PAPER *Special Section on Knowledge-Based Software Engineering*

Toward the Decision Tree for Inferring Requirements Maturation Types

Takako NAKATANI[†a)], ***Member***, **Narihito KONDO**[††], ***Nonmember***, **Junko SHIROGANE**[†††], ***Member***, **Haruhiko KAIYA**[††††], **Shozo HORI**[†††††,††††††], ***Nonmembers***, ***and*** **Keiichi KATAMINE**[†††††], ***Member***

SUMMARY Requirements are elicited step by step during the requirements engineering (RE) process. However, some types of requirements are elicited completely after the scheduled requirements elicitation process is finished. Such a situation is regarded as problematic situation. In our study, the difficulties of eliciting various kinds of requirements is observed by components. We refer to the components as observation targets (OTs) and introduce the word "Requirements maturation." It means when and how requirements are elicited completely in the project. The requirements maturation is discussed on physical and logical OTs. OTs Viewed from a logical viewpoint are called logical OTs, e.g. quality requirements. The requirements of physical OTs, e.g., modules, components, subsystems, etc., includes functional and non-functional requirements. They are influenced by their requesters' environmental changes, as well as developers' technical changes. In order to infer the requirements maturation period of each OT, we need to know how much these factors influence the OTs' requirements maturation. According to the observation of actual past projects, we defined the PRINCE (Pre Requirements Intelligence Net Consideration and Evaluation) model. It aims to guide developers in their observation of the requirements maturation of OTs. We quantitatively analyzed the actual cases with their requirements elicitation process and extracted essential factors that influence the requirements maturation. The results of interviews of project managers are analyzed by WEKA, a data mining system, from which the decision tree was derived. This paper introduces the PRINCE model and the category of logical OTs to be observed. The decision tree that helps developers infer the maturation type of an OT is also described. We evaluate the tree through real projects and discuss its ability to infer the requirements maturation types.

key words: *requirements elicitation, requirements process, requirements changes, requirements maturation, project management*

1. Introduction

This paper focuses on the requirements elicitation process during the development of software.

Inaccurate, incomplete, or vague requirements are the risks to a project [1]. There was popular research done on requirements correction [2] and a risk management method for requirements changes [3]. Software requirements should be fixed not only by correcting errors in the requirements phase, but also through requirements elicitation in the other phases [4]. For example, during the development, the developers have meetings with their customers. In the meetings, the developers can deepen their understanding of the requirements [5] and their customers may also deepen their own understanding of their requirements. As a result, inaccurate and/or incomplete requirements are added and modified after the early phase of the development.

Requirements volatility can also present risks to a project [6], [7]. There are many causes of requirements volatility. Vague requirements are volatile. Ambiguity in software requirements specifications (SRSs) sometimes mandates requirements elicitation during the software development. Unambiguity stated in IEEE Std. 830-1998 [8] is one of the recommended qualities of SRSs. Since most SRSs are written in natural language [9], it is difficult to avoid changes for the sake of completion and/or correction of ambiguity. External and internal environmental changes can cause requirements volatility. Requirements analysts (RAs) need to manage such requirements volatility.

Our way of coping with the risks inherent in requirements volatility is to provide a method for managing their elicitation process over the entirety of the development of software. To prevent project failures due to requirements volatility, we must clearly understand the actual processes of requirements elicitation. The PRINCE (Pre Requirements Intelligence Net Consideration and Evaluation) project was undertaken to learn the history of requirements elicitation from past projects and forecast the requirements elicitation process in future projects. In this context, we introduce the word "Requirements maturation." It means when and how requirements in a project are elicited completely. The name "*PRINCE*" comes from "Il Principe" by N. Machiavelli [10]. He told us that we should learn the past history and prepare for the future problems and/or troubles. We can adopt and apply his concept to requirements engineering. Thus, learning the requirements maturation process from past projects and knowing what causes late and/or early maturation is expected to help us prepare the requirements elicitation, and plan the development process.

The PRINCE model was developed by observing requirements elicitation processes of actual past projects. In our study, the maturation is observed by components. We

Manuscript received July 1, 2011.
Manuscript revised October 31, 2011.

[†]The author is with University of Tsukuba, Tokyo, 112–0012 Japan.
[††]The author is with Nagoya Management Junior College, Owariasahi-shi, 488–8711 Japan.
[†††]The author is with School of Arts and Sciences, Tokyo Woman's Christian University, Tokyo, 167–8585 Japan.
[††††]The author is with Shinshu University, Matsumoto-shi, 390–0802 Japan.
[†††††]The authors are with Kyushu Institute of Technology, Iizuka-shi, 820–8502 Japan.
[††††††]The author is with Yaskawa Information Systems Corporation, Kawasaki-shi, 215–0004 Japan.
a) E-mail: nakatani@gssm.otsuka.tsukuba.ac.jp

DOI: 10.1587/transinf.E95.D.1021

refer to the components as observation targets (OTs). Thus, RAs need to identify OTs in their project and understand the requirements elicitation process of each OT. The purpose of this paper is to answer the following questions.

- What requirements can be grouped into OTs?
  An OT is identified as the category of a component, in order to observe its problems with regard to its requirements elicitation. When the categories are provided, the RAs can identify concrete OTs in their projects.
- What determines the requirements maturation type?
  The requirements are influenced by requesters' environmental changes, as well as developers' technical changes. In order to infer the requirements maturation period of each OT, we need to know what kinds of factors influence the OTs' requirements maturation. A decision tree can provide a systematic analysis with those factors to determine the maturation type of the OT.

The OT needs a key to be distinguished from others in the project and to be compared with the same OTs of other projects. The categories should be applicable to embedded systems, information systems, and command and control systems. We define the categories of OTs in order to identify the OTs in every project.

The requirements of every OT have a proper time to be elicited. A decision tree is developed in order to help RAs infer the requirements maturation type of OT. The result of our case studies provides us quantitative requirements elicitation processes. After we interviewed the project managers, we extracted essential factors that influence the requirements maturation. The results of the interviews are analyzed by WEKA, a data mining system, from which the decision tree was derived.

This paper's structure is organized as follows. In the next section, we introduce the related work. Section 3 gives an overview of the PRINCE model. Section 4 defines the category of OT. Section 5 presents a decision tree derived from the data of three projects with WEKA: a data mining tool [11], [12]. In the last section, we discuss the results of our research and conclude the paper by outlining future work.

2. Related Work

Requirements traceability is important for managing requirements changes. Arkley and his colleagues [13] described an application of traceability. They classified the project requirements into the following categories.

- New requirements
- Unchanged requirements
- Requirements which required a minor modification
- Requirements with unresolved issues

In order to plan and manage the requirements process, we need categories of OTs for monitoring their requirements elicitation process. In this paper, "requirements process" denotes requirements elicitation, observation, and managing processes. Arkley's categories did not satisfy our goal.

Sankar and Venkat focused on a way to control requirements. They showed the percentage of requirements frozen in the development process. According to their article, 70% of all requirements were frozen during requirements gathering [14]. If most requirements could be frozen in the early development phase, we would be happy. As in an agile or unified process, RAs require their customers to participate in their projects [15]. Further, RAs should prepare to accept requirements changes during ant given project and, release the system on time. In our study, we assume that a project manager can manage his/her project with requirements changes if the RA can infer the requirements maturation period. Therefore, the RAs need a method to infer the requirements maturation period. We developed a decision tree to infer the time when the requirements are matured.

There is research based on observing requirements stability: Bush and Finkelstein introduced a method for observing initial requirements stability by the stability of goals [16]. In their goal model, goal changes are propagated to requirements changes. However, requirements without goals are not the only requirements that will be changed. As we mentioned in the previous section, there are various reasons that cause requirements changes. When we regard these changes as the products of requirements elicitation, we can discuss them in the context of the requirements elicitation process. If an RA accepts requirements changes after the early stage of a project, he/she needs to estimate the period of the completion of requirements elicitation rather than the requirements elicitation methods introduced in [17]–[20].

Nurmuliani et al. [21] proposed the measure of requirements volatility with respect to the total number of requirements and the total number of changes. There are other metrics for requirements changes. Requirements stability [22] is defined by the number of initial requirements divided by the total number of requirements. It does not take into account historical information about changes. Requirements Maturity Index (RMI) is also a metric of requirements stability. RMI is defined by Anderson et al. [23] with the second implementation of the Software Maturity Index (SMI) [24]. In their implementation, the change and addition of requirements is tracked. In our study, however, we do not focus the traceability of each requirement, but trace the outcomes of requirements elicitation on each OT. In order to observe the changing history of requirements, we represent the ratio of requirements maturation as, "at a time t".

Nakatani et al. defined $RMR(t)$: the requirements maturation ratio at time t [25]. $RMR(t)$ represents how much the requirements of an OT have matured at a specific period t. It is represented by the following expression.

$$RMR(t) = \frac{R_T(t)}{R_E} * 100 \tag{1}$$

In this expression, R_E is the total number of requirements of the OT at the end of a project, and $R_T(t)$ stands for the total number of requirements at the time t. When the project is started, the RMR_0 of every OT is equal to 0%, and when the project is completed, their RMR_E is equal to 100%. Thus, RMR can represent historical information about requirements elicitation activities on OTs, though it cannot be used before the end of the project. To solve this problem, we introduce the decision tree in order to infer the requirements maturation.

3. The PRINCE Model

Observations of actual requirements elicitation processes revealed that each requirement had its own reason to be elicited within a certain stage (i.e. early, middle, or later) of the development [25]. The OTs can be categorized into the following four maturation types on requirements:–as E-type, M-type, L-type, and U-type [26]. Note that in the PRINCE model described below, the requirements are presented in terms of OTs.

The E-type reaches maturation at the early stage of the project. The requirements of an E-typed OT are completely elicited in the early stage of the project and are stable enough. This is the ideal maturation type of OTs for every project. The second type, named an M-type, reaches maturity in the middle stage of the project. The requirements of an M-typed OT cannot be completely elicited until the middle stage of the project starts, but they may mature before the later stage of the project. The third maturation type, named an L-type, cannot be matured until the later stage of the project. The last maturation type, named a U-type, refers to unexpected elicitations at any stage of the project.

Figure 1 shows the PRINCE model with the elicitation process of each type [26]. The x-axis represents the duration of the project, and the y-axis represents RMR. Each curve shows a typical maturation process of requirements. For example, E-type requirements are elicited completely before the internal design phase starts and are never changed after this phase.

We provide the PRINCE model for project managers and RAs as a basic guide for planning and managing the requirements process. This model implies that requirements should be elicited not in the early stage of the project, but the scheduled stage.

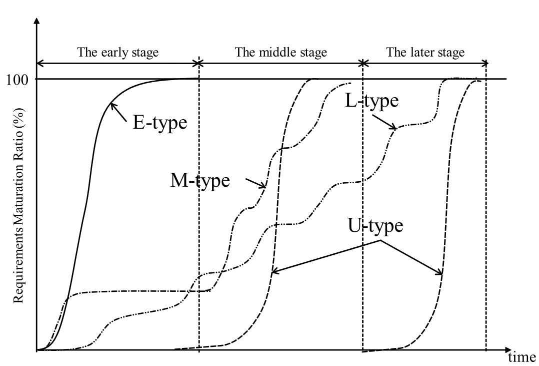

Fig. 1 The PRINCE model: a guide for requirements elicitation.

First of all, when an RA applies the PRINCE model to his/her project, he/she is expected to identify OTs and plan the requirements elicitation process for each OT. However, the model has two problems. The model does not provide RAs with the basic ideas for identifying each OT. After he/she identifies the OT, he/she has to map the maturation type to each OT and schedule the meetings for requirements elicitation with their stakeholders according to the maturation type. Thus, the PRINCE model cannot be applied any projects in which the stakeholders do not participate the requirements elicitation process. Even though the stakeholders participate the requirements elicitation process well, the second problem of the model arises. There are no rationales to determine the maturation type of each OT. In order to solve these two problems, we provide OT categories to monitor their requirements maturation and a decision tree to infer the maturation type of each OT. In the next section, we provide OT categories.

4. Category of Observation Targets

4.1 Concept and the Structure

The categories of OTs are defined in order to observe the process of requirements elicitation.

There are physical and logical viewpoints to identify the OTs. Systems, including embedded systems, information systems, and their subsystems are identifiable as physical OTs from the physical viewpoint. Every requirement of each physical OT can be mapped into multiple logical categories, because a requirement has multiple aspects. For example, when a requirement of a physical OT named "operation on books" says "*retrieve* information of a book *from the database within three seconds*," the requirement belongs to functional, design constraints, and performance OTs. Then, RAs can observe and count the number of requirements in each logical OT, as well as in each physical OT.

We define logical categories based on known characteristics of physical OTs, which are organized in three levels. The first level is a purpose level. If the purpose of a physical OT depends on the external business environment: e.g. market situation, competitors' products, etc., the changes in those requirements cannot be negotiable. It implies that there is a possibility of the OT to be the U-type maturation. On the other hand, when the purpose of an OT depends on the internal business organization of the project, the requirements on the purpose can be the subjects of triage within the organization, thus it is possible to avoid the OT falling under the U-type maturation. The OT of the first level helps RAs ascertain the possibility of a U-type on the OT. In the second level, requirements are viewed from basic categories of requirements engineering, i.e. functional and non-functional. All requirements can be categorized into functional and non-functional requirements. The third level of the categories is the detailed level under the second level.

Figure 2 presents the name and abbreviation of logical OTs. In the first level, OTs are put into two categories

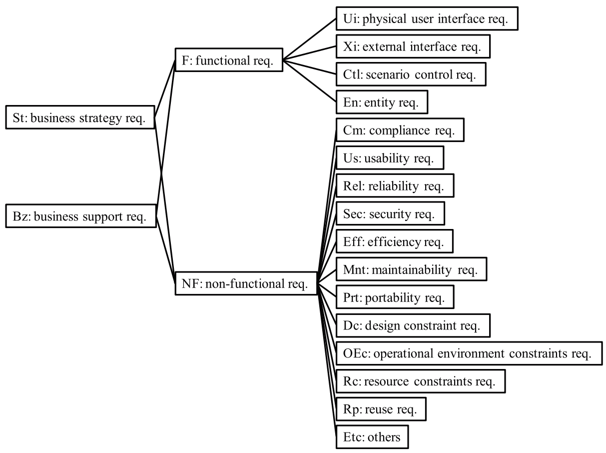

Fig. 2 Categories of observation targets.

according to their purpose: a strategic purpose (St) or a business support purpose (Bz).

- St: The requirements in this category are affected by the external business organization of the project.
  Business strategic requirements are defined to win the business and should be evolved to adapt to the external environment. Such a requirement arises from external environmental changes at any time.
- Bz: The requirements in this category are given from the internal business organization of the project.
  This category relates to supporting objectives. For example, OTs for improving business productivity belong to this category.

The requirements in Bz must be controllable and negotiable by the project constraints. This is one of the differences between St and Bz and the difference impacts the requirements volatility.

The second level consists of functional and non-functional requirements. Functional requirements are implemented in modules, while some non-functional requirements are implemented within the software architecture. Hence, focusing on these kinds of requirements and their maturation is inevitable in order to manage the time and reasons of requirements elicitation in avoiding re-development caused by the late elicitation of these requirements.

The third level is the detailed level. The functional requirements consist of components by Jacobson: i.e. boundary, control, and entity components [27]. There are two categories in boundary components according to the connected objects: external components and people. Determining the requirements of the external interface entails cooperation with vendor companies that provide the external components. End-user cooperation is also indispensable for setting the requirements of the user interfaces. In general, the interface components are more fragile than the entity components. Thus, focusing on these OTs is important for observation of requirements maturation. The non-functional requirements are composed of quality-related characteristics defined in ISO/IEC 25030 [28]. The requirements elicitation processes of these categories are candidates to be managed.

Table 1 Category of OT and its source examples.

OT	Sources: Developer's organization	Sources: Requester's organization	Sources: External organization
St		business goals	market, competitors
Bz		user's organization, operators	
Ui	feasibility	end users	market, competitors
Xi		vendor company	
En		solution domain	
Ctl		end users	market, competitors
Cm, Rel, Sec, Eff	technical limitations	budget constraints, social responsibility	agreement, rules, laws, standard
Mnt, Prt	productivity constraints	budget constraints, competitors	
Dc, OEc	feasibility, technical trend	budget constraints	
Rc, Rp	technical limitations	budget constraints	
Etc			

When we refer to the categories, we combine the names which represent the OT's characteristics. For example, BzNF_Dc is related to business support by the non-functional requirements of design constraints.

The PRINCE project developed a guideline for observing requirements elicitation history. The guideline will be introduced shortly.

It is possible to know the source of requirements in the early stage of a project. "Stakeholder" is an example of a source of requirements. The project manager and RAs of any project are responsible to identify the sources of OTs in the early stage of a project. The volatility of OT is influenced by the source of requirements [29].

Table 1 presents the examples of the sources of requirements in the categories. The table will help the project managers and RAs determine the source of OT. For example, if an OT is related to various sources, the requirements can be in conflict with each other, and these conflicts should be resolved through negotiations among the sources. We consider that the maturation type of such OTs may become the M-type or L-type, since they need time to negotiation. If an OT is related to highly performed services, its efficiency requirements should be important.

The source of categories is not enough to determine the maturation type of an OT. The decision tree was developed in order to help RAs infer the maturation type of an OT.

4.2 Guideline for the Observation

In this section, we briefly introduce the guideline that was developed to observe the requirements process of the completed projects. The observation conventions are as follows.

- Regard the requirements specified in the first version of the requirements specification as the baseline requirements.

- Do not distinguish added, modified and deleted requirements. All of them are treated as elicited requirements throughout the project.
- If it is possible, ignore requirements for the future version of the software.

All the requirements are categorized into OTs by the observation process in six steps.

1. Gather materials: requirements specifications, use cases, display images, output sheets, records with added, modified, and deleted requirements, change management reports, issue reports, error/failure reports, review reports, minutes of meetings or e-mails, Q&A reports, etc.
2. If it is possible, define physical OTs.
3. Identify requirements statements.
4. Map the requirements to the physical OTs and the logical OTs provided in the OT categories. If a requirement belongs to multiple OTs, add one to all corresponding OTs.
5. Interview the manager and customers of the project and clarify the situation, causes, and background of the requirements elicitation process.
6. Analyze each OT's maturation period with *RMR* and the result of interviews for future projects.

The detailed guideline is in [30].

Anderson and Felici showed that there was a linear relationship between the number of changes occurring in a requirements specification and its size [31]. It means that small OTs tend to receive a small number of requirements. According to our previous study [32], the RA needs to collect a certain amount of requirements. "A certain amount" means that if the number of requirements in an OT is one or two, RAs cannot evaluate the maturation period of the OT adequately. Thus, use cases that manipulate the same object are recommended to be categorized into the same OT. Such an OT comes to have enough granularity for the purpose of our observation of requirements maturation.

4.3 Evaluation of the Categories and Guidelines

We evaluated the categories and guidelines as to whether engineers can identify the categories or not. The evaluators were the engineers of three companies, two university students and a teacher. After we explained the categories and guidelines to them, the evaluators applied the categories to the same project and observed the requirements elicitation process for evaluating the applicability of the guidelines. The specifications were written by professional engineers, and all the requirements changing processes were recorded in the version controlled specifications. The various results of the evaluation did not contradict each other. As a result, we concluded that the categories and guidelines are applicable to identify OTs. When engineers apply the guidelines and categories of OTs, they need to understand what is Bz and St are, as well the basic requirements engineering terms, i.e. functional and non-functional requirements, and software qualities in ISO/IEC 25030.

The OTs identified from the physical view need the architectural design specification. In our future work, our evaluation process will be supported by the work of Kaiya et al., who developed a tool for analyzing requirements quality based on the requirements statements with a term-characteristic map [33].

In real world application, RAs may select categories for their own observation. Inoki et al. selected OTs on efficiency [34] to support engineers. In their situation, requirements on efficiency are highly prioritized, rather than improving usability or other qualities. Fujiwara [35] added a sub-strategic category to the first level of the categories. Requirements in the sub-strategic category are not directly influenced by the external environment, but indirectly influenced by the external environment. By adding the new category, they were able to improve the productivity of identifying the OTs of requirements.

5. Decision tree for Maturation Types

5.1 Maturation Factors

All the requirements do not need to be elicited in the early stage of the project. The requirements are influenced by requesters' environmental changes, as well as developers' technical changes. In order to infer the requirements maturation period of each OT, we need to know what kinds of factors influence the OTs' requirements maturation. A decision tree can provide a systematic analysis with those factors to determine the maturation type of the OT. Maturation types: i.e. E-type, M-type, or L-type are the indicators of the maturation period. With regard to developing the decision tree, we define maturation factors that may influence the requirements elicitation process and can be recognized in the early stage of projects. The sources of OTs and importance of qualities are the candidates of the factors. We assume that the factors determine the requirements maturation types. This means that, if the requirements of an OT are difficult to elicit, the maturation type of the OT belongs to the L-type.

The following maturation factors are defined as a result of interviews with the project managers. We developed a decision tree based on the data of three projects. Two projects were concluded in less than six months, while the other project was managed through incremental development for three times over 60 weeks. The OTs identified in these projects were affected by the characteristics of a problem domain, technical environment, market environment, project environment, engineers' domain knowledge, engineering knowledge, requesters' cooperation, as well as product environment. Here are the factors. Two or three choices in parenthesis were provided for the project managers as their answers.

- Characteristics of the problem domain:
  - The problem domain stability: (high, medium, low).
  - The related laws' stability: (high, medium, low).
  - User diversity: (high, medium, low).
  - Market dependent: (yes, no).
- Characteristics of the technical environment:
  - Technically dependent: (yes, no).
  - Existence of external interfaces provided by the third party vendors: (high, medium, low).
  - Existence of other OTs: (many, medium, less).
  - Importance of the OT's quality requirements:
    * Functionality importance: (high, medium, low).
    * Usability importance: (high, medium, low).
    * Security importance: (high, medium, low).
    * Efficiency importance: (high, medium, low).
    * Other nonfunctional requirements importance: (high, medium, low).
  - Design constraints importance: (high, medium, low).
- Characteristics of the project:
  - Duration of the project: (< 6 months, < 1 year, > 1 year).
  - Person in charge of defining the requirements: (sponsor, user, developer, third party organization).
  - Knowledge level and/or experience level of the developers: (enough, medium, poor).
  - Domain experts' contributions or the knowledge level of requesters: (enough, medium, poor).
  - Requesters cooperation: (good, medium, bad).

Project managers and RAs can identify the characteristics of the sources. The source of an OT can be determined in the early stage of the project. Some of these factors were not selected by the project managers and some of them were ignored by WEKA, the data mining tool, that we applied to derive the decision tree.

5.2 Data Mining

In order to derive the decision tree for the purpose of determining maturation types, we observed OTs' maturation processes in three projects. Each OT's maturation type is calculated from, and based on the records of those projects.

The maturation types do not depend on the development process, but the stage: early, middle, and later. Thus, the maturation type of each OT can be found by the following equation.

$$\text{Maturation Period: } M_p = \frac{Tr_{comp} - Tp_{st}}{Tp_{comp} - Tp_{st}} \tag{2}$$

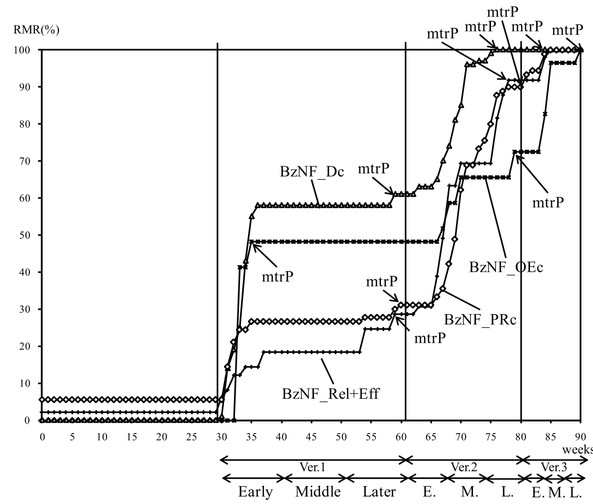

Fig. 3 History of the requirements maturation ratio of BzNF.

In this formula, Tr_{comp} represents the time when the requirements have been elicited completely, Tp_{st} represents the time when the requirements phase has started, and Tp_{comp} represents the time when the development of the OT has been completed. We got these parameters from the completed project. The decision tree will be applied to infer the maturation types of OTs in future projects. The adequacy of the decision tree can be evaluated by comparing the inferred maturation type and the actual maturation type of an OT. The actual maturation types were observed in three other projects. In total, we observed six projects. Three projects were used in deriving the decision tree, and the other three projects were used to evaluate the tree.

Figure 3 presents the historical data on the maturation of a case. The vertical lines represent the start of incremental developments. The "mtrP" is marked as the requirements maturation point of each OT in each version. As the guide of maturation types, three maturation periods are shown at the bottom of each figure.

For simplicity, if M_p is less than 1/3, 2/3, and 1, we interpret the value of M_p to the E-Type, M-Type, and L-Type respectively. We asked the project manager of each system to set values as the characteristics of each of the OTs. The values are determined by the manager's intuition. This is a limitation within our research: however, our research can show the possibility of inferring the requirements maturation types.

These records, along with the values of the attributes, are put into WEKA [11], [12]. WEKA is a data mining system for performing predictions and forecasting through the utilization of data. Its J48 is a machine learning algorithm and derives a decision tree from input data. We expect the decision tree to lead us to what determines the requirements maturation type of each OT.

5.3 Decision Tree

The 69 data that we put into WEKA are composed of the

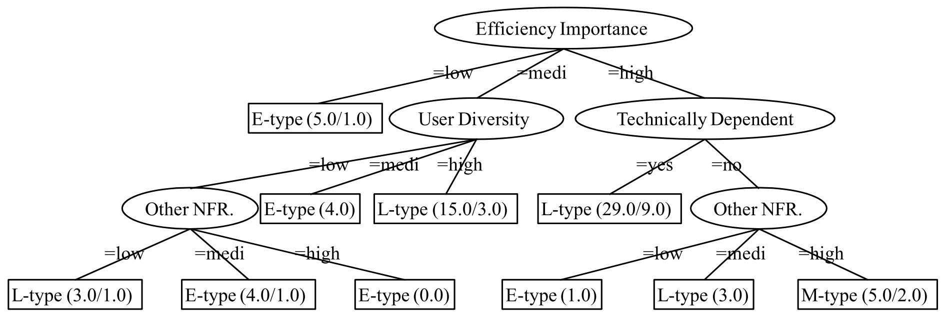

Fig. 4 Decision tree for maturation types.

characteristics of OTs with their maturation type calculated by utilizing the actual data. We input the data to WEKA and applied it several times by deleting decision nodes that do not appear in the derived decision trees. This trial and error process improved the reliability of the decision tree.

Figure 4 shows the results derived from WEKA. In the tree, a rectangular leaf represents the class of the decision, while an oval node represents a decision making attribute. Each arc has an attribute value. The numbers "(x/y)" in rectangle represent that x data has come to a decision, and that the y data does not fit the decision. For example, the left bottom leaf labeled "L-type (3.0/1.0)" means that three data have come to the leaf and one data does not fit L-type. If all data that reach to a leaf fit to the label of the leaf, only x is shown in the parentheses. The tree tells us the tendency of the maturation type of an OT. In the tree, "other NFR." represents a characteristic of the importance of non-functional requirements without regard to efficiency, usability, security, and constraints requirements.

We can interpret the tree as follows as examples:

- If "efficiency importance" of an OT is low, its maturation type becomes the E-Type.
- If "efficiency importance" of an OT is high, and the OT is a "technically dependent component", then its maturation type becomes the L-Type.

In general, requesters' cooperation is inevitable for requirements elicitation. In Fig. 4, the node "requesters' cooperation" does not exist, because most data is provided by projects proceeded under the good requesters' cooperation.

5.4 Quantitative Evaluation

We evaluate the decision tree by comparing the inferred type with the actual maturation type of the OTs. For the evaluation, we selected three actual projects for the evaluation completed in less than six months. Their project duration was similar to the three projects that provided 69 OTs for developing the decision tree. There were eight OTs in each project. We compared types of each OT with the maturation types inferred by the decision tree. If the actual type belongs to the earlier maturation type as opposed to the inferred type, we consider the decision tree to be effective. Because the type inferred by the tree represents the tendency of the OT, and, if a project manager wanted to get the requirements of an OT earlier and manage the elicitation process, the maturation type can be the earlier type than the inferred type.

Table 2 Evaluation of the decision tree.

OT Id		Maturation Type		
Project	Physical OT	Actual		Inferred
A	1	L	=	L
	2	L	=	L
	3	M	<	L
	4	L	=	L
	5	L	≫	E
	6	M	<	L
	7	L	=	L
	8	E	=	E
B	1	E	≪	L
	2	E	≪	L
	3	E	=	E
	4	E	≪	L
	5	E	<	L
	6	E	≪	L
	7	L	=	L
	8	E	=	E
C	1	E	≪	L
	2	E	=	E
	3	M	<	L
	4	M	<	L
	5	E	=	E
	6	E	≪	L
	7	E	=	E
	8	E	=	E

Table 2 shows the results of the evaluation. According to the table, 12 of 24 OTs matched the inferred types and 11 of 24 OTs matured earlier type as opposed to the inferred types. The satisfaction ratio was 95.8%. Most project managers manage the requirements elicitation process to freeze the requirements as soon as possible. If we evaluate these projects according to the decision tree, we can say that the managers could manage their requirements elicitation process well, since most maturation types of OTs were less or equal to the inferred types.

As shown in the table, all OTs of project B and C were satisfied, and the OTs of project A were almost satisfied. The OTs of project A are not categorized into the proper maturation types by the decision tree. Six of the OTs were categorized into L-Type. We contemplate that the developers' lack of knowledge of the domain may affect the results, while in other research examples, most developers had mediocre knowledge or, enough knowledge of the domain.

Project A had one exception of which maturation was delayed by the inferred period. The OT's usability was highly important. The usability node is not included in the decision tree.

In our future work, we will apply other projects' data in order to improve the accuracy of the decision tree. For example, that a developers' lack of knowledge of the domain may affect the results in some cases. The requesters' co-operation must affect the maturation type. We must analyze the priority of non-functional requirements that may lead the maturation type to L-Type. Furthermore, the decision tree needs to consider the effect of a project manager's efforts to elicit requirements in the earlier stage of their projects, as opposed to the inferred period.

5.5 Qualitative Evaluation

We interviewed project managers to qualitatively evaluate the decision tree.

A manager who is involved in telecommunication projects, who saw the tree and understood the scenario of the inference, had this to say: "If the efficiency requirements on OT are highly important and, have technical dependency with other components, its OT maturation type will be the L-type", and further, he said that the efficiency requirements should be elicited in the earliest stage of any project. The tree, however, does not imply that the OT should be the L-type, but that if the manager did not care about that, it might become the L-type. Therefore, the tree does not contradict his intention. Basically, if the efficiency requirements of an OT are important, the OT's design is usually impacted by technical and environmental changes, including hardware upgrades. In such cases, the requirements maturation type of the OT becomes the L-type, and it sometimes causes re-work by developers. In order to avoid these problematic situations, we need to estimate the maturation type of OTs in the early stage of any project. If we can know, that the maturation type of an OT may become the L-type, we can initiate and control the elicitation process.

Another manager who gave us comments focused on the user diversity. If there are various users, then their requirements need a lot of negotiation. He agreed that the leaf "L-type" under the "user diversity is high." He also mentioned that "usability importance" should appear in the tree.

The tree leads us to conclude that, if every factor is not important, then, that the maturation type of an OT will become the E-type. This is because we defined the maturation factors that negatively influence the maturation period as, "yes" or "high", which implies that the OT may become an M-type or L-type.

6. Discussion and Conclusion

6.1 The U-Type Maturation

In this paper, we did not derive the decision tree based on the U-Type maturation. Before deriving the decision tree for the U-Type maturation, we must rethink the U-Type. This type of maturation imposes a potential highly negative risk upon the project, since it is out of control and we cannot predict at which stage of the project it might happen. Therefore, it is important to identify the possibility, thus causing the U-Type maturation before it emerges. If we can know the potential possibility of this type, we can prepare for the worst. For example, we can develop the *E-Type* OT earlier in the plan; then, we can spare the resources for dealing with the later U-Type maturation. However, when we observe the requirements elicitation data, it is hard to distinguish the U-Type maturation from the other types.

In order to identify OTs that may become the U-Type, we asked the stakeholders and developers to point out any U-Type OTs within their projects. We got six U-Type OTs within six studied projects. The following situations caused the U-Type maturation.

- Under the influence of changing requirements of the external interface.
  We regard these OTs as the L-Type maturation quantitatively. Two of the studied projects had the mission to provide market-driven products. In their situation, they had to take into account their integration with multiple products provided by the third party vendors.
  One project could not acquire the correct information from the third party vendors. As a result, during the integration testing phase, the engineers had to change the requirements of the external interfaces in order to match their product to the correct interfaces. In future projects, project managers may be able to cope with U-type OTs that depend on external interfaces, if they know that such OTs tend to be U-type or L-type. The PRINCE model is expected to provide patterns to prevent U-type for project managers and RAs.
  In said project, the system was planned to support a single graphics format. Further, in the system testing phase, they decided to support other formats. This added requirement came from the market environment. It told us that OTs with the St characteristic are important OTs to be observed.
- The requirements had come from deep inside the connected component.
  The system had been developed with a component. The system had several layers, with the lower layer subcomponent having had its version updated in the middle stage of the project. No one expected such an update, but it had occurred regardless. Usually, layered architecture is free from internal interface changes, even though the inside components have changed. However, when such an update unexpectedly provides desired functions, then these specific requirements should be updated. This U-Type maturation was caused by the unusual design of the connected component. Project managers have to be prepared for a U-type case maturation of an OT that, provides desired functions in a lower layer.
- The requesters realized their goals after trying to use

the system in a real situation.
The system was developed as an educational support system in a university. In the early stage of the project, the requesters required higher usability for the recording of student attendance. As a result, the requesters had focused on the system interface too much and neglected an important function that supported the CSV (Comma Separated Values) format for the outputs.

According to the situation that caused the U-Type maturation, when an OT had an interface which was connected to the component provided by a third party vendor, the requirements of the OT became the U-Type maturation. The last example tells us that the early prototyping is still effective in saving a project from the U-Type maturation, as famously mentioned by Boehm [36]. We discussed the U-type maturation as part of the research of the project management patterns based on the PRINCE model [37].

6.2 Implication of the Decision Tree

When we collected the data to derive the decision tree, we were faced with the challenge to set a value for each element that may affect the OT's maturation type. An OT is a target to observe the requirements elicitation process. The decision tree is expected to provide us characteristics of an OT to determine its requirements maturation type. This paper answers the following research questions.

- What requirements can be grouped into OTs?
  We defined the categories of OTs with the guideline for the observation of OTs' maturation. The cases were observed according to the categories.
- What determines the requirements maturation type?
  We explored the characteristics that determine the OT's requirements maturation type. The decision tree is derived from and provides the requirements maturation factors and their impact on the maturation types.

There remain several issues. The values of maturation factors of each OT applied to WEKA were subjectively provided by project managers. The value of the factors should be derived from a more-grounded source of the OT. There must be a dependency between the characteristics of the sources and the value of the factors of the OT. We will explore the dependency in our future work. Then, we will be able to develop a guide to establish a value for each maturation factor of OTs. This paper shows the possibility to infer requirements maturation types by focusing on the OT maturation factors.

Acknowledgment

This work has been supported by the Joint Forum for Strategic Software Research (SSR) since 2007. The authors would like to thank all the members and cooperators of this project. Especially, the authors wish to thank Prof. Noriko Hanakawa and Mr. Masaki Obana of Hannan University for kindly providing the requirements data. The evaluation of the guideline was supported by Prof. Katsuro Inoue of Osaka university.

References

[1] T.R. Leishman and D.A. Cook, "Requirements risks can drown software projects," J. Defense Software Engineering, pp.1–8, April 2002.
[2] B. Boehm, Software Engineering Economics, Prentice Hall, 1981.
[3] B. Boehm, P. Bose, E. Horowitz, and M.J. Lee, "Software requirements negotiation and renegotiation aides: A theory-W based spiral approach," 17th International Conference on Software Engineering (ICSE'95), pp.243–253, 1995.
[4] G. Kotonya and I. Sommerville, Requirements Engineering, John Wiley & Sons, 2002.
[5] S. Fricker, T. Gorschek, C. Byman, and A. Schmidle, "Handshaking with implementation proposals: Negotiating requirements understanding," IEEE Softw., vol.27, no.2, pp.72–80, 2010.
[6] Y.K. Malaiya and J. Denton, "Requirements volatility and defect density," International Symposium on Software Reliability Engineering, pp.285–299, 1999.
[7] 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.
[8] IEEE Std. 830-1998, Recommended Practice for Software Requirements Specifications, 1998.
[9] D.M. Berry and E. Kamsties, "Chapter 2: Ambiguity in requirements specification," Perspectives on Software Requirements, ed. J.C.S.D.P. Leite and J.J.H. Doorn, pp.7–44, 2004.
[10] G. Bull, ed., The Prince by Niccolo Machiavelli, Penguin Classics, 2003.
[11] Z. Markov and I. Russell, "An introduction to the WEKA data mining system," ITICSE '06: The 11th annual SIGCSE conference on Innovation and technology in computer science education, pp.367–368, 2006.
[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, "The WEKA data mining software: an update," SIGKDD Explorations Newsletter, vol.11, pp.10–18, Nov. 2009.
[13] P. Arkley and S. Riddle, "Tailoring traceability information to business needs," 14th International Requirements Engineering Conference (RE'06), pp.239–244, 2006.
[14] S. Kousik and V. Raman, "Total requirements control at every stage of product development," 15th International Requirements Engineering Conference, pp.337–342, 2007.
[15] E. Carmel, R.D. Whitaker, and J.E. George, "Pd and joint application design. A transatlantic comparison," Commun. ACM, vol.36, no.4, pp.40–47, 1993.
[16] D. Bush and A. Finkelstein, "Requirements stability assessment using scenarios," 11th IEEE International Conference on Requirements Engineering, pp.23–32, 2003.
[17] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed requirements acquisition," Science of Computer Programming, vol.20, pp.3–50, 1993.
[18] E.S.K. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," 3rd International Symposium on Requirements Engineering (RE'97), pp.226–235, 1997.
[19] H.E. Eriksson and M. Penker, Business Modeling with UML, John Wiley & Sons, 2000.
[20] T. Nakatani and T. Fujino, "Rodan: A requirements elicitation method for a business domain by focusing on roles," IPSJ J., vol.48, no.8, pp.2534–2550, 2007. (in Japanese).
[21] N. Nurmuliani, D. Zowghi, and S. Fowell, "Analysis of requirements volatility during software development life cycle," Australian Software Engineering Conference, pp.28–37, 2004.
[22] N.E. Fenton and S.L. Pfleeger, Software Metrics: A Regorous and

Practical Approach, second edition, Course Technology, 1998.
[23] S. Anderson and M. Felici, "Controlling requirements evolution: An avionics case study," SAFECOMP 2000 (LNCS 1943), pp.361–370, 2000.
[24] IEEE Std 982.1, "IEEE standard dictionary of measures to produce reliable software," 1988.
[25] 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, 2008.
[26] 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), pp.145–150, 2009.
[27] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, Object-Oriented Software Engineering, Addison-Wesley, 1992.
[28] ISO/IEC 25030:2007, "Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements," 2007.
[29] T. Nakatani, T. Tsumaki, M. Tsuda, M. Inoki, S. Hori, and K. Katamine, "Requirements maturation analysis by accessibility and stability," Asia Pacific Software Engineering Conference 2011 (APSEC2011), 2011 (in printing).
[30] Integrated Requirements Process Research Project, PRINCE Model –An approach to the integrated requirements process– (in Japanese), http://www2.gssm.otsuka.tsukuba.ac.jp/staff/nakatani/SSR09/2009ssr_RE.pdf, 2010.
[31] S. Anderson and M. Felici, "Quantitative aspects of requirements evolution," COMPSAC '02: Proc. 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment, pp.27–32, 2002.
[32] T. Nakatani, S. Hori, N. Ubayashi, K. Katamine, and M. Hashimoto, "A case study of requirements elicitation process with changes," IEICE Trans. Inf. & Syst., vol.E93-D, no.8, pp.2182–2189, Aug. 2010.
[33] H. Kaiya, M. Tanigawa, S. Suzuki, T. Sato, A. Osada, and K. Kaijiri, "Improving reliability of spectrum analysis for software quality requirements using TCM," IEICE Trans. Inf. & Syst., vol.E93-D, no.4, pp.702–712, April 2010.
[34] M. Inoki, "An approach to the optimizing productivity of requirements definitions (in Japanese)," 2011.
[35] Y. Fujiwara, "Project analysis based on PRINCE Model and study of estimating the requirements maturation process (in Japanese)," Software Engineering Symposium 2011, IPSJ, 2011.
[36] B. Boehm, "A spiral model of software development and enhancement," SIGSOFT Software Engineering Notes, vol.11, no.4, pp.14–24, 1986.
[37] S. Hori, T. Nakatani, K. Katamine, N. Ubayashi, and M. Hashimoto, "Project management patterns to prevent schedule delay caused by requirement elicitation," IEICE Trans. Inf. & Syst., vol.E93-D, no.4, pp.745–753, April 2010.

Takako Nakatani received the Ph.D from the University of Tokyo in 1998. She is an associate professor of Graduate School of Business Sciences, the University of Tsukuba, Japan. She is a member of the IPSJ, JSSST, SEA, IEEE CS, ACM, and the Society of Project Management.

Narihito Kondo received the M.B.A. from the Graduate School of Systems Management, the University of Tsukuba in 2010. From 2011, he is a lecturer of Nagoya Management Junior College, Japan.

Junko Shirogane received the B.E., M.E. and D.E. degrees in information and computer science from Waseda University, Tokyo, Japan, in 1997, 1999 and 2002, respectively. She joined Media Network Center of Waseda University as a Research Assistant in 2000 and Department of Communication of Tokyo Woman's Christian University as a Lecturer in 2003. She is now an Associate Professor of School of Arts and Sciences, Tokyo Woman's Christian University. Her research interest includes support tools for development of software with graphical user interface. She is a member of IPSJ, JSSST, HIS, IEEE and ACM.

Haruhiko Kaiya is an associate professor of Software Engineering in Shinshu University, Japan.

Shuzo Hori After he graduated from the Kyushu Institute of Technology information engineering in 1980, he is working for the Yaskawa Information Systems Corporation. He has performed the embedded software development, and management for 20 years. He has been engaged in a process improvement of an organization using ISO9001 or CMM since 1998. His interest area is 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 assistant 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. His research interests include software engineering, software and knowledge modeling techniques, and project management. He is a member of the Information Processing Society of Japan and the Society of Project Management.