Modeling How to Understand a Target System: Bridging the Gap Between Software Engineers and Usability Experts

Yukiko Tanikawa[1,4](✉), Hideyuki Suzuki[2], Hiroshi Kato[3],
Shin'ichi Fukuzumi[1], and Etsuko Harada[4]

[1] NEC Corporation, Kawasaki, Japan
y-tanikawa@cw.jp.nec.com, s-fukuzumi@aj.jp.nec.com
[2] Ibaraki University, Mito, Japan
hideyuki@suzuki-lab.net
[3] The Open University of Japan, Chiba, Japan
Hiroshi@kato.com
[4] University of Tsukuba, Tsukuba, Japan
etharada@human.tsukuba.ac.jp

Abstract. In order to investigate the reasons why software engineers fail in design activities concerning usability, we devised a method to model the features of how to understand a target system. It is the method to visualize differences of system understanding between a software engineer and a usability expert from three aspects, namely qualitative difference, quantitative difference, and difference in recognition to the understanding. Moreover we devised "three layer model diagram" which consists of "system function layer", "task and workflow layer" and "user practice field layer", for describing system understanding. We conducted the experiment and interview for software engineers and analyzed them applying this method. This analysis revealed that software engineers who failed in the activities have a tendency to understand tasks correspond to functions, in addition to understanding an entire system from functions. These suggested that their function-based understandings and little recognition to "system users" and "user practice in the field" in their everyday design activities are included in their failure reason.

Keywords: Design activity · Software engineer · Usability expert · User practice · User task

1 Introduction

Advance in ICT technologies has enabled to realize various functions easily, and it has become more difficult to differentiate an information system in function and performance. Meanwhile, use environment of information systems has become diverse rapidly with the spread of smart devices (smartphones and tablet terminals). The customer of an information system used in work (a business system) has come to emphasize what kind of experience is obtained through the use of the system, more than its function and performance. Usability is one of the central factors of this user experience.

M. Kurosu (Ed.): HCI 2016, Part I, LNCS 9731, pp. 220–232, 2016.
DOI: 10.1007/978-3-319-39510-4_21

In order to make an information system usable and useful, it is necessary to put great importance to the viewpoint of users and to design an entire system, in addition to system development technology [1]. Moreover it is also important to consider cognitive and action process of users and to design interface or interaction between users and a system [1]. Then, design approach (human centered design [2]), design methodologies [3, 4], design principles [1, 5] and tools [6, 7] have been developed, to support software engineers who design a system. These are based on explicit knowledge which is formed through experiences of usability experts. We have worked on developing tools based on explicit knowledge [8, 9], integrating human centered design approach into an existing design process and developing a support method to apply the integrated process [10, 11]. However, even if these are offered, past studies [12, 13] and field trials of our method [14] revealed that there were still a lot of difficulties for software engineers in design activities considering usability.

Through the analysis of our trial results, we found that software engineers have tendencies to fail in specifying tasks to do with a target system, users in charge of the tasks, and workflow to accomplish each task [14]. An appropriate understanding of relations among a system, its users and their tasks is basics for developing a system usable and useful [15]. The activities which software engineers failed are related to this understanding. Thus, in order to investigate the reasons of their failures, it is necessary to clarify the features of how software engineers understand a target system.

Hinds [16] found about cognitive features of software engineers that persons in a system provider with extensive knowledge of a system underestimated novice performance time to complete a complex task using the system. This was revealed through the comparative experiment among experts (persons in the system provider), intermediate users (persons having 3–9 months of experience in using the system) and novice. Seffah [12, 17] and Ferre [13] point out that terminology, concepts, the perception of the role, perspective and emphasis points in system development are different between software engineers and usability experts who assume a role of usability improvement in system development project. They also pointed out that these differences cause miscommunication when software engineers and usability experts collaborate in a system development project [12, 17] as well as disuse of usability improvement techniques by software engineers [13]. However they have not investigated specific cognitive features of software engineers.

In this paper, we focus on the cognitive features of software engineers and investigate especially how software engineers understand the target system in the context of system development. Based on these features clarified, we consider why software engineers fail in design activities concerning usability.

2 Method to Model the Features of How to Understand a System

In order to investigate the features of how software engineers understand a target system in the context of system development, we thought that it is necessary to visualize the following three points through comparing their understandings with those of usability experts being voice of system users.

When a software engineer and a usability expert understand the same system,

(a) What kind of point are both understandings different in?: Qualitative difference of system understanding.
(b) How much are both understandings different?: Quantitative difference of system understanding.
(c) How different are both recognitions to “range to be conscious of and “emphasis point” in system development?: Difference in recognition to system understanding.

Relations of the differences visualized from the aspect of above (a)(b)(c) are analyzed and system understanding peculiar to software engineers is modeled.

2.1 Three Layer Model Diagram to Describe How to Understand a Target System

Before visualizing above (a)(b)(c), it is necessary to visually express how to understand the relations among functions of a target system, its users and their tasks, which are essential to design a system with high usability. There are system visualization expression methods which are commonly employed in software engineering, such as workflow diagram, use case diagram, and activity diagram [18]. However, these expression methods are not enough to describe system understandings or to enable intuitively grasping them through the description. For example, workflow diagram can describe tasks, workflow of each task and a role that perform each work item. However it has difficulty in describing relations between these and functions or users. Use case diagram can describe relations between users (actors) and tasks, while it cannot describe workflow (chronological order). Activity diagram can describe relations between functions, its users and their tasks, while it has difficulty in understanding intuitively the relations among these three.

For solving such problems of existing expression methods, we devised “three layer model diagram” to describe how to understand a target system. It is the diagram build up from “system function layer”, “task and workflow layer” and “user practice field layer”. This diagram describes the understandings about a target system from viewpoint of system functions, tasks and their workflow, users in the field and relations among them. More precisely, the understandings of relations among system functions, tasks using the system, workflow of each task, a role that perform each work item, and system users in the field are described on this diagram. Figure 1 shows a case of “three layer model diagram” which describes understanding of a usability expert for order subsystem.

2.2 Method to Visualize Qualitative Difference of System Understanding

Using “three layer model diagram”, we describe how a software engineer and a usability expert understand the same system. That is, respective understandings of relations among system functions, tasks and each workflow, roles that perform each

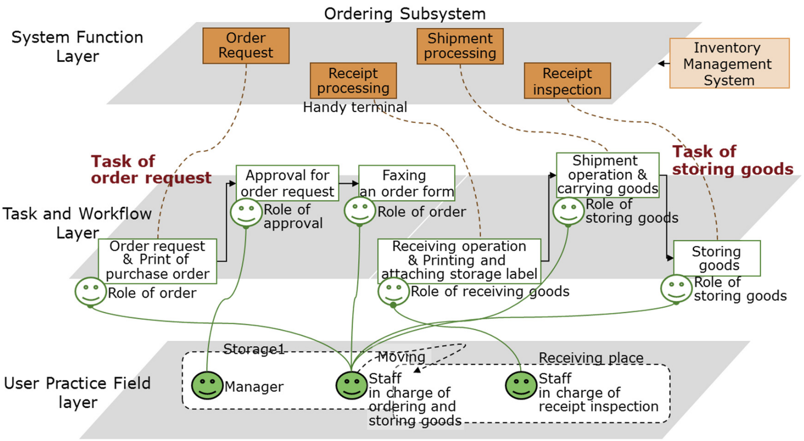


Fig. 1. Three layer model diagram which describes understanding of a usability expert for order subsystem

work item, and system users are described in this diagram. Figure 2 shows a case of "three layer model diagram" which describes understanding of a software engineer for the system identical with the case indicated on Fig. 1. These two diagrams, namely one describes system understanding of a software engineer and the other describes that of a usability expert, are compared and difference between them are extracted. Moreover the differences found in more than one software engineer in common are defined as visualized "qualitative differences of system understanding (a)" between them.

2.3 Method to Visualize Quantitative Difference of System Understanding

We calculate concordance rate between two kinds of "three layer model diagram", namely system understanding of a software engineer and that of a usability expert. Moreover this concordance rate is defined as visualized "quantitative difference of system understanding (b)" between them. The concordance rate is more specifically illustrated below using two cases of Figs. 1 and 2.

The three layer model diagram has elements (e.g. "Order request & Print of purchase order" on "task and workflow layer" in Fig. 1) and relation among elements on each layer. Relation among elements has two types which are order relation (e.g. relation between "Order request & Print of purchase order" and "Approval for order request" on "task and workflow layer" in Fig. 1) and inclusive relation (e.g. relation between "Order request & Print of purchase order" and "Task of order request" on "task and workflow layer" in Fig. 1). Besides, there is relation between elements on neighboring layers (e.g. relation between "Approval for order request" on "task and workflow layer" and "Manager" on "user practice field layer" in Fig. 1. This means relation between a work item and its performer.).

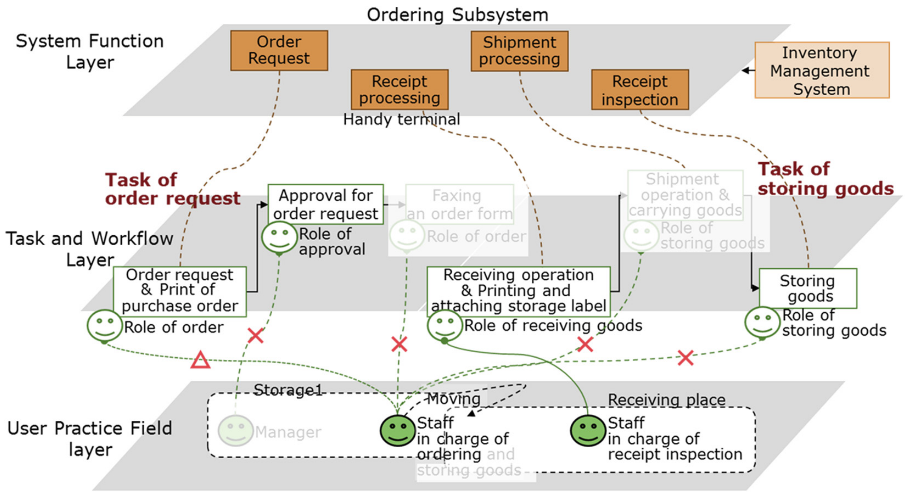
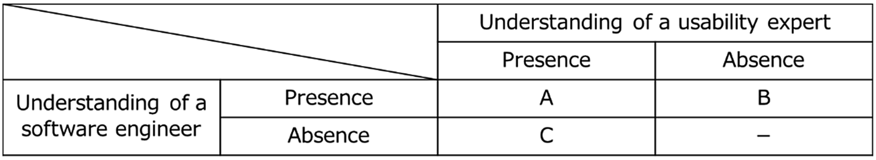


Fig. 2. Three layer model diagram which describes understanding of a software engineer for order subsystem

We compare Fig. 1 which describes understanding of a usability expert with Fig. 2 which describes understanding of a software engineer, focusing on presence and absence of these elements and relations among elements. Comparison patterns defined by presence and absence of these are organized in Table 1.

Table 1. Comparison patterns about an element and relation among elements



		Understanding of a usability expert	
		Presence	Absence
Understanding of a software engineer	Presence	A	B
	Absence	C	–

Meaning of each pattern is illustrated with specific examples in Figs. 1 and 2.

- Pattern A

This pattern means that understanding of a software engineer accords with that of a usability expert. “Order request & Print of purchase order” on “task and workflow layer” in Figs. 1 and 2 is a specific example of this pattern.

- Pattern B

This pattern means that some elements or relations in understanding of a software engineer are missing from that of a usability expert. Relation between “Approval for order request” on “task and workflow layer” and “Manager” on “user practice field layer” in Fig. 1 is a specific example of this pattern.

- Pattern C

This pattern means difference between understanding of a software engineer and that of a usability expert. "Staff in charge of ordering" on "user practice field layer" is a specific example of this pattern.

The number of elements and relations corresponding to each pattern is calculated. Next, ratio of the number of "pattern A" divided by the number of all patterns is calculated. This ratio is defined as concordance rate of system understandings between software engineer and usability expert. Specific math formula is given as follows.

$$\text{Concordance rate of system understandings} = \frac{A}{A + B + C}$$

2.4 Method to Visualize Difference in Recognition to System Understanding

Semi-structured interview is conducted for each of a software engineer and a usability expert to delve into their respective understanding for a target system, using the three layer model diagram which describes understanding of the usability expert. Protocol analysis for this interview is performed focusing on how they understand a target system. Specifically, verbal protocols related to considering order, priority and awareness scope for each layer of the diagram are extracted. These protocols are classified according to their main message. Besides, a label of short sentence representing its essence (e.g. "System function layer is considered first", "Confirming workflow and users together is kept in mind") is given to every classification. Relation between this label and utterer's profession (software engineer or usability expert), is analyzed. Moreover a difference about priority and conscious scope in "three layer model diagram" between them is visualized. This difference is defined as "difference in recognition to system understanding".

3 Experiment for Modeling

The observation experiment was conducted for ten software engineers (seven young engineers who had less than 10 years of practical experience in system development, and three expert engineers) and three usability experts. These research participants were required to try activities representing the approach of human centered design. The specific experiment task was a series of activities to understand relations between a task using the system and its performer (a system user). The activities were,

(1) Specify main task using a system.
(2) Specify workflow of each task and a role that perform each item.
(3) Specify system users and describe the roles of which each user takes charge.

These activities correspond to describe "task and workflow layer" and "user practice field layer" on "three layer model diagram".

The participants were offered the fill-in form made for these activities. They analyzed a system in charge now or in the past and filled out the form. After finishing the task, participants were given in-depth interview about the target system they analyzed, using the completed form. Moreover they were given semi-structured interview about impression and difficulty of the task.

Based on the verbal protocol of a participant in the interview and the completed form by a participant, two usability experts different from the experiment participants (including a experimenter) conducted the same analysis (above-mentioned activities of (1)(2)(3)) about the same system as a participant. Besides, they described the system understanding in "three layer model diagram".

The second semi-structured interview was conducted for the same participant half a year later by using this diagram which described a system understanding of the usability experts to delve into how the participant understand the target system.

The entire process of working on the experiment task and the interview were recorded as video data.

4 Analysis Procedure for Modeling

We correlated the analysis information by the experiment participant to the three layer model diagram which was described understanding of a usability expert for the same system. Besides, we specified points that analysis differed between a participant and usability expert. These points, namely, a difference of element and that of relation among elements (as described in Sect. 2.3) are described on the three layer model diagram.

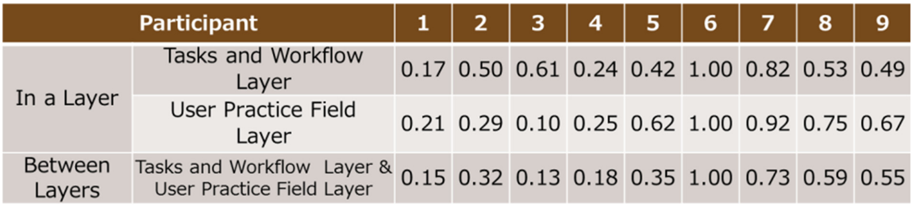
We compared these two diagrams, namely system understanding of a software engineer and that of a usability expert. The differences found in more than one participant in common were specified as "qualitative differences of system understanding (a)" between a participant and a usability expert.

Concordance rate between these diagrams was calculated as "quantitative difference of system understanding (b)" between them. We calculated this ratio about "task and workflow layer", that of "user practice field layer", and relation between these two layers. The subject of the ratio calculation was the analysis cases of nine participants who finished the second semi-structured interview (seven software engineers and two usability experts). Table 2 shows a list of calculated concordance rate.

Protocol analysis for interview video data of participants was performed focusing on how they understand a target system. We derived the kind of participant's recognition about priority and conscious scope in "three layer model diagram" through analysis. Moreover relation between a kind of recognition and utterer's profession (software engineer or usability expert) was analyzed. We specified this analysis as "difference in recognition to system understanding (c)".

Besids, we analyzed relation among above-mentioned "qualitative differences of system understanding (a)", the concordance rate as "quantitative difference of system understanding (b)", and "difference in recognition to system understanding (c)".

Table 2. Concordance rate of system understanding between an experiment participant and a usability expert



Participant		1	2	3	4	5	6	7	8	9
In a Layer	Tasks and Workflow Layer	0.17	0.50	0.61	0.24	0.42	1.00	0.82	0.53	0.49
	User Practice Field Layer	0.21	0.29	0.10	0.25	0.62	1.00	0.92	0.75	0.67
Between Layers	Tasks and Workflow Layer & User Practice Field Layer	0.15	0.32	0.13	0.18	0.35	1.00	0.73	0.59	0.55

5 Modeling the Features of How to Understand a System

It became clear that system understanding has patterns with a different feature from the following two perspectives.

- Understanding for entire system.
- Understanding for a layer of tasks and workflow.

5.1 Understanding for Entire System

We found that there are two types of understanding for an entire system. One is the understanding that makes the function a starting point, and the other is the under-standing that makes the user's tasks a center.

System Understanding that Makes the Function a Starting Point. This understanding is characterized by considering functions first and putting the highest priority on functions. Persons of such understanding consider by order of "system function layer", "task and workflow layer", and "user practice field layer" on "three layer model diagram", as shown in Table 3. In addition, they are hardly conscious of "user practice field layer", as shown in Tables 4 and 5. Participants who are presumed from their utterance to understand a system from functions are mostly correspond to those whose concordance rate about "user practice field layer" is less than 0.3. We consider that the low consideration for "user practice field layer" is a feature of system understanding that makes the function a starting point.

Table 3. Fragment of protocols about confirmation order in "three layer model diagram"

Participant A:	*First, I watched "task and workflow layer". But it's difficult to understand the description on the diagramr. (Omission) I confirm the diagram from here ("system function layer"), because it seems to be easier for me to understand it if I thought along the actual processing of the system. Then, I relatively can relate a function to workflow*

System Understanding that Makes the User and Their Tasks a Center. This understanding is characterized by putting the highest priority on system users, their

Table 4. Fragment of protocols about scope conscious in everyday work

Participant B:	*I'm rarely conscious about who is a user. I'm not also conscious about how the user works in the field and uses such functions. What I'm conscious about is basically only exchange of data between systems*

Table 5. Fragment of protocols about a "user practice field layer"

Participant C:	*I feel I can test better if I understood the layer ("user practice field layer"). But I have not been conscious about the layer. Still I can test*

tasks, and how they practice work in the field. Persons of such system understanding consider "task and workflow layer" and "user practice field layer" a center, on "three layer model diagram". Besides, they consider "system function layer" based on these two layers, as shown in Table 6. Thus, they are clearly conscious of "user practice field layer", as shown in Table 7. Participant who are presumed from their utterance to understand in the system users and their tasks center are mostly correspond to those whose concordance rate about "user practice field layer" is more than 0.7. We consider that the high consideration for "user practice field layer" is a feature of system understanding that makes the user and their tasks a center.

Table 6. Fragment of protocols about a layer which a participant usually consider first in "three layer model diagram"

Participant E:	*Considering from here (indicating "user practice field layer"), procedure of simultaneous message distribution (one of the function) was not appropriate. So I asked the project members to revise i.*

Table 7. 2nd fragment of protocols about a "user practice field layer"

Participant F:	*I always try to ask who use the system, even though I don't have a chance to contact with end users*

5.2 Understanding of Tasks on "Task and Workflow Layer"

Targets of understanding on "task and workflow layer" are the tasks using a system and workflow of each task. About the tasks, we found that there are two types of understanding. One is the understanding which corresponds to functions, and the other is that from the viewpoint of user practice in the field.

Figure 3 shows a case of "three layer model diagram" which describes understanding of a usability expert about customer information system for sales staff. In this diagram, sales operation utilizing the information provided by the system is defined as the main task. This task is carried out by sales staff who is a primary system user. Figure 4 shows a case of "three layer model diagram" which describes understanding of a participant about the same system. In Fig. 4, sales operation by sales staff is not defined as a task, although the same work items (e.g. "Confirmation of the list of

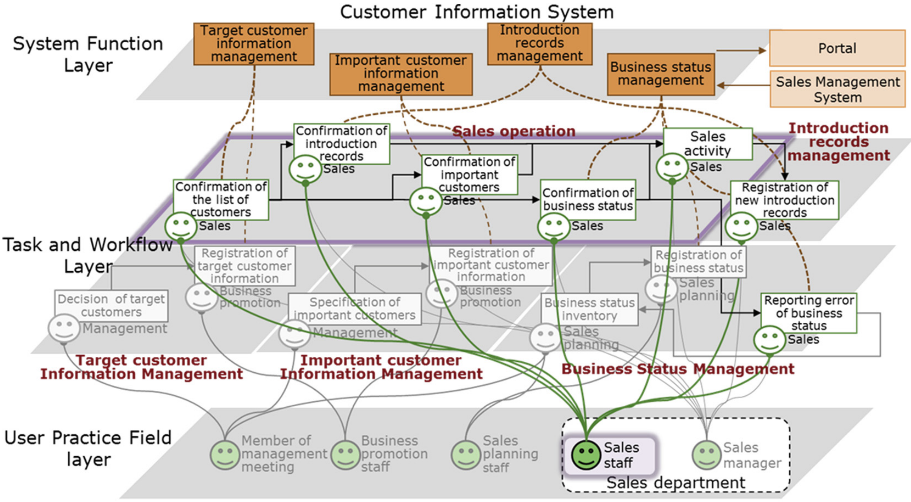

Customer Information System
System Function Layer
Target customer information management
Introduction records management
Important customer information management
Business status management
Portal
Sales Management System
Sales operation
Introduction records management
Confirmation of introduction records
Sales
Confirmation of important customers
Sales
Confirmation of the list of customers
Sales
Confirmation of business status
Sales
Sales activity
Sales
Registration of new introduction records
Sales
Task and Workflow Layer
Registration of target customer information
Business promotion
Registration of important customer information
Business promotion
Registration of business status
Sales planning
Decision of target customers
Management
Specification of important customers
Management
Business status inventory
Sales planning
Reporting error of business status
Sales
Target customer Information Management
Important customer Information Management
Business Status Management
User Practice Field layer
Member of management meeting
Business promotion staff
Sales planning staff
Sales staff
Sales manager
Sales department

Fig. 3. Three layer model diagram which describes understanding of a usability expert about customer information system for sales staff

customers", "confirmation of business status") is defined as those which sales staff carries out. Tasks defined by the participant are administrative task of various information provided by the system. These tasks correspond to the functions designed according to information type (functions for information life-cycle management) by one to one. The work item carried out by sales staff is located at the end of workflow in this information administrative task. We consider that the participant who analyzes the system of Fig. 4 understands tasks by corresponding to life-cycle management function of information itself, not from the view point of the information users.

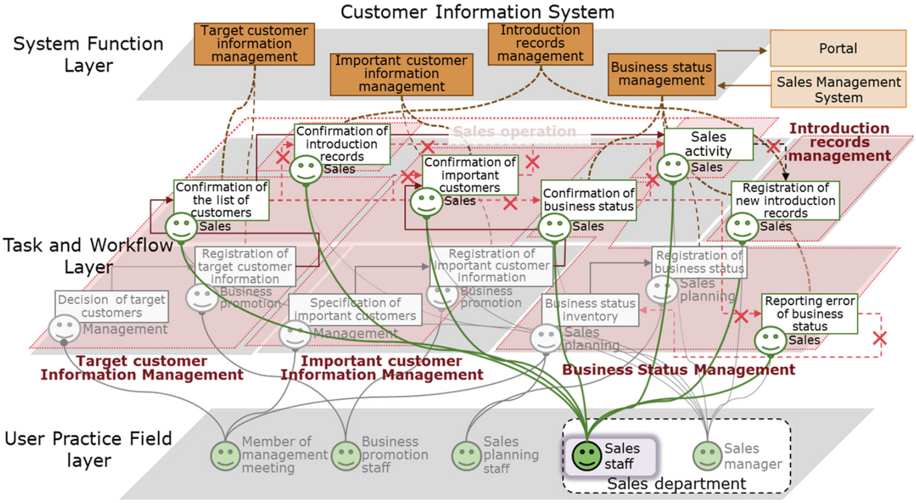

Customer Information System
System Function Layer
Target customer information management
Introduction records management
Important customer information management
Business status management
Portal
Sales Management System
Sales operation
Introduction records management
Confirmation of introduction records
Sales
Confirmation of important customers
Sales
Confirmation of the list of customers
Sales
Confirmation of business status
Sales
Sales activity
Sales
Registration of new introduction records
Sales
Task and Workflow Layer
Registration of target customer information
Business promotion
Registration of important customer information
Business promotion
Registration of business status
Sales planning
Decision of target customers
Management
Specification of important customers
Management
Business status inventory
Sales planning
Reporting error of business status
Sales
Target customer Information Management
Important customer Information Management
Business Status Management
User Practice Field layer
Member of management meeting
Business promotion staff
Sales planning staff
Sales staff
Sales manager
Sales department

Fig. 4. Three layer model diagram which describes understanding of a software engineer about customer information system for sales staff

The Defined Tasks Differ Between Two Types of Understanding. The gap between two types of understanding, namely the understanding which corresponds to functions and that from the viewpoint of user practice in the field, often derives difference of the defined tasks for the same system. Table 8 shows protocols about this gap.

Table 8. Fragment of interaction protocols about difference of tasks defined

Experimenter:	*They consider the task and its workflow that relates closely to functions for data processing in the system, do they?*
Participant U:	*Yes. About tasks relating to how to process as a system. But a task relates to here (indicating "user practice field layer"), so the task defined by them doesn't agree with ours very much*

Function Based Understanding Tends to Act on Dominance. We found the understanding described above in the three layer model diagrams of participants whose concordance rate about "user practice field layer" is comparatively high (those who understand in the system users and their tasks a center), in addition to participants of the low concordance rate (those who understand a system from functions). This suggests that a participant might give priority to function based understanding in thinking about tasks though they understand "user practice field layer". There were plural experiment participants in which the concordance rate of "task and workflow layer" and that of relation between "task and workflow layer" and "user practice field layer" are quite lower than that of "user practice field layer".

These suggests that function based understanding tends to act on dominance in a software engineer who is in charge of system development.

6 Conclusion

We focused on the cognitive features of software engineers and tried to model especially how software engineers understand the target system in the context of system development, applying the method for modeling we devised. This investigation revealed that there are two types of understanding for an entire system. One is the understanding that makes the function a starting point, and the other is the understanding that makes the user and their tasks a center. Moreover, it was found that the software engineers who understand the entire system based on functions are hardly aware of user practice in the field. We also found that there are two types of understanding for the tasks. One is the understanding which corresponds to the functions, and the other is the understanding from the viewpoint of user practice in the field.

Besides, it was revealed that software engineers who failed in design activities concerning usability have a tendency to understand tasks correspond to functions, in addition to understanding an entire system from functions. These suggested that their function-based understandings and little recognition to "system users" and "user practice in the field" in their everyday design activities are included in their failure reason.

From these, modeling how to understand the target system was found to be effective in specifying the reasons why software engineers fail in design activities concerning usability.

References

1. Norman, D.A.: The psychology of everyday things. Revised and Expanded Edition. Basic Books (2013)
2. IS9241-210: Human-centred design for interactive systems (2010)
3. Metzker, E., Offergeld, M.: An interdisciplinary approach for successfully integrating human-centered design methods into development processes practiced by industrial software development organizations. In: Nigay, L., Little, M. (eds.) EHCI 2001. LNCS, vol. 2254, pp. 19–33. Springer, Heidelberg (2001)
4. Nebe, K., Paelke, V.: Usability-engineering-requirements as a basis for the integration with software engineering. In: Jacko, J.A. (ed.) HCI International 2009, Part I. LNCS, vol. 5610, pp. 652–659. Springer, Heidelberg (2009)
5. Nielsen, J.: Heuristic evaluation. In: Nielsen, J., Mack, R.L. (eds.) Usability Inspection Methods. Wiley, New York (1994)
6. Palanque, P., et al.: A model-based approach for supporting engineering usability evaluation of interaction techniques. In: Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2011, pp. 21–30 (2011)
7. Propp, S., et al.: Integration of usability evaluation and model-based software development. Adv. Eng. Softw. **40**(12), 1223–1230 (2009)
8. Suzuki, S., et al.: Variation in importance of time-on-task with familiarity with mobile phone models. In: Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems, CHI 2011, pp. 2551–2554 (2011)
9. Fukuzumi, S., Ikegami, T., Okada, H.: Development of quantitative usability evaluation method. In: Jacko, J.A. (ed.) HCI International 2009, Part I. LNCS, vol. 5610, pp. 252–258. Springer, Heidelberg (2009)
10. Hiramatsu, T., et al.: Applying human-centered design process to system director enterprise development methodology. NEC Tech. J. **3**(2), 12–16 (2008)
11. Tanikawa, Y., Okubo, R., Fukuzumi, S.: Proposal of human-centered design process support environment for system design and development. In: Proceedings of the 4th Applied Human Factors and Ergonomics (AHFE) International Conference, pp. 7825–7834 (2012)
12. Seffah, A., Metzker, E.: The obstacles and myths of usability and software engineering. Commun. ACM **47**(12), 71–76 (2004)
13. Ferre, X.: Integration of usability techniques into the software development process. In: Proceedings of the ICSE Workshop on Bridging the Gaps between Software Engineering and Human–Computer Interaction, pp. 28–35 (2003)
14. Tanikawa, Y., Suzuki, H., Kato, H., Fukuzumi, S.: Problems in usability improvement activity by software engineers. In: Yamamoto, S. (ed.) HCI 2014, Part I. LNCS, vol. 8521, pp. 641–651. Springer, Heidelberg (2014)
15. Shackel, B.: Usability – context, framework, definition, design and evaluation. In: Shackel, B., Richardson, S. (eds.) Human Factors for Informatics Usability. Cambridge University Press, Cambridge (1991)
16. Hinds, P.: The curse of expertise: the effects of expertise and debiasing methods on predictions of novice performance. J. Exp. Psychol. Appl. **5**, 205–221 (1999)

17. Seffah, A., Gulliksen, J., Desmarais, M.C. (eds.): Human-Centered Software Engineering-Integrating Usability in the Software Development Lifecycle, vol. 8. Springer Science & Business Media, New York (2005)
18. Ambler, S.W.: The Object Primer 3rd Edition: Agile Modeling Driven Development with UML 2. Cambridge University Press (2004)