Construction of a traceability matrix for high quality project management

— A proposal of a basic theory toward a change from process-centric management to information-centric project management —

Akihiro SAKAEDANI[1,2]*, Yoshiaki OHKAMI[1] and Naohiko KOHTAKE[1]

[Translation from *Synthesiology*, Vol.5, No.1, p.1-15 (2012)]

Design information is important for software development projects because it determines their cost and product. In this research, a model has been made which can trace how information moves in a project by paying attention to the design information which is hard to trace by process-centric project management. On the basis of the model, a traceability matrix method has been constructed which quantifies the complexity of the traceability. It has been confirmed that high quality information-centric project management is realized by applying the model and the method to software development projects.

Keywords **:** Traceability matrix, complexity, quantitation, information centric, project management

1 Introduction

According to PMBOK,[1] "a project is an organic work conducted to create some original product, service, or artifact." To create original products and services, it is necessary to synthesize various elements such as technological and human resources. This means that the project itself is synthesis. Under the subject of synthesiology, various efforts have been made to conduct high quality project management and there have been several previous studies. Along with Visualizing Project Management[2] and PMBOK, SWEBOK[3] and Rational Unified Process[4][5] organize know-how into project management and development process methods, and address software development. Nevertheless, the success rates of projects have not improved, particularly in software development. For example, according to a report[6] by the Japan Users Association of Information Systems, the majority of projects on the scale of 500 person-months from 2004 to 2008 are over budget, and this is a trend observed every year. Frank[7] cites the report[8] by the Standish Group that states that "the present status with projects is that 68 % of them are failures," and questions the effectiveness of current management techniques. Specifically, the standards (such as ISO15288, IEEE1220, EIA632, CMMI, INCOSE Handbook, and PMBOK Guide) for project management and systems engineering are reviewed, and the fact that they are all process-centric is indicated. Frank also states, "The current project management technology and system engineering technology seek methodology for better management," and points out the limitations of the current process-centric management.

In the present study, the objective is the construction of methodology for realizing high quality project management. To achieve this object, the whole of the project, and its details, are analyzed to construct the methodology of "seeing both the forest and the trees." As the specific methodology, the architecture (or the organization of elements that compose the system and the relationships among elements) of the project is clarified. The quantitative management of the project is made possible by using the complexity of the project architecture as an index to gain an overview of the whole project, and by creating indices for the difficulty of individual elements and the relationships among the elements (Fig. 1). Finally, the methodology to realize high quality project management is considered by using the project model for which the architecture is shown.

2 Current state of project management and analysis of relevant issues

2.1 Difference between information and objects from the perspective of transfer cost

Information and objects have different properties from the perspective of transfer cost. Hereinafter, information is defined as knowledge and know-how that people have or that have been formalized as products or texts; and an object is defined as a physically tangible thing. In this way of thinking,

1. Graduate School of System Design and Management, Keio University 4-1-1 Hiyoshi, Kouhoku-ku, Yokohama 223-8526, Japan,
2. NTT COMWARE CORPORATION NTT Shinagawa TWINS Annex Bldg. 1-9-1 Konan, Minato-ku, Tokyo 108-8019, Japan
∗ E-mail: a.sakaedani@sdm.keio.ac.jp

Original manuscript received July 26, 2011, Revisions received November 7, 2011, Accepted December 28, 2011

work progresses when the person in charge finishes one task and hands the object over to the next worker, for example, in the case of an automobile factory where the object is assembled in steps. However, in a software development project, the main subject of transfer is information, which is different from objects in terms of transfer cost characteristics. The information transfer cost is defined as the total expense required to transfer information in a form that can be used by the information seeker (receiver).[9] To control the information transfer cost, the difficulty of transferring implicit knowledge, the difficulty of transfer due to different capabilities between the receiver and sender of information, and the difficulty of transferring high-volume information must be considered (Table 1).

It is difficult to maintain the accuracy and details of the information without the worker checking the original information transferred by the previous worker. Unlike objects, it is difficult to conduct information transfer by clearly segmenting the process as in object transfer, and there is the characteristic that the same process must be repeated several times to transfer information.

2.2 Necessity of shifting from process-centric to information-centric project management

In a software development project, the main work is the transfer of information rather than a physical object. To improve the production efficiency of an information-centric software development project, mere process-centric improvement is limited to pursuing a smooth workflow according to a set procedure. The issues that arise from the information transfer characteristics (the difficulty of transferring implicit knowledge, the difficulty of transfer due to different capabilities between the receiver and sender of information, and the difficulty of transferring high-volume information must be considered) will not be solved by a process-centric approach. Accordingly, a change from process- to information-centric is necessary

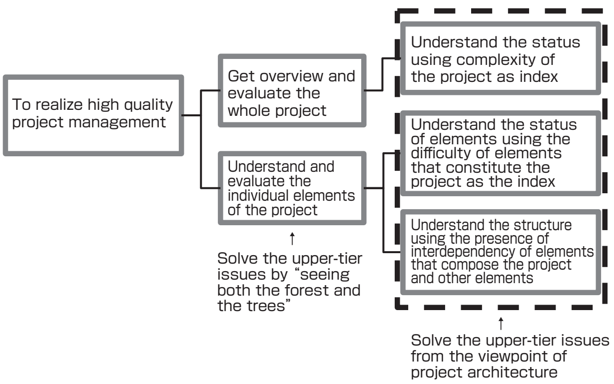


Fig. 1 Research aims

Table 1. Difference in the transfer cost of information and object

Determining factor of transfer cost	Information	Object
Characteristic of transfer subject	Large transfer cost is necessary due to implicitness of information (such as know-how).	Transfer cost does not depend on the characteristics of the object itself (instead, the cost depends on transfer conditions such as rate and volume).
Attributes of the sender and receiver	Information transfer cost varies according to the capabilities of the sender and receiver.	Not dependent on sender/receiver attributes (transfer is completed when it is handed over from the sender to receiver).
Amount to be transferred	Transfer cost increases because more information exchange becomes necessary as complexity and information volume increase.	Transfer cost increases as volume increases (even if the volume is high, it takes only one session to complete the transfer activity).

2.3 Characteristics of information: equivocality and information stickiness

The information transfer cost will be reconsidered from an information-centric perspective. Focus will be placed on the following two concepts pertaining to information. The relationships between the information transfer cost and the two concepts will be explained.

Information stickiness: the difficulty of information transfer[9]

Equivocality: the characteristic where the information may be given diverse meanings

Information stickiness is a term that describes the difficulty of transferring information from the sender to the receiver due to stickiness, a characteristic of information. Stickiness indicates that the element and its information are indivisible, and the information transfer cost arises due to this stickiness. There are three decisive factors of stickiness as shown in Table 1. When the sender and receiver of the information have a 1:1 relationship, the information transfer cost is governed by the information stickiness that arises from the three factors. For example, many companies work on formalizing implicit knowledge; this is done because the information stickiness of implicit knowledge is high and the information transfer cost, or the cost needed to teach the know-how of a worker to another within the company, is very high. It may be necessary to exchange the information several times to extract or to learn the information. It may be impossible, to begin with, to transfer information to other workers. As a result, it may become a barrier in raising the basic capacity of the company as a whole. Therefore, it is necessary to reduce the information transfer cost by reducing the number of information exchanges by formalizing the worker's knowledge (or making it visible).

On the other hand, information stickiness is insufficient to explain the information transfer cost in a 1:n relationship in an organization or a project. That is because if there are separate senders, the relationship with the receiver will be determined uniquely, and the information transfer cost can be calculated as the sum total of the information stickiness of the sender. However, if the sender is the same, the cost to modify the sender's information stickiness to individual receivers must be added. In other words, considerations must be made for the cost of preparation to change the destination of the information transfer from one receiver to another. To add this into the cost, the concept of equivocality is introduced.

Equivocality means that certain information may take on several meanings according to the receiver's perspective. For example, when information is transferred to one receiver, one piece of information may be given two meanings A and B. However, in the case where there are multiple receivers, the meaning may be not only A and B, but also C and D. As the number of information transfer destinations increases, the equivocality increases. Attempting to ensure that the multiple receivers arrive at the correct meaning, the sender of the information may add preliminary information that can be correctly understood by multiple receivers, for example, in instructional material, or the material may be rewritten to match the receiver. If the receiver is specified and the information transfer starts, this can be addressed as the issue of information stickiness. However, in the case where the receiver is not specified, or before the information transfer, it is necessary to consider the transfer cost of equivocality rather than information stickiness. As seen from above, the cost of equivocality arising from the 1:n relationship is dependent on the number of receivers. To control equivocality, it is necessary to reduce the number of receivers. Next, in the case of n:1, it is necessary to organize the thought by integrating the number n of transferred pieces of ambiguous information to provide one meaning. By maintaining a consistent meaning of n pieces of information, the information is integrated to have one meaning. The information transfer cost is expected to change since the integration work is dependent on n, and like 1:n, it is desirable that n be as small as possible.

Moreover, in a project conducted under limited budget and time, there may be cases where the activity is conducted before the information transfer is completed, due to the limitations of higher information transfer cost. To avoid such a situation, it is necessary to create and manage a condition where the transfer cost is minimized as much as possible within the project. Therefore, in an information-centric software development project, the management of information stickiness and equivocality is expected to promote accurate information transfer between the sender and receiver, and is important in realizing high quality project management.

3 Construction of the project architecture (traceability matrix)

In managing information stickiness and equivocality in an information-centric project, it is necessary to see what elements constitute the project and to organize the relationships of each element. For this purpose, a model will be constructed.

3.1 Concept of element extraction

The elements are extracted by object-oriented business modeling. According to requirements engineering,[10] which incorporates the concept of object orientation, needs,[Term 1] feature,[Term 2] and requirement[Term 3] elements are extracted to constitute the system called a project. The needs are related to features, and features are related to requirements. Function[Term 4] and component[Term 5] elements are extracted according to "all things have functions."[11] The components are related to functions. From process flow, artifact,[Term 6] activity,[Term 7] and team[Term 8] elements are extracted. Teams are related to activities, and activities are related to artifacts.

3.2 Concept for organizing relationships among elements and examples of relationships

In defining the architecture for a whole system called a project, focus is placed on two concepts: "axiomatic design"[12][13] of mechanical engineering, and "business architecture"[14][15] of organization science. In designing the organizational activity, enterprise architecture (EA) may be similar, but the EA method does not indicate the reference architecture.[Term 9 [16]] Therefore, it is necessary to define the elements that compose the system and their relationships. These two concepts have the major characteristics that they consider the manufacturing process and the organization involved in manufacturing as well as the customer requirement of what should be manufactured in the first place, and provide a guideline for the relationships among the elements.

The thinking of axiomatic design is the concept that the information is mapped between the domain of customer, functional, physical, and process, and the design activity takes place. It is the thinking where, for example, the information for requirements in the customer domain that concerns design is mapped as the specification of the function in the functional domain, to realize the function. Likely, the same information is mapped in each domain, and then it is translated and processed in the optimal form in each domain. The mapping of that information is thought to occur interactively. When the elements extracted in subchapter 3.1 are applied to this concept, it is as follows. Since the customer domain designates the customer's requirements, it is composed of needs, feature, and requirement elements. The functional domain is composed of function elements as stated above. The physical domain designates the design solution or the product itself, and therefore it is composed of component or physical elements. Since the process domain designates the production conditions, it is composed of artifact, activity, and team elements. This designates the architecture of the organization system called the development project, and the elements of the development project and their relationships are clarified.

3.3 Concept for realizing information transfer tracing

Figure 2 shows the organization of relationships of the elements that are analyzed and extracted by object orientation, based on the theories of axiomatic design.

To understand the condition, first the needs are understood and then organized as features (these two items are eliminated for simplification in Fig. 2), and are then defined as requirements. The component with implemented function to fulfill the requirement is necessary, and to create the component, there must be an artifact that is a summary of the design content. The development activity to create the artifact is executed by each team.

The transfer of information mapped in the software development project includes, for example, the following case. When realizing the function for "checking the number of input characters when the user ID (UID) and password (PW) are entered," how and where the function is implemented must be designed. For example, let us consider the following two structures: (1) the character number check is implemented as separate built-in functions within the UID and PW systems, and (2) the character number check is implemented as a standalone function that can be shared by the UID and PW systems. The component design will differ accordingly. If the shared character check function is employed, its development cannot be started until both the UID system and the PW system are established. However, when implemented as a built-in function within those individual systems, the design can be started when one of the ID systems is established. Therefore, the function, component, and activity elements translate, transfer, and influence the design information into a necessary form, and the cost of information transfer affects the quality of the project.

Next, the information transfer cost is considered. The left side of Fig. 2 shows the linear interdependency of each element, and the right side shows the mesh structure. With the mesh structure, multiple elements and information are exchanged; thus, it is thought that more information transfer cost will be needed, as explained in the example of equivocality. Also, on the right side of Fig. 2, for the element activity, the information transfer cost increases because information stickiness increases as difficulty increases. When using the original notation to summarize the content of the design, there may be some content that cannot be expressed as design information. As a result, design information may be lost and design mistakes may be induced. This is a case where high transfer cost is generated due to the factor, or the original notation, that makes the information transfer difficult. Therefore, in terms of the relationships

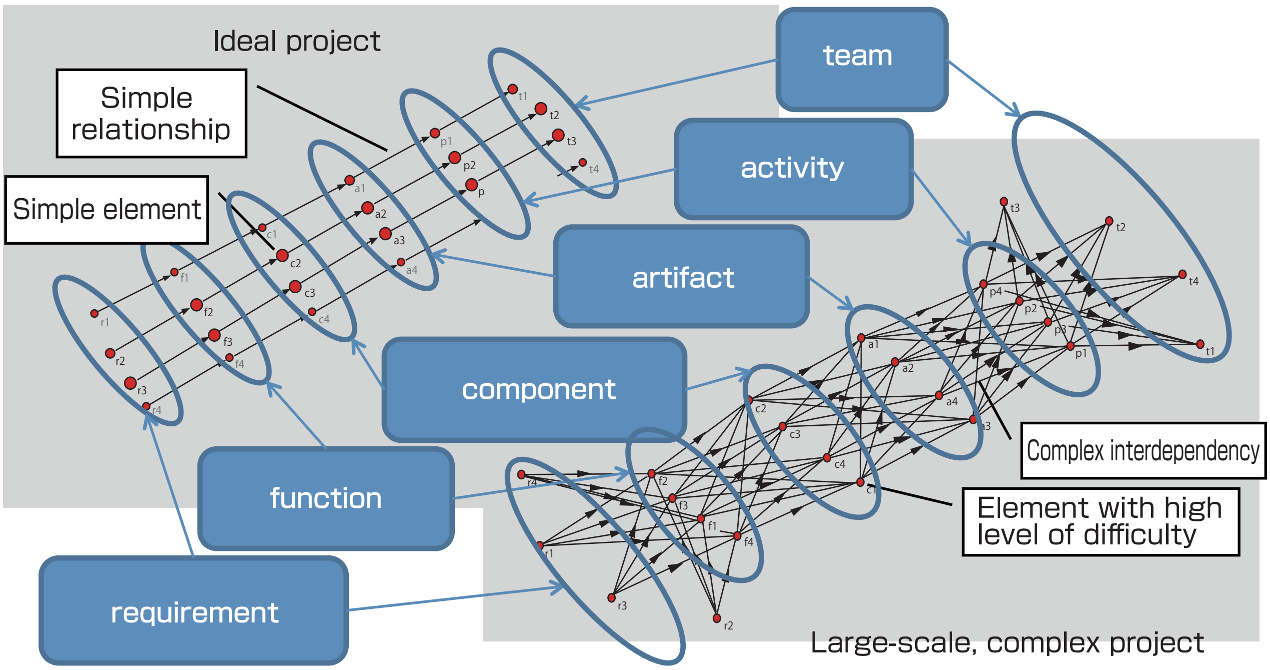


Fig. 2 Network model of an ideal project and large-scale, complex project

among elements, to reduce the information transfer cost, it is necessary to maintain independence as much as possible, or more specifically, to approximate the diagonal matrix with the highest independence when organizing the relationship between two elements using the matrix, and to reduce the difficulty (the concept that designates the difficulty of the aforementioned activity; details are explained later) of the element itself. In other words, to reduce the information transfer cost and to transfer the information accurately in the project architecture, it is necessary to control information stickiness and equivocality by using the interdependency between the elements and the difficulty of the element itself.

3.4 Project architecture

Figure 3 shows, as basic concepts, the relationship expressed as a matrix in Fig. 2.

This relationship agrees with the following matrix calculation (Equation 1) when the vector is used.

$$\begin{pmatrix} r1 \\ r2 \\ r3 \\ r4 \end{pmatrix} = \begin{pmatrix} \text{Requirement-} \\ \text{Function} \end{pmatrix} \times \begin{pmatrix} \text{Function-} \\ \text{Component} \end{pmatrix} \times \begin{pmatrix} \text{Component-} \\ \text{Artifact} \end{pmatrix} \times \begin{pmatrix} \text{Artifact-} \\ \text{Activity} \end{pmatrix} \times \begin{pmatrix} \text{Activity-} \\ \text{Team} \end{pmatrix} \times \begin{pmatrix} t1 \\ t2 \\ t3 \\ t4 \end{pmatrix}$$

$$\vec{r} = A\vec{t} \quad \cdots\cdots \text{(Equation 1)}$$

$\vec{r}$: requirement vector $\quad$ $\vec{t}$: team vector

A : system matrix

In this study, the traceability matrix obtained as a result will be called the system matrix as it designates the properties of the whole system. By multiplying the matrix, the traceability matrix (system matrix) that clarifies the relationship between the team and the requirement can be obtained

4 Index for seeing both the forest and the trees

To promote accurate information transfer among elements and to realize high quality project management, it is necessary to manage information stickiness and equivocality as mentioned earlier. To manage equivocality, it is necessary to control the relationships among elements between and within the team, activity, artifact, and other regions (i.e., to control "interdependency," hereinafter). This is because the probability of error in information transfer increases along with the number of related elements. This also means that equivocality increases with increasing interdependency. When the cost of equivocality is forcefully decreased, the possibility that multiple receivers attach the wrong meaning to a piece of information increases. In addition, to manage information stickiness, it is necessary to control the factors that inhibit accurate information transfer of the element itself (i.e., to control the "difficulty," hereinafter). Depending on the property of the element itself, the essential information that must be understood is not conveyed when transferring the information, and the possibility of error in information transfer increases. This means that information stickiness increases as difficulty increases, and the transfer may be finished before the necessary information is absorbed.

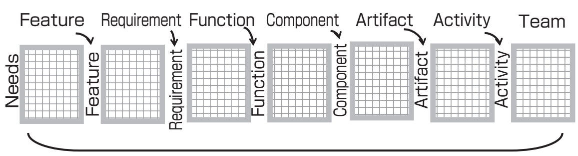


Fig. 3 Traceability matrix of project

From the above, the accuracy of information transfer is determined by interdependency and difficulty. Overall, the possibility that information is transferred accurately within the whole project is determined by the sum of the probabilities of (1) accurate information transfer by controlling interdependency and (2) accurate information transfer by decreasing the difficulty. Therefore, for the whole project, it is necessary to define the sum of the interdependency and difficulty as the complexity, and to use complexity as the control index.

4.1 Quantification of interdependency

The overall perspective improves as it nears the diagonal matrix. Moreover, equivocality decreases from the perspective of information transfer. Therefore, to evaluate the interdependency among the elements that may be factors of equivocality, the distance between the system matrix and the unit matrix can be measured. In comparing the unit matrix, it is necessary that the increasing equivocality be expressed as the increasing value of the non-diagonal component. Therefore, the linear Euclidean norm is used to measure the distance. The unit matrix is subtracted from the system matrix to be evaluated, and the Euclidean norm of that matrix will be the interdependency. However, the component value of the system matrix with respect to interdependency will be set as 1 when there are relationships among elements, and 0 when there are none. This will be called system matrix **s** hereafter.

4.2 Quantification of difficulty

The difficulty of the elements from their information stickiness will be expressed by the component values of the system matrix; the system matrix that has difficulty as the component value will be called system matrix **n**. The quantification of difficulty is made possible by evaluating the magnitude of the entire system matrix **n**. Therefore, the evaluation will be done according to how many times the matrix is greater than the unit matrix, or the Euclidean norm of system matrix **n**. However, the component value of system matrix **n** with respect to difficulty is determined by evaluating the difficulty of the elements (guideline for difficulty setting: Reference Material 1). The standard value is 1, and the value becomes greater than 1 at higher difficulty, or less than 1 at lower difficulty.

4.3 Definition of complexity

The complexity of information transfer is defined as follows:

$$\text{Complexity} = \text{difficulty} \times \text{interdependency}$$

However, difficulty is given by the Euclidean norm of system matrix **n**, and interdependency is given by the Euclidean norm of system matrix **s** minus the unit matrix (sample calculations: Reference Material 2).

The difficulty and interdependency are variables that reflect the situation of the individual elements and relationships among elements of the project. Therefore, by understanding their indices, it is possible to understand the elements of the matrix and the relationships among them. At the same time, the whole project can be understood via the changes in complexity obtained by multiplying indices.

4.4 Obtaining a square matrix

The relationships among the elements are not necessarily in a square matrix. In such a case, it is necessary to form the square matrix for the diagonalization that is necessary for the calculation of interdependency. To form the square matrix, the component value 0 is given. Since the component value is 0, the interdependency value changes as much as the degree added to the square matrix formation (this is because the diagonal component is subtracted for the row or column added by the square matrix formation, and the additional diagonal component becomes -1). However, considering that complete independence is expressed by the square matrix called the unit matrix, the change in the interdependency value by adding the component value 0 in the non-square matrix must be understood as the index that indicates the independence of the non-square matrix. From the above, the square matrix formation by adding the component value 0 is set as the rule.

5 PDCA cycle of project management using the traceability matrix

This chapter explains the methodology for understanding the whole of a project, and its details, and for "seeing both the forest and the trees." For explanation, the PDCA (plan → do → check → act) cycle will be used as the scenario (Fig. 4).

First, to create the traceability matrix, specific elements are organized and the matrix is created. The complexity of the project as a whole is reduced by improving the difficulty and interdependency in the created traceability matrix. The actual development activity is conducted in the project, and the progress is checked. The complexity of the project is used as the index of progress, and an overview of the state of the project is gained by looking at the change. If the change of complexity is on an increasing trend, it is because some problems have developed in the project, and the causative element is sought. The difficulty and interdependency of the elements are changed by simulation, and the element that

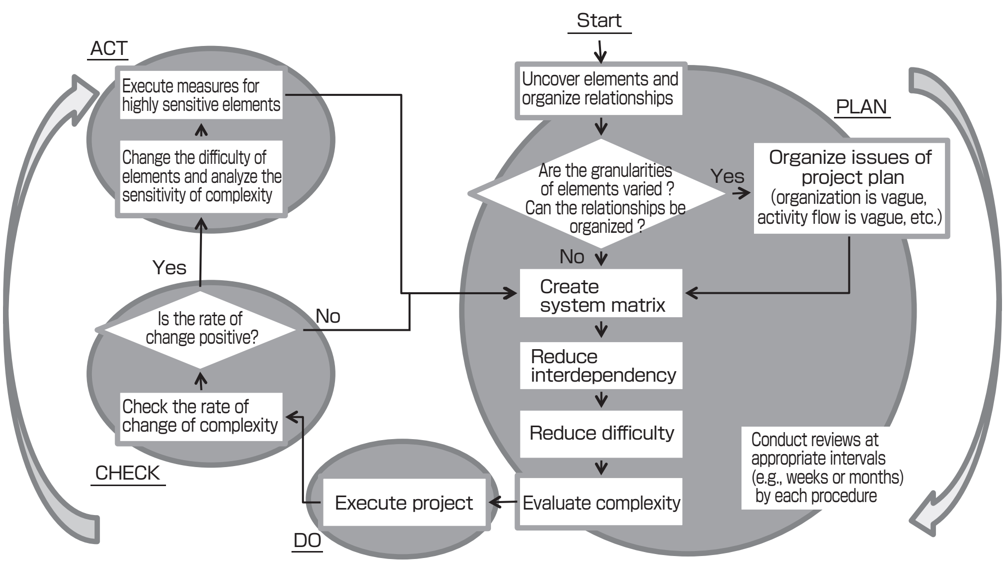


Fig. 4 PDCA cycle of project management

is most effective for reducing the complexity of the whole project is found in order to improve the project status. What follows is an explanation following the PDCA cycle.

5.1 PLAN

5.1.1 Uncovering the element and organizing the relationships

The analysis of the project status is conducted as shown in Fig. 4, the elements needed for the creation of system matrix are uncovered, and their relationships are organized. However, it can be expected that the system matrix cannot be created because the granularity of the elements may vary or the relationships are unknown. In that case, it can be assumed that there is some sort of problem in the project plan itself, and the solution is sought.

5.1.2 Organize the issues of the project plan

For example, if there is a team where the organization is loose and capable of being described in large granularity only, it is necessary to reorganize such a team. In another example, when elements are uncovered in the upstream process, the component and function may not be clear in the downstream process. In such a case, it is necessary to check the basis of the estimate. There must have been some basis when making the estimate, and if the basis is unclear, it is necessary to review the plan quickly, as an issue in establishing the project plan.

The above points are organized, and system matrix **s** and system matrix **n** are created.

5.1.3 Reduction of the interdependency and difficulty of individual matrices and the evaluation of complexity

Improvements are done to diagonalize the individual matrices. However, there may be cases where the difficulty of an element increases as a result. Moreover, there may be cases where the difficulty of the elements of the non-diagonalized matrices may increase. Therefore, it is always necessary to check the degree of influence on the whole project by calculating the complexity, to maintain the overview of the project.

As shown in Fig. 3, the system matrix that shows the project structure is calculated by the multiplication of seven matrices. Due to the properties of matrix multiplication, the solution of the calculation for the full matrix or one that contains a triangular matrix will always be the full matrix or the triangular matrix. To obtain the diagonal matrix as a solution, it is necessary to make each matrix into a diagonal matrix.

In step 1, the relationships of each of the seven matrices that constitute system matrix **s** are organized, and the interdependency is reduced by approximating the diagonal matrix. In step 2, the difficulty of the matrix is reduced by reducing the relationship with highest component value among the relationships organized in the matrices that compose system matrix **n**, or the relationships with high degree of difficulty. The complexity (= interdependency × difficulty) of the whole system matrix is reduced by the above technique.

However, in a real project, it is difficult to diagonalize all seven matrices of system matrix **s**. Therefore, the design of the project is improved (reduction of interdependency) by, for example, forming the triangular matrix for each matrix. The relationships among the elements are more simply by, for example, introducing some development tool, in order to reduce the component value of system matrix **n** (reduction of difficulty). There is also a plan to reduce the complexity of the whole project. As an alternative plan for effective improvement, the full matrix is concentrated into one or two matrices of the seven matrices of system matrix **s**, the remaining five or six matrices are diagonalized, and thereby the component value of system matrix **n** is made as low as possible. In this way of thinking, while some parts of the full matrix may have high complexity, by reducing the complexity of the diagonalized area, the complexity of the whole is reduced. However, since some cases may not necessarily be effective for decreasing the complexity, it is necessary to conduct a comparative review by simulation at the design stage of the project.

For example, in planning scratch development[Term 10] where software is all made by hand, the case of using the tool to automatically generate the source code (hereinafter, called “generator”) or the case of using the package software (hereinafter, called “PKG”) are considered to reduce the complexity. The flow of this review will be explained (Fig. 5). The relationship of the whole elements when scratch development is done is shown in Reference Material 3. The elements that are considered particularly important are extracted in Fig. 6. The figure deals with only the screen

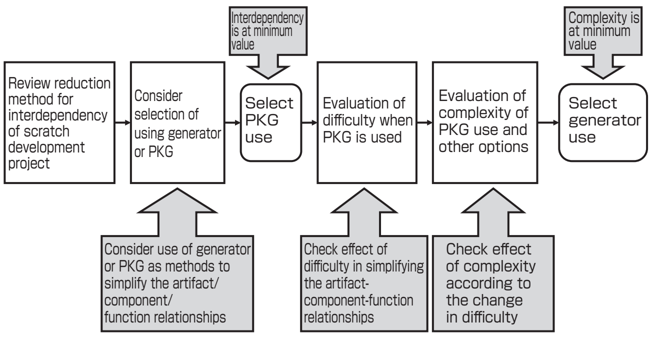


Fig. 5 Evaluation flow of complexity considering the interdependency and difficulty

Table 2. Team-activity matrix

		Team				
		Requirement management team	Architecture design team	Detailed design team	Implementation team	Test team
Activity	Screen design	2.25	2.25	2.25	2.25	2.25
	Work logic design	1.5	2.25	1.5	1.5	1.5
	DB design	1.5	2.25	1	1	1

Table 3. Final evaluation of each plan

	Interdependency	Difficulty	Complexity
Scratch development	77.79	78.18	6081.42
Generator	64.40	64.75	4169.44
PKG (low skill level)	57.16	104.89	5995.12

transition (the presentation layer of the so called three-layer model). The activity elements are the screen design; artifact elements are the screen specification and screen transition diagram, and their source code; components include the login screen and menu screen; and functions include the UID input function and PW input function.

Since there are many elements with high interdependency, improvement is attempted. Since the PKG includes all of the functions within the product, the designer and programmer do not think about the individual components or functions, and therefore, the component is the screen control PKG and the function is the screen control function only (Fig. 7).

For the generator, the structure of the components and functions is the same as in the case of scratch. Since the source code is generated automatically from the design diagram, the designer and programmer do not have to think about the source code. Therefore, the difference from scratch in terms of element is the presence of the source code (Fig. 8).

When the above three types are compared and evaluated, it is thought appropriate to select the PKG, which has the lowest interdependency. However, the effect of peripheral elements when such a selection is made is reconsidered. This means that according to the level of proficiency in the PKG, the difficulty of some other elements may increase. The component value of each system matrix **n** is set to a standard value of 1, but as the proficiency of this PKG is considered, the component values are set as in Table 2 (the component value of other matrices such as relationships of function and component remain 1). In Table 2, difficulty levels (1

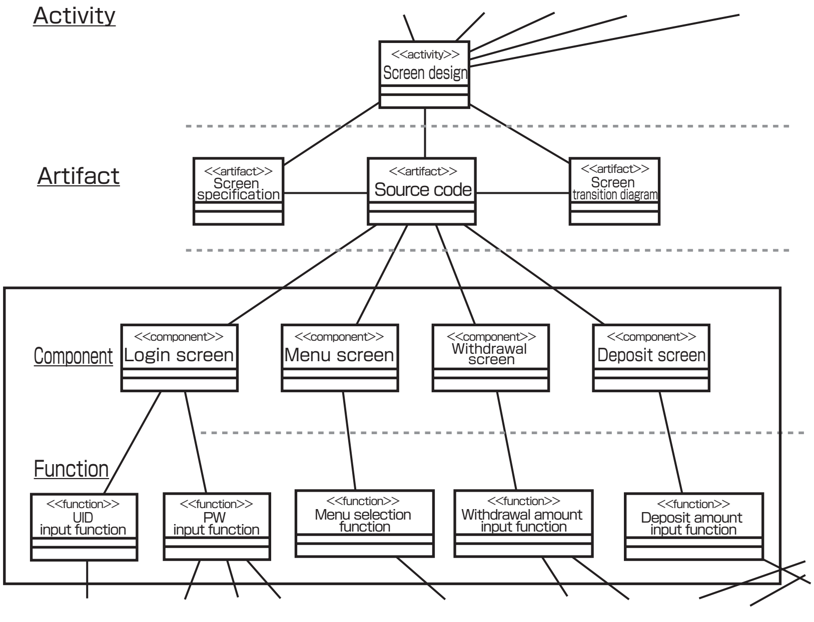


Fig. 6 Network model of scratch development (for screen transition only)

or 1.5) are assigned to each team and activity element, and the component value of the matrix is set by multiplying the difficulty of the row and column.

As a result, it is found that a difficulty level of 104.89 is obtained for the whole project (Table 3). This is greater than the value of 78.18 in scratch development, and is also greater than the difficulty level of 64.75 when the generator is used as an alternative plan.

Based on the above data, the complexities of the project when PKG, generator, and scratch are used are reevaluated. The results are shown in Table 3, and the conclusion is that the use of the generator is optimal.

5.2 DO

The project is executed as planned, and the development is carried out. For example, in the aforementioned case, the development is conducted using the generator.

5.3 CHECK

Up to the previous subchapter, the comparison of multiple projects is discussed, but in this subchapter, the discussion will be focused on one project and on its changes over time.

The complexity decreases as development progresses. This is because the interdependency disappears between elements where information transfer has been completed, and ultimately the value of matrix elements that initially showed interdependency will reach 0. At the same time, the difficulty decreases with the change in progress over time, owing to the effect of proficiency. However, the complexity does not necessary become an elemental value of 0. For example, it is normally difficult to understand all the specifications of the package software and to become proficient in skills that allow dealing with any kind of situation within the limited development period. However, the interdependency becomes 0 when the information transfer is completed. Therefore, by setting the system matrix that expresses difficulty as Sn and the order of the system matrix as N, in the matrix component

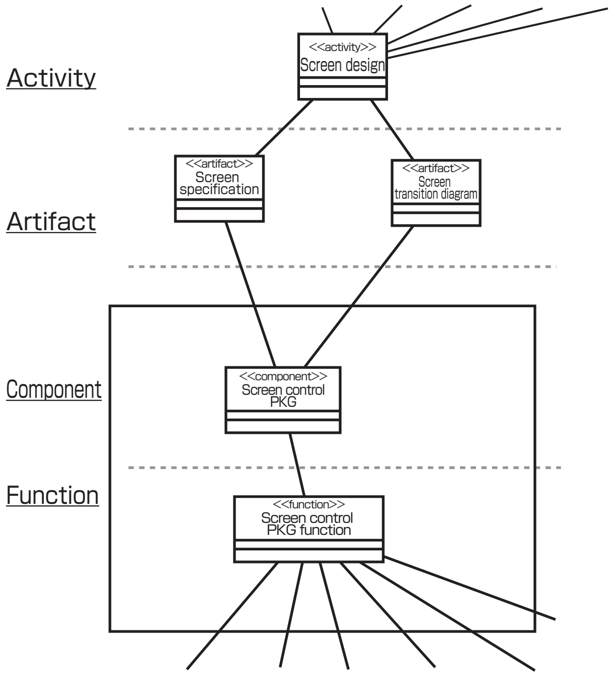


Fig. 7 Network model of PKG development (for screen transition only)

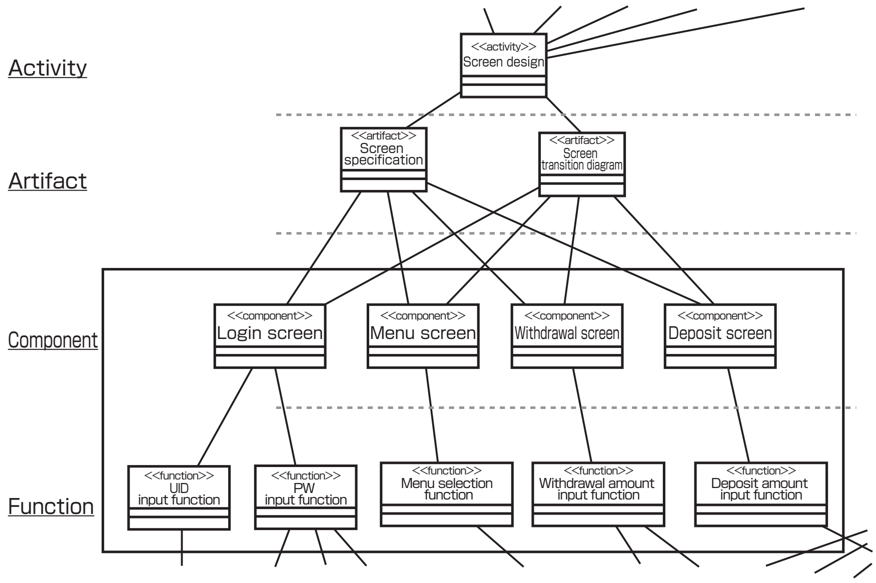


Fig. 8 Network model of generator development (for screen transition only)

where the unit matrix is subtracted from system matrix **s**, all diagonal components will be -1. Therefore, the norm will be $\sqrt{N}$, and the complexity will converge to the following value (Equation 2).

$$\text{Complexity} = \sqrt{N} \times \|Sn\| \quad \cdots \text{(Equation2)}$$

A system matrix that is perfectly diagonalized is considered a singularity, and is not subject to Equation 2. A perfectly diagonalized system matrix is an ideal project according to Fig. 2, and is not a subject of discussion in regard to complexity.

Therefore, two perspectives can be considered as ways to manage the progress status of the whole project.

One perspective is gaining an overview of the whole project (seeing the forest), and progress is monitored by using the complexity as the managing index. If the complexity is decreasing, it shows that the development is progressing smoothly, whereas if the complexity is increasing, it shows that some problem may be occurring in the project (Fig. 9).

The other perspective is managing the individual elements of the project (seeing the trees), and progress is monitored by using interdependency as the managing index. For example, the element allotted to each team is determined, and the interdependency of the elements is managed. One proposal is to use the number of interdependencies and the rate of change as indices of progress.

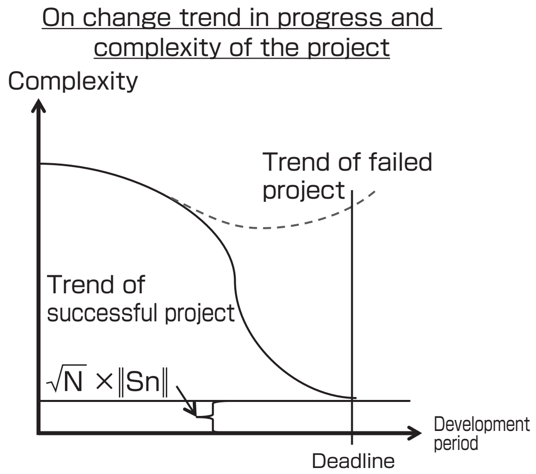


Fig. 9 Successive change of complexity and project status

5.4 ACT

From the perspective "seeing the trees" of the project, if there is a positive rate of change in complexity, an overview of the problems is gained from the model and the issues are uncovered. If the problem can be solved directly and the difficulty and interdependency can be reduced, measures are implemented. If the measures cannot be taken directly, simulation is done for other elements and a sensitivity analysis is done (Fig. 10). In this example, it can be seen that element a is more effective for reducing the complexity than element b.

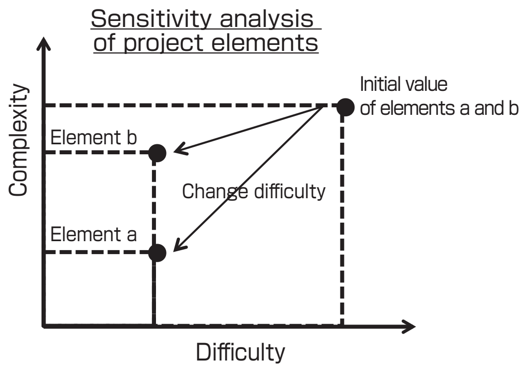


Fig. 10 Example of sensitivity analysis for project complexity

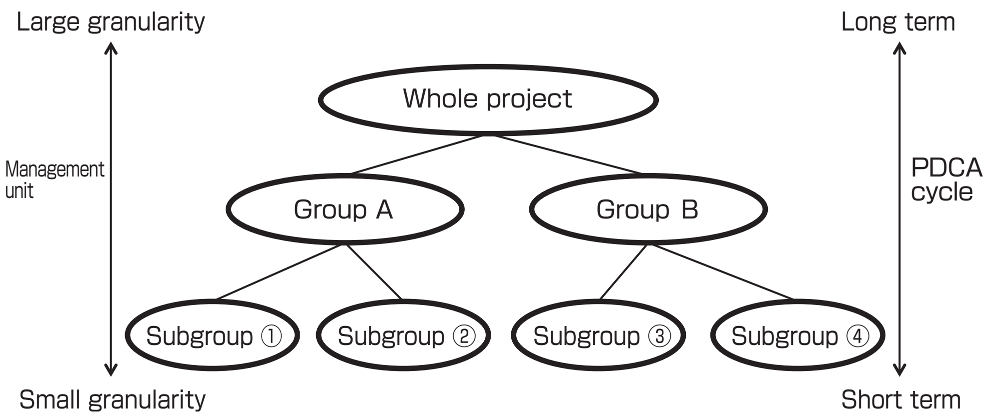


Fig. 11 Management unit and frequency

The element effective for reducing the complexity is uncovered through the reduction of difficulty of the element, and the elements to be dealt with are narrowed down. When the element is determined, necessary measures are implemented to reduce the difficulty of that element.

5.5 Notes in executing the PDCA cycle

From the perspective of project management operations, the following points must be taken into account when executing the PDCA cycle.

First, it is necessary to review the granularity of the elements that constitute the matrix and the management unit of the system matrix. One must determine whether to create the system matrix to gain an overview of the whole project, or to create the system matrix with groups and subgroups within the matrix. If the elements are too detailed, management operations will be overwhelmed by the maintenance and management of the matrix, and proper management will be impossible. Next, it is necessary to consider iterations of the PDCA cycle. In regard to the above, as the management unit of system matrix or the granularity of elements increases, the cycle should be longer. If the granularity is low, the cycle should be short (Fig. 11).

Specifically, the management unit of the system matrix should be set so that the number of elements will be 10 to 20 for each regional unit such as team, activity, or artifact. For a greater number of elements, the understanding of the current project status will be more difficult when using the system matrix and the network model, and improvements will be difficult to implement. To keep the number of elements to 10 or 20 in each region, it is necessary to limit the number of PDCA cycles as well as the management unit of the matrix. For example, when the elements for the total development process are uncovered at the subgroup level, the number will be great and will be far more than 10 to 20. From this perspective, it is necessary to limit the number of cycles.

Information-centric project management can be executed by considering the above points.

6 Conclusion

A traceability matrix was constructed as a framework for managing a software development project. For the scenario of the PDCA cycle of management in an information-centric software development project, the method using the traceability matrix was explained. Also, it was explained that this method enables gaining an overview of the project and understanding its detailed status as a whole. Furthermore, this method was shown to be an information management method for realizing high quality project management.

Acknowledgement

This research was conducted as part of the Global COE Program "Center for Symbiotic, Safe and Secure System Design" of the Ministry of Education, Culture, Sports, Science and Technology. The authors thank Professor Ryuichi Teshima and Project Professor Toshiyuki Yasui of the Keio University Graduate School for discussions of this research.

Reference materials

Reference Material 1: Guideline for setting the difficulty

◆ Setting the standard of difficulty of the element as 1, ranking is provided according to the following indices.

No.	Element to be surveyed	Subject of measurement	Measurement index (example)	Difficulty ranking policy
1	Team	Skill of each team or personnel	Years of experience, proficiency level (skill)	Highly skilled team → Low difficulty
			Role (leader? assistant? etc.)	
2	Activity	Importance of each activity	Is this activity a critical point? (Is it supplementary activity?)	Important activity → High-difficulty standard activity → Low difficulty
			Is it standard activity?	
3	Artifact	Importance of each artifact	Is special technology needed in creating the artifact? (Is it necessary to learn special language, etc.?)	Important artifact → High difficulty (special technology necessary → high difficulty, high referencing → high difficulty, amount of artifact→ high difficulty, high reusability → low difficulty)
			Is it frequently referenced while conducting other activities? (assumed that important artifact will be referenced many times)	
			Is the number of pages or lines greater for the measured artifact than the others?	
			Reusability (Is the artifact reused from another project?), or expandability of base by prototyping, etc.	
4	Component	Importance of each component (basic function, frequency of use, etc.)	Is the component essential to realize a high-priority requirement?	Important component → High difficulty (high priority → high difficulty, used many times → high difficulty, high reusability → low difficulty)
			Is the component frequently used to realize the requirement? (assumed that major component will have high number of interfaces with other components)	
			Reusability (Is it a reuse of artifact from another project?), or expandability of base by prototyping, etc.	
5	Function	Importance of each function (basic function, frequency of use, etc.)	Ils the function essential to realize a high-priority requirement?	Important function → High difficulty (high priority → high difficulty, used many times → high difficulty, high reusability → low difficulty)
			Is the function frequently used to realize the requirement? (assumed that operation frequency of major function will be high)	
			Reusability (Is it a reuse of artifact from another project?), or expandability of base by prototyping, etc.	
6	Requirement Feature Needs	Importance and priority of requirement	High or low priority of requirement to be realized	High-priority requirement, feature, and needs → High difficulty

• In setting the ranking, the following should be noted.

The standard is set as 1, and it is necessary to assign values from 0.1 to 1.9 by dividing by the number of elements to be surveyed within the same category. For example, if there are 10 elements but there are only three difficulty values, the difference of the difficulty level of the ten elements will be rounded off when the values are assigned, and as a result, the difference will be hard to see. Of course, there is no problem in setting the difficulty in three steps, if it is determined that the difficulty levels are truly the same. The figures are set from 0.1 to 1.9 because the upper and lower limits are set within the same range when 1 is set as the standard.

Reference Material 2: Example of complexity calculation

(r1, r2) is the vector that indicates the requirements, while (t1, t2) is the vector that indicates the teams.

The complexity of the development project expressed by this system matrix is calculated. For interdependency, the component value of the matrix is set to either 0 or 1 according to the presence of the relationship (system matrix **s**). For difficulty, the figures are set according to the difficulty level of each element to determine the matrix component value (system matrix **n**). The complexity of the whole development project is calculated by multiplying the interdependency by the difficulty, and the result is as follows.

$$\begin{pmatrix} r1 \\ r2 \end{pmatrix} = \begin{pmatrix} \text{Requirement-} \\ \text{Function} \end{pmatrix} \times \begin{pmatrix} \text{Function-} \\ \text{Component} \end{pmatrix} \times \begin{pmatrix} \text{Component-} \\ \text{Artifact} \end{pmatrix} \times \begin{pmatrix} \text{Artifact-} \\ \text{Activity} \end{pmatrix} \times \begin{pmatrix} \text{Activity-} \\ \text{Team} \end{pmatrix} \times \begin{pmatrix} t1 \\ t2 \end{pmatrix}$$

$$= \begin{pmatrix} 1/2 & 1 \\ 1 & 1/2 \end{pmatrix} \times \begin{pmatrix} t1 \\ t2 \end{pmatrix}$$

$$\text{Difficulty} = \left\| \text{system matrix } \mathbf{n} \right\|$$

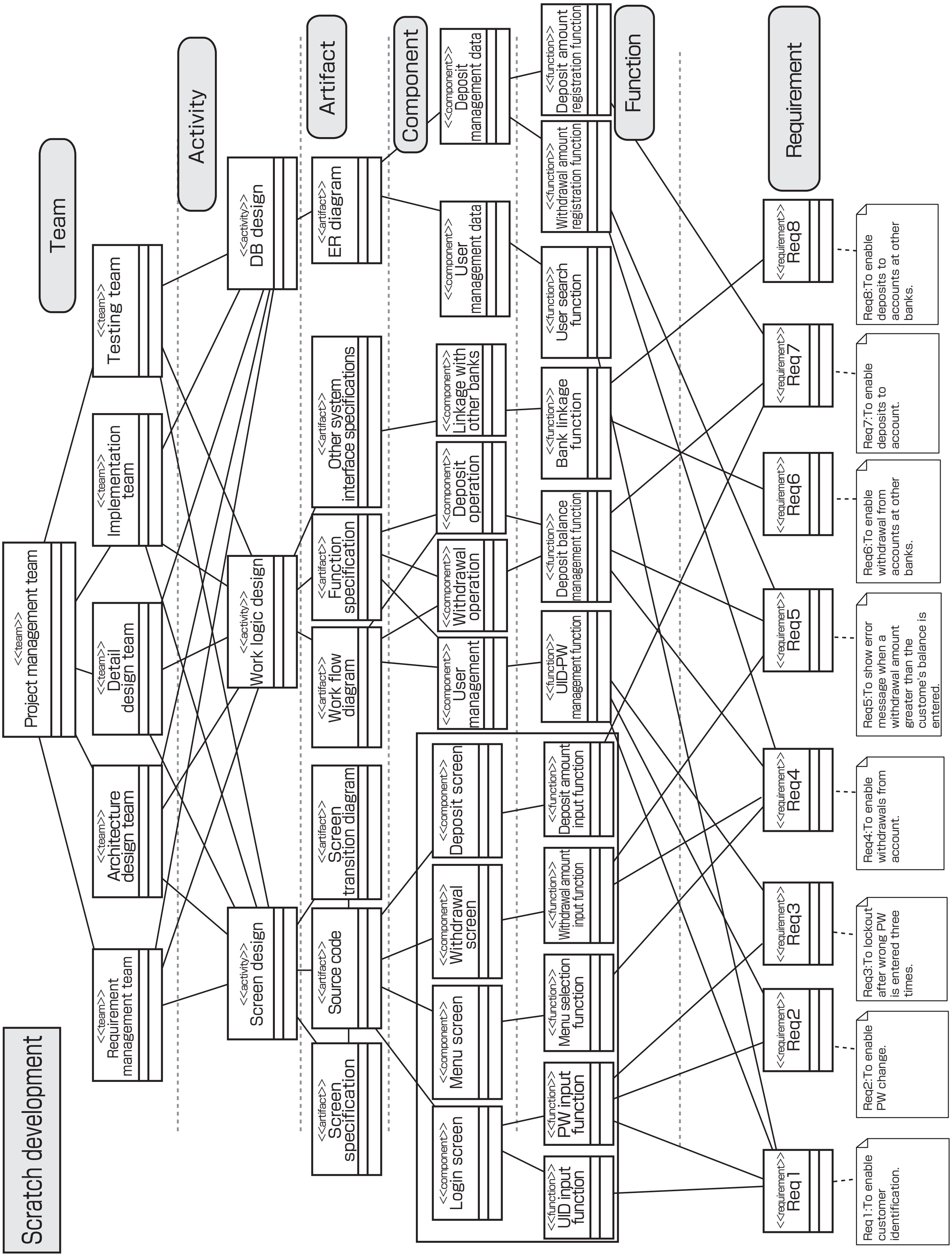
$$= \left\| \begin{pmatrix} 1/2 & 1 \\ 1 & 1/2 \end{pmatrix} \right\| = \sqrt{5/2}$$

$$\text{Interdependency} = \left\| \text{system matrix } \mathbf{s} - \text{Unit matrix} \right\|$$

$$= \left\| \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\| = \left\| \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right\| = \sqrt{2}$$

$$\text{Complexity} = \sqrt{5/2} \times \sqrt{2} = \sqrt{5}$$

Reference Material 3: Network model of scratch development project



Terminology

Term 1. Needs expressed, for example, as "we want x."
Term 2. Feature: expressed, for example, as "system is to realize x."
Term 3. Requirement: expressed, for example, as "x is done by the system, and y is output."
Term 4. Function: indicates the roles of the substance or component in realizing the requirement.
Term 5. Component: has implemented artifact and has function.
Term 6. Artifact: produced by activity; for example, a design plan.
Term 7. Activity: indicates work done by the team, produces artifact.
Term 8. Team: individual or group assigned to various roles within the organization.
Term 9. Reference architecture: architecture created especially for a certain region, and used as reference in conducting the system design for that region.
Term 10. Scratch development: development in which the developer implements everything

References

[1] Project Management Institute: *A Guide to the Project Management Body of Knowledge: Official Japanese Translation*, Global Standard (2008) (in Japanese).
[2] K. Forsberg, H. Moozand and H. Cotterman: *Visualizing Project Management Models and Frameworks for Mastering Complex Systems, Third Edition*, John Wiley & Sons (2005).
[3] *SWEBOK*, http://www.computer.org/portal/web/swebok, accessed in 2011.
[4] P. Krucheten: *The Rational Unified Process: An Introduction, Second Edition,* Pearson Education (2001).
[5] W. Royce: *Software Project Management A Unified Framework*, Addison-Wesley (1998).
[6] Japan Users Association of Information Systems: *Dai 15 Kai Kigyo IT Doko Chosa 2009* (15th Corporate IT Trend Survey 2009), http://www.juas.or.jp/servey/it09/summary09_0507.pdf, accessed in 2010 (in Japanese).
[7] M. Frank, A. Sadeh and S. Ashkenasi: The relationship among systems engineers' capacity for engineering systems thinking, project types, and project success, *Project Management Journal*, 42 (5), 31-41 (2011).
[8] Standish Group: Chaos Summary 2009: The ten laws of chaos, http://www1.standishgroup.com/newsroom/chaos_2009.php, accessed in 2011.
[9] E. V. Hippel: "Sticky information" and the locus of the problem solving: Implications for innovation, *Management Science*, 40 (4), 429-439 (1994).
[10] D. Leffingwell and D. Widrig: *Managing Software Requirements: A Unified Approach*, Addison-Wesley (2000).
[11] H. Yoshikawa: Introduction to theory of service engineering: Framework for theoretical study of service engineering, *Synthesiology*, 1(2), 111-122 (2008).
[12] N. P. Suh: *Axiomatic Design: Advances and Applications*, Oxford University Press, New York (2001) [M. Nakao, K. Iino and Y. Hatamura, trans.: *Koriteki Keikaku*, Morikita Publishing, Tokyo (2004)].
[13] N. P. Suh: *The Principles of Design*, Oxford University Press, New York (1990).
[14] Y. Aoshima and A. Takeishi: Architecture thinking, *Business Architecture*, Ch. 2, 27-70, Yukikaku (2001) (in Japanese).
[15] A. Takeishi, T. Fujimoto and S. H. Ku: Modularization in automotive industry: Interlinked hierarchies of product, production, and procurement system, *Business Architecture*, Ch. 4, 101-120, Yukikaku (2001) (in Japanese).
[16] *TOGAF*, https://www2.opengroup.org/ogsys/jsp/publications/PublicationsBySubjectType.jsp?limit=mainSubjectId-50:secSubjectId-50:statusId-1, accessed in 2011.

Authors

Akihiro SAKAEDANI

Currently in residence at the doctoral program at the Graduate School of System Design and Management, Keio University. Works at NTT COMWARE Corporation. Completed the master's program at the Graduate School, Gakushuin University, in 1994, and joined NTT. Worked as a system engineer in information system development for about 15 years. Completed the master's program at the Graduate School of System Design and Management, Keio University, in 2010. In this study, proposed the concept of the traceability matrix and was in charge of the construction of scenario and solution.

Yoshiaki OHKAMI

Received a Doctor of Engineering from the Graduate School of Engineering, Tokyo Institute of Technology, in 1968. Worked at the National Aerospace Laboratory of Japan, Science and Technology Agency; Department of Mechano-Aerospace Engineering, Tokyo Institute of Technology; Professor, Department of System Design Engineering, Keio University; and Professor and Chairman, Graduate School of System Design and Management, Keio University. Currently, Advisor, SDM Research Institute, Keio University. Visiting Researcher at UCLA, and Research Director, Japan Aerospace Exploration Agency (JAXA). Specializes in dynamics and control of space system. Fellow of the Japan Society of Mechanical Engineers. INCOSE Fellow. Member of The Society of Instrument and Control Engineers, The Japan Society for Aerospace and Space Sciences, IEEE, and others. For this study, was in charge of mathematical verification.

Naohiko KOHTAKE

Completed the course at the Graduate School of Science and Technology, Keio University, in 1998. Joined the National Space Development Agency of Japan (NASDA) in 1998. Engaged in the R&D for devices onboard the H-IIA rocket. After working as a Visiting Researcher at the European Space Agency, returned to the Japan Aerospace Exploration Agency (JAXA) as a senior developer to work on the independency

verification and efficacy check of software onboard spacecraft. Currently, Associate Professor, Keio University and Director, VSE Center, Keio University. Engages in research in systems engineering, process assessment, and design and management of space systems and ubiquitous systems. Member of INCOSE, IEEE, Information Processing Society of Japan, and others. Promotes international collaboration and interactive courses at the Graduate School of SDM. Doctor (Policy and Media). For this study, was in charge of research integration.

Discussion with Reviewers

1 The paper as a practice of synthesiology

Comment (Motoyuki Akamatsu, Human Technology Research Institute, AIST)

I understood this paper as a proposal for a traceable modeling method of how the design information for software is communicated or how the information travels in an organization. It shows how quantitative evaluation can be done for the level of complexity of a project, using this method.

To create software quickly and accurately, it is important to design a project eliminating complexity, and in that sense, this is a paper related to synthesiology. However, it focuses on the explanation of the method for traceability matrix to evaluate the information communication efficiency. Although it is shown that the matrix can be used as a comparative tool of complexity, there is no example given where it is used directly, for pointing out the relevance to synthesis. Please add this point.

Comment (Hideyuki Nakashima, Future University Hakodate)

Please indicate specifically how the method proposed here is useful in synthesis.

Answer (Akihiro Sakaedani)

The point is explained in "5.1.3 Reduction of the interdependency and difficulty of individual matrix and the evaluation of complexity." I explained how the individual matrix of the traceability matrix should be based on the properties of matrix calculation, and used specific examples to describe how to create the best synthesis using interdependency and difficulty.

2 Utilization of this method in management

Comment (Motoyuki Akamatsu)

I can intuitively understand that the management becomes difficult as the complexity of the organization increases. However that is a hypothesis only, and I think you should describe how much management would become easier if this method is used. Since the specific management method is an important point in synthesiology, please indicate specifically how management will become easier by using this method, and how management should be done based on the indices obtained in this method.

Answer (Akihiro Sakaedani)

I described the PDCA cycle of management using the traceability matrix in "5. PDCA cycle of project management using the traceability matrix." An explanation is added from the perspective of evaluating the project status using complexity as the index, and from the perspective of evaluating the status of individual elements by focusing on the difficulty and interdependency.

3 PDCA cycle

Question (Motoyuki Akamatsu)

You write about the method of turning the PDCA cycle using the traceability matrix, but I imagine that calculating the complexity by rewriting the relationships and difficulties on the system matrix while turning the cycle is a rather troublesome task. I don't think it is easy to see where the element of a certain region is used in another region, or how to determine the magnitude of the difficulty value. I think you need some maneuvering to turn the PDCA in a realistic manner. Can you please present your thoughts on this point?

Answer (Akihiro Sakaedani)

I added the point you indicated. An explanation is given of reviewing the unit of management and selecting the appropriate granularity of the elements, and of setting the PDCA cycle that matches the granularity. Below, I give a detailed view of the setting of component values of the system matrix below.

• On setting the difficulty

There are two types of difficulties in the difficulty setting. One is to understand the difficulty of an element qualitatively, and the other is to quantify that qualitative understanding.

First, I shall explain the qualitative understanding of difficulty. Understanding the difficulty qualitatively also involves entering information such as the progress and risk of the project. Since progress management and risk management are done by using conventional project management technology, they can be organized, without a problem, as input information in setting the difficulty level. In fact, the project manager, architect, or team leaders have an intuitive understanding of the changing difficulty of each element. For example, in many projects, in the everyday conversion that takes place during the project, there are many discussions pertaining to the difficulty of elements that constitute the project, such as who is the key person, which tool has problems, or which activity is critical. Therefore, there should not be a particular barrier in understanding the difficulty qualitatively.

However, there are issues in quantifying the risk items into difficulty levels. I have yet to verify whether independent evaluations can arrive at equivalent results based on the guidelines indicated in the paper, and this is an area targeted for future research.

• On setting the interdependency

In a project without some sort of activity standard, it is reasonable to expect difficulty in organizing the interdependency. In a project without an activity standard, current process-centric management is difficult in the first place. In that sense, using the proposed model may appear cumbersome due to lack of experience, but I believe it can function effectively as a tool for understanding the status of the project. In a project that already has an activity standard, in general, it defines activities and artifacts, as well as the roles of each team, and it is possible to organize the interdependency in that phase. Once the organization is done, each project can be customized by using the template, and there should be no problem in reusing such a template.

• Overall

As you indicated, this proposal may, at first glance, be a very difficult management technology. In conventional project management, the main method is the management of progress status centering on process and the management of budget based on process. Therefore, the problems that arise from those perspectives were analyzed and measures were taken against the cause of the problems. However, the concept of this method is to understand the problems arising in the project through the difficulty and interdependency of individual elements based on the model, and then taking measures after gaining an overview of the whole project. This means that conventional project management relied on an individual optimal solution centered on individual elements because it was analytical, but what is proposed here is a tool to consider a solution that is optimal for the whole project. In that aspect, while there may be cases where a difference in thinking may become a barrier, if a shift in thinking can be made, no major problem is expected. As I mentioned in the section on difficulty, project managers and leaders have intuitive understanding, and I believe the proposed method is much easier than the execution of something like earned value management (EVM).

4 Completed project

Question (Hideyuki Nakashima)

You say, “The interdependency disappears among the elements when the information transfer has been completed.” Does this mean there is no information dependency between completed projects? However, for example, if some new technology is discovered and the product of Project 1 is improved, isn’t it the case that the same thing can be applied to the product of Project 2?

Answer (Akihiro Sakaedani)

Consideration of the presence of interdependency changes according to how the unit of the project is defined. If you consider Project 1 that established a new technology and the subsequent Project 2 as a single large project, the interdependency will not disappear. However, if you consider Project 1 and Project 2 to be two different projects, interdependency in Project 1 disappears, and information transfer is reflected in the evaluation, such as in setting the difficulty of the team lower in Project 2 than in Project 1, due to the improved skills of the team, for instance.