CCPM and S-DBR trainer. He is a member of the IEICE, IPSJ, and the Society of Project Management.

Michio Tsuda received the B.E. degree in electrical engineering from Doshisha University in 1970 and the D.E. degree in information and computer sciences from Osaka University in 2008. He was a systems engineer in the industrial field and a developer for software engineering, Hitachi, Ltd. and Hitachi Solutons, Ltd. from 1970 to 2008.

Toshihiko Tsumaki received the Ph.D. degree from the University of Tokyo in 2007. His past affiliations are Nihon Unisys Ltd. and Shibaura Institute of Technology. Current affiliations are National Institute of Informatics, Hosei University and Nippon Institute of Technology. His main research area is requirements engineering. Now his research interests include diversity, justice as well as autonomy. He is a member of the Information Processing Society of Japan.