Infrastructure for Knowledge of Development in a Japanese ICT Company

Tomonori Yamashita, Kunio Shirahada, and Michitaka Kosaka
Japan Advanced Institute of Science and Technology, School of knowledge Science, Tokyo - JAPAN

***Abstract*--Further functions and enhanced performance requirements made to various information and communication technology (ICT) equipment have improved technical standards and the scale of development in design. Computer aided design (CAD) systems that support those development are becoming increasingly more complicated and extensive than they were several years ago. CAD system engineers have to learn not only software, but also hardware design knowledge to make CAD system. Engineers try to share and utilize knowledge on technologies that they obtain from designers who are participating in development. However, it is hard to share and utilize information due to the increasing fluidity of engineers who are involved in development due to diversified technologies and short-term development. The authors researched the sharing and utilization of development knowledge at an ICT development company that they worked at, and report the current situation that were found from interviews with engineers, and shows the consideration of the future knowledge management infrastructure.**

I. INTRODUCTION

The technologies for servers, networks, mobile communications, and other platforms that support IT have rapidly advanced almost on a daily basis due to technological innovations. Performance of the computer is improving exponential rates (Fig. 1). Mobile phones was only voice calls in the 1990s. However, today's mobile phones include e-mailing, Web browsing, photography, high definition resolution video, the capacity for over 10,000 songs and video games.

Fig. 1 Computing performance (TOP500[1])

The demands of customers and the market have been rapidly changing in recent years and the lifecycles of products on the ICT market have tended to shorten. The Ministry of Economy, Trade and Industry (METI) of Japan reported the lifecycle of popular products was over five years for 60% of all products in the 1970s, but it became only 5.6% in the 2000s, and products that had a lifecycle of under two years increased from 11% to over 50% (Fig. 2).

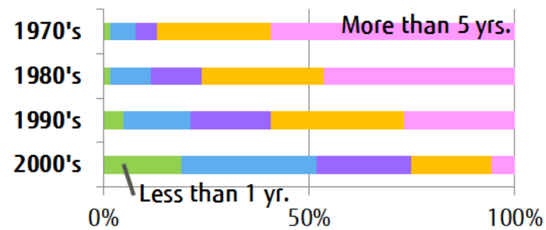

Fig. 2 Change of the product life cycle (METI)

There is a system that is called computer aided design (CAD) used to design such ICT equipment (
Fig. 3). CAD system has the characteristic that it covers wide variety of design such as electrical circuits and mechanical components and it uses complex algorithm including parallel calculation for rapid simulation, and it runs on the several kind of computers such as Windows, Linux, etc.

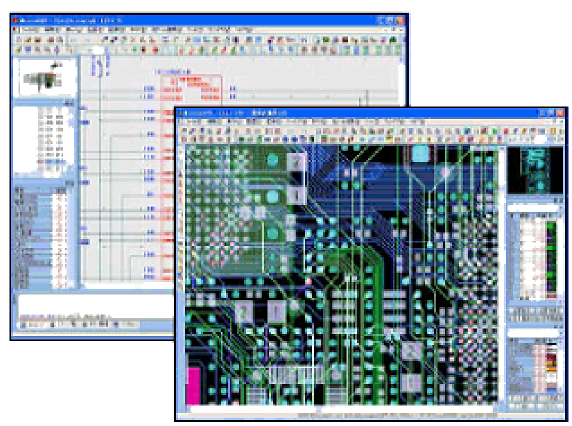

Fig. 3 CAD software (FUJITSU[2])

II. RESEARCH OBJECTIVE

Opportunities for CAD system engineers are increasing in many different kinds of fields in development due to the increase in scale, extensions, and the appearance of new services as cloud recently. CAD system engineers not only

[1] http://www.top500.org/statistics/perfdevel/

[2] http://www.fujitsu.com/downloads/MAG/vol43-1/paper10.pdf

have to understand software but also hardware technologies and operation environments. They may require advanced knowledge to be created, acquired, and transferred, which corresponds to accelerated lifecycles and rapidly advancing technology. Quality of the CAD system directly affect the products, therefore, CAD project managers are concerned about not only CAD system, but also renewed knowledge of CAD system engineers. Companies attempted various challenges by introducing more sophisticated development systems and shortening development periods. The utilization of knowledge management was one of these. There are a lot of research about knowledge management and systems, but it seems the problems are still existed. Therefore, we researched the present conditions and issues in knowledge management in CAD software department, and try to find ways for future knowledge management to assist engineering in the future environment.

A. Research questions

We posed a major research question and three subsidiary research questions.

Major Research Question:

What do we need to provide for technological development knowledge infrastructure for CAD system engineers?

Subsidiary Research Question 1:

Why is it difficult CAD development knowledge to utilize nowadays?

Subsidiary Research Question 2:

What development knowledge model is it now in CAD design?

Subsidiary Research Question 3:

What strategy is suited for development knowledge utilization in CAD design?

B. Research Method

We researched changes in circumstances surrounding development and related research on knowledge management and systems. We then interviewed CAD system engineers in the ICT company that the first author works at on sharing knowledge on development, utilizing it, and dealing with issues concerning it. Finally, we analyzed situations and considered answers to the research questions and future services in knowledge management. But, we just generated a hypothesis and do not evaluate it in this research.

III. RELATED RESEARCH

In this chapter, first we review knowledge and knowledge management model, and then we review knowledge management systems and services.

A. Knowledge model

In the research about knowledge growing in the company, it was concluded that the knowledge in an enterprise would grow and circulate with Socialization-Externalization-Combination-Internalization (SECI) and also would spread spirally in the space of achieved and recognized dimensions [10]. The concept of "Ba (Place)" was added into knowledge management [21]. "Ba" is the concept of a shared situation and environmental space in the process of knowledge creation. Toyama showed "autonomy, creative chaos, variety, redundancy, trust, and commitment" are factors of "Ba" tied to the creation of knowledge. In “Enabling of knowledge creation”, it was reported "mutual reliability, active sympathy, concrete support, tolerant judgments, and courage" were necessary for knowledge creation [23]. About factors, there was also the report that three more factors of "Permission of excursions into the unknown, consideration of other processes were added into, and original technical acquisitions" to Krogh’s five factors by analysis factors where knowledge was not offered at the company [18]. Classification of the knowledge transfer was reported into five kinds: Serial, Near, Far, Strategic, and Expert [5]. She found the model of knowledge transfer in a successful company would change from “one way style by experts” to an “alternative style between teams”. In “Knowledge management for teams and projects”, it was described that the knowledge was acquired, shared, and accumulated in a project by peer assistance when the project began, during peer reviews of the project, and retrospection after its completion [9]. It was mentioned the “knowledge cycle model” in which knowledge acquired by the project team was extracted as the best practice by the project crossing organization to become a corporate standard that was used again by the project team. About knowledge sharing strategy in a company, it was pointed out two strategies by Tierney. The first was called “codification”, which means knowledge is formalized to let users share it, and the second was called “personalization”, which enables the person who created knowledge to be contacted directly [20]. He suggested that we should execute them to raise the specific gravity of either depending on a corporate strategy. There was not only “codification and personalization” but also “hybridization”, which was an environment that unified real space in the office and virtual space on the Internet for knowledge creation [22].

Development processes and methods of quality control have been studied to increase quality and productivity since the 1960s. Huge systems such as mainframes in the 1960s were often systematically developed. A model to develop water flows, where water flowed from the top to the bottom was announced in such a circumstance [15]. There was another model of spiral development that simplified the analysis, design, implementation, and testing from start to finish of developments to make it easy to cope with changes in engineering announced after that [4]. A concept called “Agile development” that accepts changes in demand caused by the diversification of customers has been appeared around 2000, and it was explained in “An agile manifest” in 2001 [3].

B. Knowledge management systems

Now, let us move on to reviewing related knowledge management systems and services. It was proposed a system called “peer-to-peer system” that it provides a place to communicate with the person who had knowledge for the person who searches knowledge [6]. Investigation about "visualization of objective, visualization of knowledge and visualization of context" as success factors in knowledge offers was reported, and presented an example of groupware design in a "solution to the type of problem, to the type of ability improvement, and to the type of innovation" depending on a strategy of knowledge utilization [11]. It was also introduced a method where the knowledge offer and user sides that became "the interdependent relations" in a social network much more effectively obtained a wider variety of answers than the technique of one-sided questions when problems occurred [12]. There was the study about the relation of the internal motivation and knowledge offer. In this investigation, the motivation was improved by "autonomy, abilities, and technical relations", and clarified that this statistically affected technical knowledge offers [7]. There was the research about the action of members. A pattern to dynamically raise communities was derived by the action and communities were categorized based on this pattern to promote information sharing in this research. They extracted nine system requirements for collected intelligence, and suggested "a method of organizing content for information sharing systems" [2].

Shimazu likened knowledge management to Web 2.0 that Tim O'Reilly announced, and defined formalized knowledge that parties constructed as a knowledge base as KM1.0. KM1.5 automatically extracted knowledge from business applications and constructed data from various knowledge sources from blogs of the crowd were defined as KM2.0, and showed the way to a future knowledge management system [17]. He also concluded information disclosure causes new mindsets and the choice of information by users is tied to the selection of poor quality information and circulation of high quality information established by the competition principle from “Hatena Q&A Service” success case. There was the explanation that the evolution of the Web enhanced the characteristics of the location of much unspecified information that centered on individual exchanges and generated a knowledge community that did not distinguish information creators from receivers, and the activities of this community created innovations [8]. Value creation as the collaboration between enterprises from the aspect of services was studied and four steps of “sharing both collaborative people and purposes and conditions, defining service locations, creating new ideas, and implementing those ideas” was defined as the KIKI model [13]. They tried to adopt this model to the service business of energy saving company. Recently, research that formalized knowledge based on information where users do not take care is announced. A technique to acquire knowledge for classification by refining it in a system involving users in real time was suggested [19]. Problems including differences were obtained from records acquired from the logs or sensors of the system. Another suggestion, a system that automatically input knowledge by using a method of extracting characteristics from blogs, e-mails, or registration records and sending them to the file server was reported [1].

C. Knowledge systems on the market

There are numerous knowledge management systems on the market. Most systems of them are stock-type which store formalized knowledge into databases (DB). Groupware represents one kind of knowledge management system even though it is not called a knowledge management system. Groupware has functions such as "e-mail, electronic bulletin boards, and real-time conferences" that support communications between members, "document management" that makes DBs and stores documents in which ideas and design information are written, and a "scheduler" that shares schedules between members. A "work flow" function controls the flow of processing. Members of organizations transmit and share technological knowledge by using these functions. Google was established from the viewpoint of the “wisdom of crowds” by Larry Page and Sergey Brin in 1995 who constructed a mechanism where everyone was able to easily acquire information from all over the world with information presented by using high information-gathering capacities. Internet encyclopedia "Wikipedia" built by Jimmy Wales in 2001 whose content was enhanced by many unspecified people all over the world who supplied information through the Internet. A system that encourages knowledge to circulate by questioners making queries through the Internet has become general with the spread of Web 2.0 after 2000, and answers have been collected from around the world. About seven million questions and 23 million answers or more have accumulated on the "OKWave[3]" that OK Wave of Japan has managed through more than 140 QA sites in 2013, and these have been increasing daily. These knowledge systems are open to the public through the Web system, which has features that ensure the accuracy and validity of content due to the number of contributors and openness.

IV. CASE STUDY

In this chapter, we report changes in the CAD software development environment, information, knowledge, issues and comments on software development made by engineers in interviews we had with them.

A. Changes in software development

The computer-aided design (CAD) development organization that we researched began to develop CAD systems on mainframe from the early 1970. A mainframe was installed at its computer center and both ICT and CAD engineers could simultaneously use this computer. According

[3] http://okwave.jp/

to the records, CAD system was developed with FORTRAN language on proprietary operating system (OS). FORTRAN is designed for especially scientific calculation defined by the International Standards Organization. The mainframe stopped being used in the late 1990s, and it was replaced only three times in 20 years. OS of the mainframe was updated every time, but fundamental specification was not changed and FORTRAN was used consistently, too. Specification of FORTRAN was also updated three times in 1966, 1977, and 1990. We found they used to leave source codes as a common library that they utilized. In addition, we could confirm the development guidelines that had been printed on paper for development. However, we could not find records as to how far or where the guidelines had spread into their organization. According to the interview, they started to develop CAD system in the late 1980s on the Workstation (WS). WS was smaller but outperformed the mainframe and UNIX was used as the OS. UNIX was developed by Bell labs for WSs and had the characteristic of advantage of the network. The CAD system on WSs was developed with C language not FORTRAN. The reason why they changed from FORTRAN to C as a development language were 1)C language was a development language of UNIX, its affinity of operation with WSs was superior, 2)there was a global reference "the C Programming Language" which published by D. Ritchie [14]. Also, they said that they had problems sometime in the beginning of the introduction, but they supported one another if problems occurred, because that many CAD engineers had simultaneously learned the same language in the same environment. However, we could not confirm development guidelines or common libraries that had been developed in that organization. They had a PC that was used as a WS during the same period, but it was not used as a platform for CAD development because it did not have sufficient performance, but they began to develop from the late 1990s. They used C++ language that adopted object oriented programming into C language on Windows development. The WSs and PCs were connected through the company internal network in 1991 and enabled the use of e-mails. Web became available in 1994 there. Web was developed by CERN and was strong feature to share the information. They made static contents using only HTML first. But, Web technology was advancing rapidly, and several languages and techniques were proposed all over the world. Complying with this trend, they also developed web based information management system for engineering after 2000. They used HTML and JavaScript for screen layout, JAVA was used for logic programing, and SQL was used for database operation (Fig. 4).

With computer and network enhancing, they developed a grid computing system for high speed simulation that bundled up to several hundred PCs and also they used special computer called "Many cores" which had more than 500 CPU cores for parallel computing after 2010. A feature which presently attracted our attention, they used open source software (OSS) that volunteers around the world created and could be used freely. In the interview, they answered that they would use OSS recently for development to be shortened.

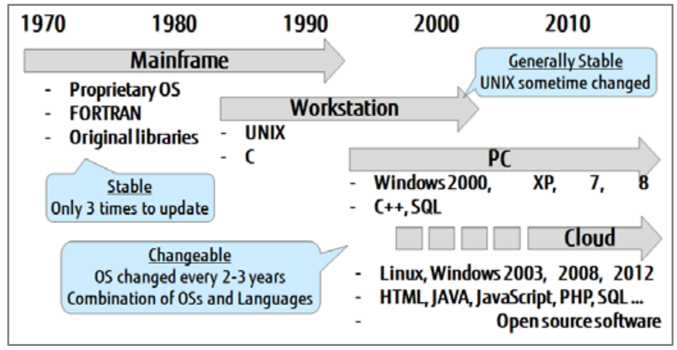

Fig. 4 Computers and languages in CAD

B. Knowledge on software development

This subsection presents the results from a questionnaire to CAD engineers to ask what and where they acquired knowledge in their development. Documents includes coded information such as Web sites and people includes meetings or seminars on following answers. By the result, they developed CAD software on the next steps.

START → Fundamental research → Marketing → Functional design → Architectural design → Coding → Verification → RELEASE

At the beginning development, they had meetings with ICT designer, and they acquired information and knowledge as specification and constraints of the product. CAD engineers think and review the necessary function of the design and user interface with related engineers next. So, they mainly acquired it from person in these steps. After detailed design such as architecture, coding and verification, they refer documents well, so the result shows they got information and knowledge from documents. (Fig. 5)

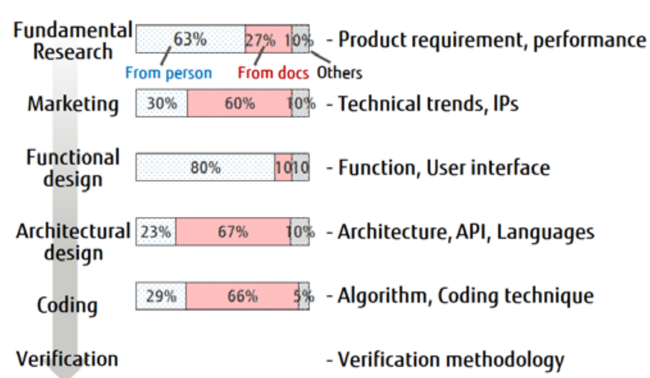

Fig. 5 Acquired info and knowledge

We show their notice when referring to information from outside the company. (

Fig. 6)

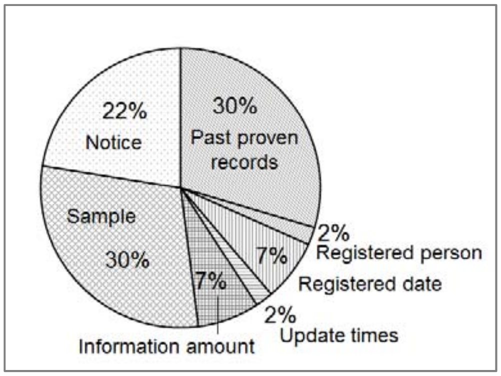

Fig. 6 Notes for referencing information

C. Management things in development process

The information during the project were managed with engineering information system, and this system could record following information and data.

- Documents or data that were input or output into or from the process
- Standards or rules that were defined to be performed within the process
- The schedule and progress of the project
- Occurred problems during the project.

The most characteristic function of this system was defining the input/output between processes and that output was then dealt with automatically and delivered to the target process. Once defined processes (like process flows) could be re-used as templates for other projects, and other projects could use them. We found simple work flow template which was defined in the organization, but it was too simple, so they had to add or change this template. On the other hand, changed work flow by the project was specialized and it was difficult to re-use for the other project. In addition, they couldn't open their work including designed products or information sometimes because of security. Developed source code was managed by open source software called Subversion to manage configurations. Problems that occurred during development were registered as "articles". Processes, registered output, source codes, and problems were linked and they could trace items between them. When problems and questions that occurred during development were registered, the system sent an e-mail to notify all the project members. The engineers who dealt with the problem that arrived coped with it by themselves. The development team had a fifteen-minute meeting every morning that tackled problems that had occurred or new requirements by ICT engineers. They shared situations that corresponded to problems. We asked CAD engineers the necessity of this daily meeting, they answered all the members in the project should share problems and had to confirm whether methods corresponded or not. They also had a monthly meeting with ICT engineers to discuss problems or requirements. Members in another project were made aware of problems during engineering with Post-It notes, which were stacked onto thick paper attached to the wall.(Fig. 7) Only key words were written on them. According to the engineer who placed them there, posting them on the wall made information easier to find than just saving it on a PC, and it promoted communications between engineers because they could easily figure things out. However, they experienced some disadvantage in that they needed time to electronically re-register information, some Post-Its did not update because of security matter. And, they also had to eliminate notes because of a lack of space so that it was difficult to understand the relations between Post-Its.

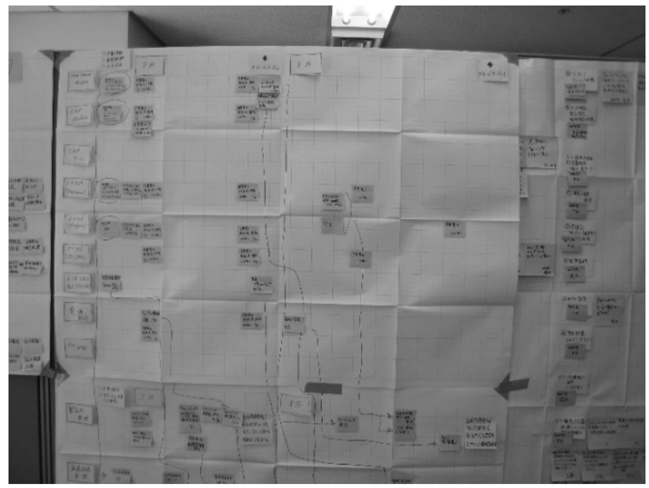

Fig. 7 Awareness board

D. Interview

Comments from the interviews with CAD engineers were as follows.

Q: What do you feel/think about development knowledge?

- As the timeframe for projects is short, there is not enough time to write technologies, know-how and even if wrote them, it would not be enough time for maintenance.
- Knowledge may not be able to be reused because technologies change so fast in particular Web developments. In addition, before knowledge utilization, it is hard to find knowledge which is able to utilize because too hard to guard the information not to leak outside the project. Sometimes, we cannot know what they make at the neighbor project.
- As architecture and languages are varied according to the CAD software, engineers may first be confused when participating in projects.
- Meeting is very important, and we have short meeting every morning to verify the situation.
- Big problems that influence other engineers are known to the others in the meeting, but small problems cannot be communicated sometimes because of not enough time.
- To share technological knowledge explained orally at meetings are easier than writing on the document, especially if there are fewer than 10 members.

- To understand how to use verification tools by the demonstration is better than the explanation because of many situations and contexts.
- A special knowledge such as parallel programming for simulation is too much complex and hard to understand, so very few engineer knows.
- Searching information or knowledge on the internet is easy, but technology is very rapid advancing, so it is difficult to choosing appropriate information.

Q: What do you feel/think about the development knowledge such as languages?
- They should only previously have known FORTRAN and C languages, but had to learn various languages and frameworks in recent system development.
- Recent technologies and knowledge greatly differ from system to system, so that it is difficult to reuse ideas, languages.
- Few people have mastered the new technologies and language, and it is difficult to find the person who has them when projects start or problems occur.

Q: What do you feel/think about software codes?
- OSS is changeable and offers may suddenly be stopped, so CAD engineers should notice carefully.
- Algorithms or sample source code can be found on the internet with Google, and these are sometimes used.
- It is necessary to note risks with OSS when problems occur as there are no guarantees.
- Reliability raises concerns because use results may not be provided. In addition, the number of downloads is not directly related to reliability.
- There is worry that coding techniques will lose by not writing codes.
- OSS is rapidly advancing, but obsolescence may also occur particularly early.

V. DISCUSSION

A. Knowledge utilization situation and strategy

We considered knowledge share and utilization strategy. Tierney advocated codification and personalization. Mainframe OS and FORTRAN language updated for approximately 8 years as reported in the previous section. That meant that the development environment were stable. When the same environment remained for a long time, development technology and knowledge might accumulate tacitly in individuals, but guidelines and codes remained. Why did they leave them? The following are thought to be reasons. Japanese ICT companies continued to have high growth over the late 1980s from the 1970s when systems were developed with mainframes, and development engineers were added to organizations as new members. It might have been necessary to educate many engineers together to accomplish high quality production. It would have been more effective to deliver guidelines that provided useful knowledge for development rather than teaching engineers individually in such situation. Further, as mainframes were proprietary products that information and knowledge might not have been opened. Reuse effectively ensures the best quality when planning to shorten man-hours. Therefore, they might have to extract parts from their sources or documents as guidelines and libraries. The environment was stable, product lifecycles were long, there was sufficient time to formalize individual knowledge, and they may have obtained greater advantages than those lost in supply and maintenance. We thought they formalized knowledge by such reason, and the environment was beneficial for the codification strategy. However, there is not as much time for engineers as there was before in the current environment from WSs to PCs because of rapidly advancing technology and accelerated development periods. Especially, the technology for Web development changes every 2–3 years according to Sato from Unisys [16]. Actually, we could know they used a lot of languages and technologies in development. It is natural to think that engineers in that situation feel these activities are useless because the formalized development knowledge that they learned cannot be utilized in the future. The motivation to formalize engineers' knowledge will not increase as they are told "we can find the information by searching on the Internet". There is another thought. Current Japanese ICT companies are losing growth due to the rise of neighboring countries in Asia, and the number of new engineers employed is decreasing. This may mean there are few engineer to whom they can transfer knowledge. In addition, the engineers' statement of "We don't pass on small problems and solutions to members because there are many problems in development and not enough time to do them" might make knowledge deviation potentially possible and is determined by individual judgments. Engineers in environments where technology changes sequentially in the short term refuse to formalize development knowledge, which is accumulated in individuals or organizations, because they have to immediately focus on the next project without any formalized knowledge. We can have the reason not to utilize knowledge from another point of view. In these days, it is easy to leak information by the advancing network system, so that they build firewall not to go out their properties including knowledge. In such situation, we thought knowledge would be innately transferred to engineers involved in project, and tacitly and invisible. This is the answer of our first subsidiary research question, and it shows this situation model on Fig. 8 which is the answer of the second subsidiary research question.

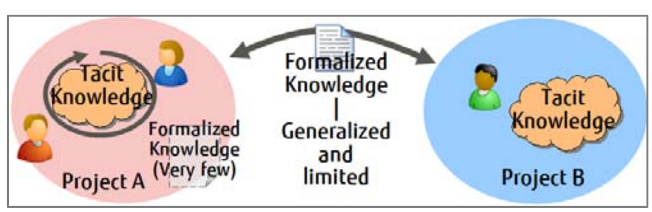

Fig. 8 Knowledge circulation in organization

Tierney advocated a personalization strategy for individual knowledge to be utilized. However, we cannot state they should undertake a strategy of personalization in the current situation because they cannot ensure that members who have knowledge will remain in the same projects or organizations by accelerating development. A member with knowledge leaves due to various circumstances might not be able to pass on information to the other members, or they can't ask the engineer who has knowledge after leaving when they want to know. They have recently been not able to specify or find appropriate people from interviewees who have stated "There are large differences between systems and these are difficult to understand if development is not involved". Therefore, if we use this strategy, we have to keep the person who has knowledge. Because personalization strategy is based on the premise that we know the person who has knowledge, and On the other hand, opinions were solicited whether "algorithms and source code samples could be found in example on Google", and there is coded information on the internet even if there was no coded information and formalized knowledge in the company. In these days, searching technology is advancing and it is easier to find necessary information than before. And we thought we could find who accessed information that was discovered from operation histories or current logs using current tracking technology. Though it is hypothesis, it is possible that the person who accessed information outside has its knowledge after access, and if we codify this information we can both information and information of the person who has information or knowledge. As we explained, it is difficult to expect that engineers leave formalized knowledge nowadays, engineers should leave footprint if they try to find information or knowledge, and we should pursue this information. At this time, we have to think something from words "Easy searching, difficult selecting". Also, they said that understanding how to use verification tools because of many contexts in the interview. We guess obtained information has less contexts and situation, so we think we should accumulate footprint with reason why they search. Important thing is who, where and why access the information. It is similar codification strategy, but it is not only codification but also personalization by tracing footprints. This is the answers of the third subsidiary question.

B. Required knowledge infrastructure

Let us now consider development knowledge infrastructure from problem management. CAD engineers take action to think about solutions, and present these after having confirmed their effects when questions from ICT and other engineers arrive in problem tracking systems. If they cannot find solutions they can adopt, they must try to find other solutions. This action creates various knowledge with or without effects. As problems are registered in the system, they can record what they did to solve them, where they found them, who they cooperated with, and how many solutions they tried. Now, let us move onto a discussion on the value in use the knowledge. There is a structure that enables voting for contributed content in SNS by using "Good and poor" results. If they put this mechanism into their tracking database, it is difficult to measure the value of knowledge using this though, but it is possible that they know what problem and solution is popular. Unfortunately, we could not check what kind of problem data and solutions they record, but we knew there was no voting and display of reference count. If they open their issue tracking database to other engineers including ICT engineers and measure access information. Interested parties in the problem have the possibility of finding answers to it. Further, a problem that is referred to a lot of times may involve important technology and provide valuable knowledge. "Good" or "Poor" is just accumulated from the past in general SNS. So, "Good" was "Good", but it is not possible to say "Good" is "Good" when engineers refer it clearly, because ICT is advancing and rapidly changing. We think it may enable us to determine by analyzing access logs or recorded "Good or poor" counts even if "Good" was not recorded explicitly. We think evaluation of knowledge may affected by votes, references and context. And utilization is influenced by this evaluation. So, we think the visibility of the evaluation of the information at first and provide database base on new codification strategy. We currently need to do more research on effects to achieve such a system and mechanism in future work. This is the answer of the major research question.

VI. CONCLUSION

In this research, we clarified the reason why knowledge could not become to utilize recently by the situation in past and present in the CAD system development in a Japanese company. In the current development situation and circumstance involving highly advancing technology and characteristic of the projects, software engineers cannot formalized their knowledge that acquired in design. In addition, an Internet and advancing of the searching technology and security of the internal properties make engineers tend to obtain necessary knowledge from outside to make up for their lack of knowledge. Especially, to shorten development period, they use OSS by reason that is easy to search and useful. Advancing technology makes diversification of the customer requirement and it makes to shorten development period. Shorten development period makes fluidity of engineers who are involved in development project due to diversified technologies and the reshuffling members. Therefore, it hard to leave know-how and their own libraries. It is hard to take codification and personalization strategy, however, we think advancing technology gives another solution. We made hypothesis that the future knowledge management infrastructure would build by sensing, collecting, and searching. At this moment, we could not build the infrastructure with securities based on Know-where strategy, but we are going to setup their environment with this kind of service and research an effect and the influence of their production in the practical development.

REFERENCES

[1] Aoki, W. “A contents organization method for information sharing systems” Kochi University of Technology Library, http://www.kochi-tech.ac.jp/library/ron/2009/g18/g18.html, 2010

[2] Arise, K. “A sharing Engine of Knowledge Unions” The Institute of electronics, information and communication engineers, *Technical report of IEICE*, 0IS2004-102, pp. 43-47, 2005

[3] Beck, K. M, Beedle A, van Bennekum A, Cockburn W, Cunningham M,Fowler et al. "Manifesto for Agile Software Development" Agile Alliance, 2001

[4] Boehm, B. ”A spiral model of software development and enhancement” *ACM Sigsoft software engineering notes*, vol. 11, pp. 22-41, 1988

[5] Dixon, N. *”Common knowledge: how companies thrive by sharing what they know”* Harvard Business Press, 2000

[6] Hisano, T. “Discussion on Peer-to-Peer Oriented Knowledge Management System” *The Japan Society for Management Information Proceeding*, 2002 spring

[7] Horie, T. Atsushi, I. and Yasuo, I. “Relation between Knowledge Contribution and Intrinsic Motivation within An R&D Organization” *Japanese Journal of Administrative Science*, vol. 20, No.1, pp.1-12, 2007

[8] Inoue, S. “Web 2.0 jidai no kigyou keiei” Nomura Research Institute, *Chiteki Shisan Souzou*, Jan.2007

[9] Milton, N. *“Knowledge Management: For Teams and Projects”* Oxford, 2005

[10] Nonaka, I. and H, Takeuchi *“The knowledge-Creating Company”* Oxford, 1995

[11] Nomura, T. “Strategic Knowledge Pattern: Guideline for Designing Groupware to Generate Knowledge-creating `Ba’” *Information and Processing Society of Japan*, vol. 45, No. 1, pp. 121-130, 2004

[12] Nomura, T. T, Katayama K, Saito and K, Okada “Potential Social Network Search Mechanism for Driving Knowledge Sharing” *Information and Processing Society of Japan*, vol. 46, No. Sig 8, 2005

[13] Qi, Z. M, Kosaka K, Shirahada and T, Yabutani “A proposal of B to B Collaboration Process Model based on a Concept of Sevice and its Application to Energy Saving Service Business” *IEEJ Transactions on Electronics, Information and Systems*, vol. 132, No. 6, pp. 1035-1040, 2011

[14] Ritchie, D. and Kernighan, B. *“The C Programming Language”* Prentice-Hall, 1978

[15] Royce.W. ”Managing the development of large software systems” *Proceedings IEEE WESCON*, pp.1-9, August 1970

[16] Sato, K “History of Web Technologies” *Unisys Technology Review*, No. 110, Nov. 2011

[17] Shimazu,H. and Shinichi, K. “KM Saikou:Web2.0 jidai no knowledge management” *Information processing society of Japan*, vol. 47, No. 7, p. 768-774, 2006

[18] Taguchi, T. “Naze chishiki no teikyou ga okonawarenai noka?” *The Japan Society for Science Policy and Reseach Management Proceeding*, 20(2), pp. 867-870, 2005

[19] Takenaka, T. “Classification Knowledge Acquisition Method for a Group by Presenting the Issues”, *Information processing society of Japan*, vol. 62, No. 1, pp. 231-250, Jan.2006

[20] Tierney, T. N, Nohria and T, H. Hansen “What’s Your Strategy for Managing Knowledge?”, *Harvard Business Review*, vol. 77, No. 2, pp. 106-116, 1999

[21] Toyama, R. and I, Nonaka “What Is a Good Ba?-- The Role of Leadership in Organizational Knowledge Creation”, *Hitotsubashi business review*, issue 48 pp. 4-17, 2000

[22] Umemoto, K. “Knowledge management: Saikin no rikai to doukou”, *Information science and technology association*, issue 62, vol. 7, pp. 276-280, 2012

[23] Von Krogh, G *“Enabling knowledge creation”* Oxford press, 2000