# A DESIGN PROCESS MANAGEMENT METHOD FOR PRODUCT-SERVICE SYSTEMS

S. Minato, Y. Idei, Y. Mitake and Y. Shimomura

## Abstract

Product-Service Systems (PSS) has been regarded as an attractive business concept that create high added value by integrated provisions of products and services. Since both products and services are included in the design object, the PSS design process has become increasingly complicated. To support such PSS design process, this paper proposes a design process management method for PSS. The proposed method uses "PSS design cards" which organize the tasks in PSS design and enables designers to grapple what they need to focus on during PSS design process in a tangible way.

*Keywords: product-service systems (PSS), design process, design management, design teams*

## 1. Introduction

With economic globalisation, many companies struggle to make a profit because of increasing number of the competitors in their business. Manufacturers are finding it difficult to increase their profit by selling only products because many companies lower their prices using cheaper labour (Neely, 2009). To generate a higher sustainable profit, product-service systems (PSS) have begun to attract attention as an option for income generation. In general, PSS is a business model characterised by the integration of products and services (Baines et al., 2007). In PSS, sustainable relationships with customers can be built by providing services in the entire product life cycle. Therefore, it is expected as a new way to differentiate it from the traditional business models focusing on the "hit and run" method of products. However, PSS has features that differ from the traditional business model, and few studies have been done on the PSS design process. Thus, the tasks companies should perform for PSS design are not clear. This prevents them from realising successful PSS. To overcome this situation, it is necessary to organise the tasks in PSS design and to manage the progress accurately.

In this paper, the authors propose a design process management method using "PSS design cards". PSS design cards systematically organise tasks in PSS design into cards based on the SEMAT (Jacobson et al., 2013) that manages software development based on tasks. PSS design cards make it possible to visualise and manage the progress of the PSS design process in a tangible (touchable) way.

## 2. Existing studies

### 2.1. SEMAT

Software development is a complex and risky business, as various stakeholders are involved in the process. Therefore, a common ground for software development is needed to improve the process. With this background, Software Engineering Methods and Theory (SEMAT) (Jacobson et al., 2013) provides a theoretical software development framework, Software Engineering Methods and Theory (SEMAT) (Jacobson et al., 2013) was developed. In SEMAT, software developers bring forward software

development using SEMAT kernel cards (Figure 1) that consist of alphas, alpha states and checklists. The alphas are characterised by a simple set of states that represent their progress. Namely, this is a viewpoint that the software developers should monitor in software development. The alpha states indicate the progress status of the development in each alpha. And a checklist is a summary of tasks for each alpha state. The SEMAT kernel cards are constructed with the above items.
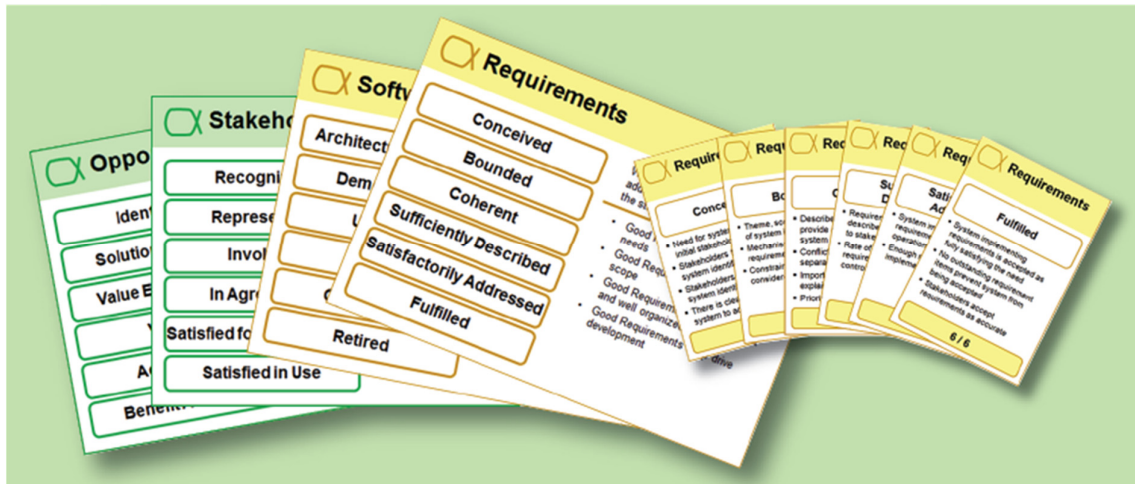


**Figure 1.  SEMAT kernel cards (Jacobson et al., 2013)**

The SEMAT kernel was constructed according to three principles: actionable, extensible and practical. These principles bring out three unique features of the SEMAT kernel, and these make it possible to work as a common foundation for software development.

First, the kernel handles the routine work. These routine works are captured as alphas rather than work products (such as documents). In software development, developers must do many tasks in a balanced manner. For that, the kernel is an effective thinking framework that supports software development not as a linear process, but rather as a network of collaborating elements that should be balanced. By identifying the current state of each alpha, the kernel becomes actionable and enables it to guide the behaviour of software developers.

Second, the kernel can be extended to support different projects. This is because SEMAT is built as a common foundation for software development, so it can be used regardless of the type of software. Also, The SEMAT has a specific approach as a common foundation. Therefore, the SEMAT can be used with other software development methods.

Third, the kernel is a hands-on, tangible thinking framework that supports software developers to carry out their work. This is the most important feature of the kernel to work as a common foundation. For example, the kernel can be touched and used through cards. Team members can keep the kernel as a small deck of cards in their pockets, which they can easily pull out to discuss the current state of development and the work assignment and collaboration among team members. Team members can also discuss areas of improvement by referring to the cards. Thus, the kernel is not merely a heavyweight description of team tasks. Rather, it forms an essential part of what they are doing each day.

The SEMAT was constructed according to the following procedure (Jacobson, 2012). First, the authors arranged "things we always work with" (what to handle) (Figure 2) and "things we always do" (what to do) (Figure 3) in software development based on the survey of the software development method. Here, the authors defined alphas from what to handle and defined alpha states and checklists from what to do. Next, the authors confirmed both what to handle and what to do and classified what to do according to the alpha. Finally, a checklist was organised in the form of development tasks, and the authors integrated alphas, alpha states and checklists as the SEMAT kernel card.
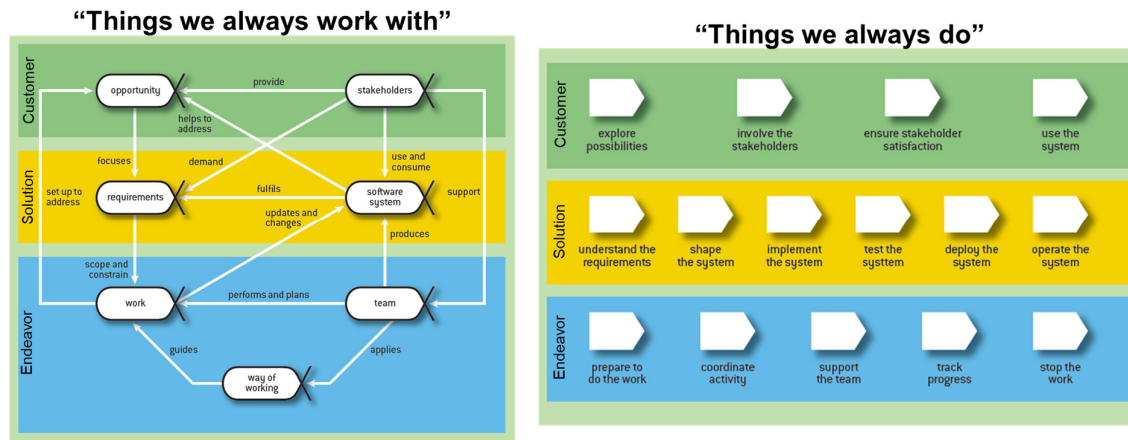
**Figure 2. "Things we always work with" and "Things we always do" (Jacobson et al., 2013)**

## 2.2. PSS design methods

A number of researchers have proposed a design method for PSS. Based on business model domains, Ostaeyen et al. (2011) proposed a method for PSS option generation. Meier et al. (2011) developed a PSS resource planning method using heuristic optimisation. Rese et al. (2012) proposed an ontology-based PSS business model design method, and Wiesner et al. (2013) proposed a strategy for designing a PSS business model using a business model canvas.

There are many studies that independently support a stage of PSS design, however, there are few studies that continuously support more stages of PSS design. The project of "PROTEUS" proposes practical PSS design and development process, however, since they focus on the specific area of the maritime, PROTEUS is not sufficient for practical design. As described above, a method that can be a common foundation that can continuously support more stages of PSS design is required in order to support companies to shift to PSS providers. In particular, what the company should do in the PSS design process has not yet been clarified. Therefore, it is difficult for companies to manage the process of PSS design in the absence of evaluation criteria of the design process. Furthermore, it is also a problem that the bottleneck which companies should overcome when shifting to PSS providers is not clarified.

## 2.3. Approach of this study

There are two different approaches in software development: a process-oriented approach called waterfall development and a state-oriented approach called agile development. SEMAT takes the state-oriented approach that manages the development process by evaluating the state with deliverables. In PSS design, it is pointed out that it is necessary to iterate the PSS design process and to perform improved design due to the complexity of its design process (Akasaka et al., 2013). In this respect, the state-oriented approach of SEMAT seems to be suitable for the iterance of the PSS design process and improved design.

Furthermore, SEMAT is a thinking framework that forms the basis of daily work as a common foundation by a specific approach. Therefore, it is usually assumed to be used together with other software development methods such as Scrum (Schwaber, 1997), Extreme programming (Paulk, 2001) and Use case driven object modeling (Rosenberg et al. 2007). In this respect, there is no study other than SEMAT to be a common foundation for software development. In PSS, there is a need for not only methods that independently support a stage of PSS design but also a common foundation to support their use.

For these reasons, this paper proposes a common foundation to support PSS design by extending SEMAT. Specifically, it clearly shows the tasks to be addressed during the PSS design process, and makes it possible to manage the PSS design process.

## 3. Design process management using PSS design cards

This paper proposes a method to manage the PSS design process using PSS design cards. This proposed method contributes to (1) analysing the current state of the design process, (2) setting the next target of the design process and (3) setting up the next team tasks. The proposed method was constructed in three steps in Figure 3. First, the authors construct the PSS design kernel with reference to SEMAT. Second, PSS design cards are developed based on the PSS design kernel. Third, the design process management of PSS is performed using the developed PSS design cards. The first and second steps are performed to develop tools for design process management, and in the third step, design process management is performed cyclically by a discussion among the design team. This chapter explains the proposed method following these steps.
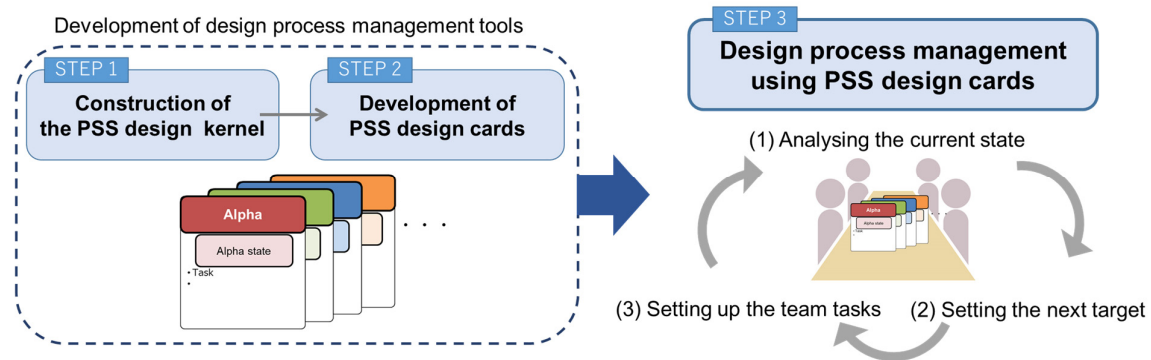


**Figure 3. Outline of the proposed method**

## 3.1. Construction of the PSS design kernel

As a preliminary step, the PSS design kernel was constructed according to the construction procedure of SEMAT. Specifically, this study constructed the PSS design kernel via the following procedure.

### 3.1.1. Organising "what to handle" and "what to do"

First, this study organised "what to handle" and "what to do" in the PSS design. In this way, the authors identify the subjects to work with in the management of the design process. In SEMAT, these are organised in three layers: "customer", "solution" and "endeavour". The layers of "customer" and "solution" describe the design object, indicating what software designers should tackle. On the other hand, "endeavour" describes the design subject, meaning how designers should perform. As the first step, this study focuses on "customer" and "solution" layers and clarifies the PSS design object.

Regarding "what to handle", the viewpoint of the "software system" in SEMAT was interpreted as "products and services" in the PSS design (Figure 4). Regarding "what to do", the authors decided that the PSS design process follows the same process as SEMAT.
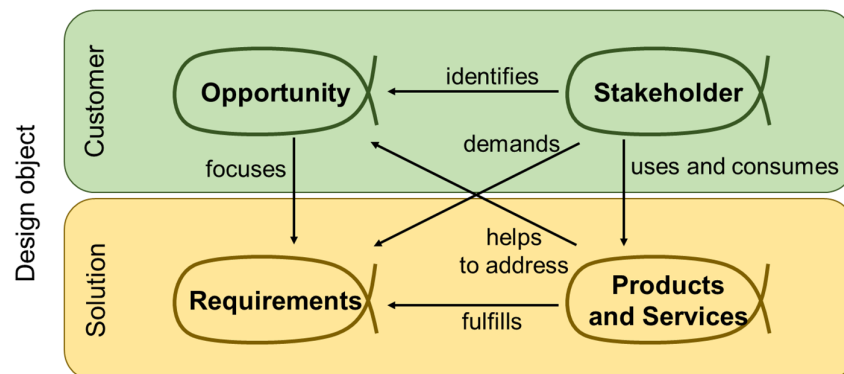


**Figure 4. "What to handle" in the PSS design**

*3.1.2. Literature survey on PSS*

In order to ensure the practicality of the proposed method, a literature survey on PSS was performed. The object of the survey were 78 articles containing the phrase "product-service systems" from the articles registered in the literature database (Web of Science) provided by Thomson Reuters. In the literature survey, the authors comprehensively identified and organized PSS based on the following point.

- Points that are important in PSS design
- Design process of PSS
- Milestones of PSS design
- Successful cases of PSS and its success factors
- Important matters in introducing and operating PSS

*3.1.3. Clarifying the difference between the software development process and the PSS design process*

In order to ensure the generality of the proposed method, the authors abstracted the points to be noted in PSS design identified by the PSS literature survey, and interpreted these as general ones. Then, these were organised using the KJ method (Kawakita, 1967) and these compared the software development process with the PSS design process. Based on these results, this study interpreted existing alphas, alpha states and tasks of SEMAT as new alphas, alpha states and tasks corresponding to PSS. As a result, the alpha states "actor network selected" and "continuous improvement" in the alpha of "products and services" were newly extended especially as a point unique to PSS.

"Actor network selected" was added considering the point that the structure of an actor network plays an important role in PSS (Morelli, 2006). The process of providing PSSs must include not only the stage of using products or services, but also customer activities. Therefore, new actors are often involved in the network.

"Continuous improvement" was added considering that it is useful to manage and reuse design knowledge gained through the PSS design. This is called continuous improvement in the PSS life cycle (Akasaka et al., 2013). In particular, in service design, it is necessary to iterate the cycle of design and improvement; therefore, it is necessary to iterate the cycle of design and improvement even in the PSS design.

## 3.2. Development of PSS design cards

After the construction of the PSS design kernel, PSS design cards were developed based on the PSS design kernel constructed in the previous procedure. Here, this study summarised the tasks as a checklist for each alpha state and organised it together with alphas and alpha states into the PSS design cards.

Tables 1–4 show components of the PSS design cards in each alpha. The cards consist of three items, alphas, alpha states and checklists. An alpha is the viewpoint of PSS design and consists of all four types of alphas: "opportunity", "stakeholder", "requirements" and "products and services". Each alpha has six alpha states. Each alpha state indicates the progress state of the alpha in PSS design, and the name of the alpha state indicates the goal to be reached. These alpha states are constructed according to "what to do" in the previous procedure: the first step of Chapter 3.1, and the "what to do" indicates the order of the basic design process when not considering the iteration of the design process. A checklist is a task list of the alpha state, and it becomes a specific evaluation criterion for judging the arrival at each alpha state.

The alpha of "stakeholder" indicates the state of stakeholders involved in PSS design and PSS operation. In this viewpoint, first, designers identify the stakeholders involved in designing a PSS and clarify their roles. Next, they develop a communication and hearing agreement to build good cooperative relationships with stakeholders. After the outline of the system is determined and an agreement is reached among stakeholders, designers confirm the resources and supply plan needed to develop the system. After the system is introduced, designers evaluate whether the stakeholders can accept the designed PSS, and it meets the minimum needs and expectations of PSS operation.

The alpha of "opportunity" indicates achievement items for maintaining an appropriate state to develop or improve the PSS provided by the stakeholders. In this viewpoint, after designers determine the solution by identifying the problem to solve with the PSS, they decide upon the value to be realised. Next, designers comprehensively evaluate the necessary processes and resources to realise the solution and constraints and risks of PSS operation. After the introduction of PSS, designers evaluate whether the problem is actually solved, whether profits assumed in PSS design is obtained and whether operational problems occur.

**Table 1. Components of the PSS design cards (Stakeholder)**

| Alpha | Stakeholder | | |
|---|---|---|---|
| **Alpha state** | ①Recognized | ②Represented | ③Involved |
| **Checklist** | • Agents on Flow model (potential provider, receiver, relay agent)have been identified | • A planner (mediator between agents) have been assigned | • Agents have played their role |
| | • Representatives of all agent have been agreed | • A planner have agreed own functions or scopes | • Agents have respective feedback and involve decision-making |
| | • Functions or scopes carried out by each agent have been defined | • How to collaborate between agents (how to realize the function) has been agreed | • Communications between agents have been well |
| | | • Agents respect each effort | |
| **Alpha state** | ④In Agreement | ⑤Satisfied for Deployment | ⑥Satisfied in Use |
| **Checklist** | • Value for each agents have been defined and other agents have agreed it | • Feedbacks have been provided to the entire system from each agent's point of view | • A system has been exceeded minimum expected value of agents |
| | • All agent have been agreed functions that will be realized with the priority | • A ready to operate the system have been confirmed | • Needs and expectations of the agents are satisfied |
| | • All agent have been agreed minimum expected value | | |

**Table 2. Components of the PSS design cards (Opportunity)**

| Alpha | Opportunity | | |
|---|---|---|---|
| **Alpha state** | ①Identified | ②Solution needed | ③Value established |
| **Checklist** | • Receiver's activities that can be supported by solutions of products and services have been identified | • Needs for solutions of products and services have been promised | • Value-in-use have been defined when a solution succeeded |
| | • Receivers have grasp the potential value and have wishes to invest | • Receiver's demand have been identified | • Receivers have understand the secondary effects of solutions |
| | • Agents that share the supportable receiver's activities have been identified | • Potential problems and root causes have been found | • Receivers have understand Value-in-exchange of products and services |
| | | • At least one solution of products and services have proposed | |
| **Alpha state** | ④Viable | ⑤Addressed | ⑥Benefit accrued |
| **Checklist** | • Main resource and process of products and services have been described | • A demonstrated solution have been provided | • Obvious benefits in the operation have been created |
| | • Constraints when a solution was launched and deployed have been cleared | • Effective systems have been available | • Predictable investment effects have been obtained |
| | • Risks have been under control | • Receivers have agreed to the provide function | |
| | | • Receivers have been satisfied the solution | |

The alpha of "requirements" indicates the tasks necessary for satisfying the stakeholders' needs and taking care of the opportunities. In this viewpoint, designers first clarify the required functions and scope of the PSS. Next, they confirm whether the required functions of the PSS meet the receiver's requirements and whether a problem occurs when developing the system. After introducing the system, designers investigate whether the required functions have been realised and there is a task in the system operation.

The alpha of "products and services" indicates a structure to realise the required functions clarified in "requirements". In this viewpoint, first, designers consider which products and services are suitable to realise the required functions and construct a network of actors that provide those products and services.

Next, designers confirm whether the actor network and the products and services can operate accurately as a system by evaluating it. Then, after reaching an agreement with the stakeholders, designers start to operate products and services. During the operation, designers monitor whether products and services are operated in an appropriate environment and whether the required functions are maintained.

**Table 3.  Components of the PSS design cards (Requirements)**

| Alpha | Requirements | | |
|---|---|---|---|
| **Alpha state** | **①Conceived** | **②Bounded** | **③Coherent** |
| **Checklist** | • Requirements and functions of the system have been clear | • The range of requirements and functions that should be implemented have been agreed | • The overall picture of the PSS have been shared to the agents |
| | • Users have been identified | • The Criteria for succession (KPIs) have been clear | • The critical usage scenario have been shared |
| | • First capital investor have been identified | • The change management of requirements and functions have been agreed | • The priority of requirements and functions have been clear |
| | | • The non-functional requirements have been identified | • Collision of requirements and functions have been eliminated |
| | | | • The effect of requirements and functions have been understand |
| **Alpha state** | **④Acceptable** | **⑤Addressed** | **⑥Fulfilled** |
| **Checklist** | • Acceptable solutions for agents have been provided | • Required functions and sufficient for system acceptance are implemented | • The system have  satisfied the requirements and needs |
| | • The agreed requirements and functions have had low probability to change | • Agents have agreed the system can be operated | • There have  been no unsolved requirement to interfere with the completion |
| | • Value have been clear | | |

**Table 4.  Components of the PSS design cards (Products and Services)**

| Alpha | Products and Services | | |
|---|---|---|---|
| **Alpha state** | **①Actor network selected** | **②Demonstrable** | **③Usable** |
| **Checklist** | • The products and services architecture that can cope with important risks is adopted | • Features of actor network are verified by using simulations or reefing prior case | • Products and services have been available and  the required quality attributes have been achieved |
| | • Criteria for selecting actors is agreed | • Stakeholders agree the adequacy of actor network | • Users have been able to operate the products and services |
| | • Competences such as skills and techniques to use are selected | • Important service encounter and process are verified | • Functions and performance have been tested and demonstrated |
| | • Purchasing, construction and reuse policy is decided | | • The defect level have been allowed to receivers |
| | | | • The contents of the products and services are well-known in each version |
| **Alpha state** | **④Ready** | **⑤Operational** | **⑥Continuous improvement** |
| **Checklist** | • The products and services manual have been available | • Products and services have been used in the receiver environment | • A system for observing information about customers has been established |
| | • Receivers have accepted the products and services | • Products and services have been used by supposed operator | • A team for continuous improvement have been organized |
| | • Receivers have prepared the use of products and services | • There have been a use case that all functions of products and services have been operated | • Process for continuous improvement has been defined |
| | | • The maintenance level of products and services has been agreed | |

### 3.3. Design process management using PSS design cards

As the third step, this paper proposes a PSS design process management method. The proposed design process management is performed in a tangible (touchable) way using the PSS design cards constructed in the previous steps. As a particularly important point, this management should not be performed by only one person like a project manager of the design team, but instead by all members of the design team. The proposed method manages the design process by handling and operating PSS design cards in an actual design scene.

The PSS design process management is performed following steps using the developed PSS design cards. This management is cyclically performed in the PSS design process.

#### 3.3.1. Analysing the current state of the PSS design process

To analyse the current state of the PSS design process, all cards are arranged in the order of the alpha state, as shown in Figure 5. Then, team members discuss and evaluate whether the checklist described on the card is being fulfilled. Based on this, team members classified these cards into two groups: the checklist being fulfilled or not fulfilled. This grouping enables team members to judge the progress of the design process.
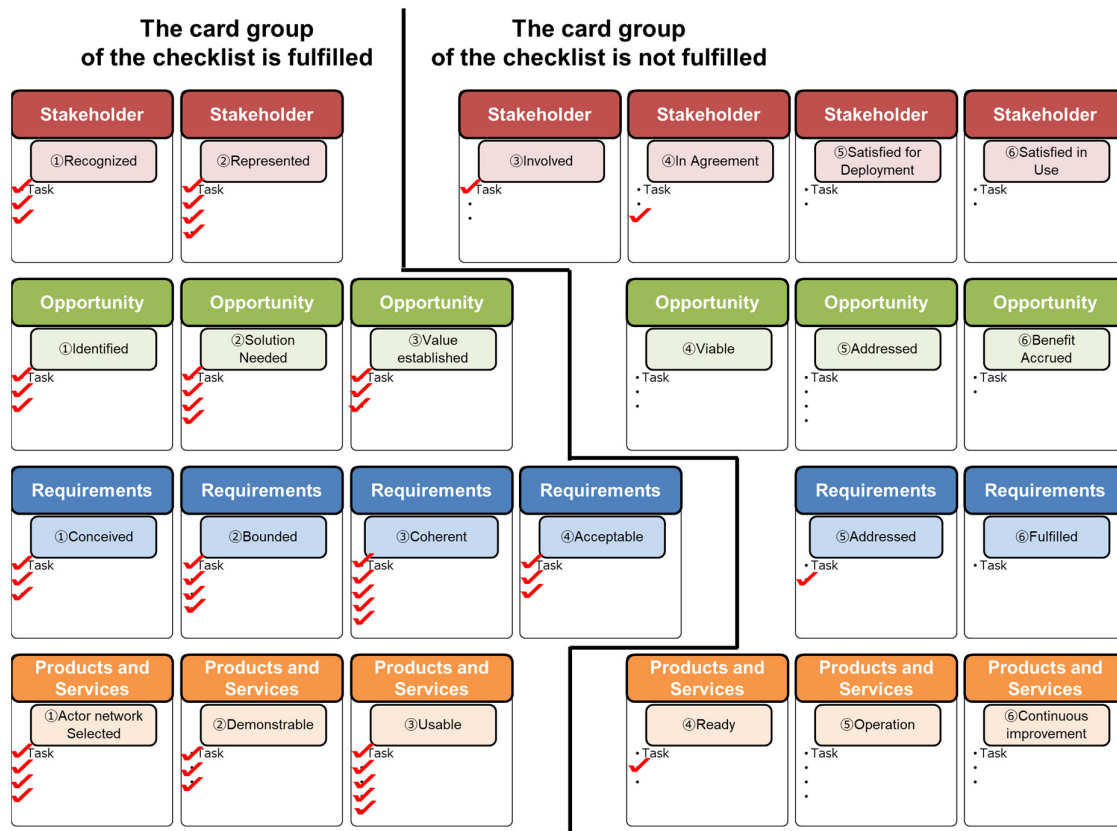


**Figure 5. Analysing the current state of the PSS design process**

#### 3.3.2. Setting the next target in the PSS design process

Team members decide the next target alpha state after understanding the current state of the PSS design process. At that time, team members discuss which alphas to focus on by referring to the analysis results of the current state and determine the next target alpha. Then, team members decide which unfulfilled alpha state the team should reach next.

*3.3.3. Setting up the next team tasks to achieve the next target*

Finally, team members discuss the tasks to achieve after setting the next target. At that time, team members decide which tasks to work on next through task assignment and prioritisation.

## 4. Application

This study applied the proposed method to the developing support service of basic software that is utilized in automobile parts. This service facilitates the interactions between the product development team and the manager, supporting software developers by providing product specification data. This application aims to verify that the proposed method can comprehensively organise tasks for the development support service. Specifically, through an interview with a practitioner of this service, the checklist in the PSS design cards was associated with actual development tasks.

Table 5 shows the case application result of alpha: "stakeholder" and alpha state: "recognized". The authors associated development tasks of the service with the checklist of the card. As a result, corresponding points were interpreted as follows. In the first task, agents on Flow model are associated with customers. In the second task, representatives are associated with promoters who is the windows of the supplier side. In the third task, functions or scopes carried out by each agent are associated with divisions and organizations that are responsible for decision in the supplier.

**Table 5. A part of the case application results (Stakeholder ①Recognized)**

| Alpha | Stakeholder | |
|---|---|---|
| Alpha state | ①Recognized | |
| | PSS design card | Case application |
| Checklist | • Agents on Flow model (potential provider, receiver, relay agent)have been identified | • Customers have been identified |
| | • Representatives of all agent have been agreed | • Promoters who is the windows of the supplier side have been decided |
| | • Functions or scopes carried out by each agent have been defined | • Divisions and organizations that are responsible for business judgment, product line development, and product development in the supplier have been identified |

In the same way as Table 5, the authors associated development tasks of the service with the checklist of the card in all alphas and alpha states. Table 5-8 shows the case application result for one alpha state in each alpha. In this way, it was possible to associate almost all the development tasks of the support service prepared in advance with the tasks in the checklist of the PSS design cards.

**Table 6. A part of the case application results (Opportunity ①Identified)**

| Alpha | Opportunity | |
|---|---|---|
| Alpha state | ①Identified | |
| | PSS design card | Case application |
| Checklist | • Receiver's activities that can be supported by solutions of products and services have been identified | • problems in the software introduction to be solved by support service are identified |
| | • Receivers have grasp the potential value and have wishes to invest | • Manager understands ROI to be realized and product development team understands QCD to be realized, and they wants to invest |
| | • Agents that share the supportable receiver's activities have been identified | • · Problems identified by stakeholders are shared |

**Table 7. A part of the case application results (Requirements ①Conceived)**

| Alpha | Requirements | |
|---|---|---|
| **Alpha state** | ①Conceived | |
| | **PSS design card** | **Case application** |
| **Checklist** | • Requirements and functions of the system have been clear | • Requirements of the support service have been clear |
| | • Users have been identified | • Receivers of the support service have been identified |
| | • First capital investor have been identified | • A manager who will invest have been identified |

**Table 8. A part of the case application results (Products and Services ①Actor network selected)**

| Alpha | Products and Services | |
|---|---|---|
| **Alpha state** | ①Actor network selected | |
| | **PSS design card** | **Case application** |
| **Checklist** | • The products and services architecture that can cope with important risks is adopted | • The products and services architecture that can cope with important risks is adopted |
| | • Criteria for selecting actors is agreed | • Criteria for selecting actors is agreed |
| | • Competences such as skills and techniques to use are selected | •Resources to use are selected |
| | • Purchasing, construction and reuse policy is decided | • Purchasing, construction and reuse policy of resource is decided |

## 5. Discussion

### 5.1. Effectiveness

Tasks to achieve in PSS design are structured and organised in the construction of the PSS design kernel in this study. Therefore, it can be expected that designers understand the tasks in PSS design accurately and manage the design process based on the proposed tools and method. In addition, this tools and method also enable designers to identify the barriers that prevent them from shifting to PSS providers by evaluating the ease of the execution of tasks described on the card in advance. By clarifying the tasks necessary to resolve identified barriers, it is possible to support companies to smoothly shift to PSS providers.

In the economics field, the Pareto principle (also known as the 80/20 rule) (Pareto, 1971) states that, for many cases, roughly 80% of the effects come from 20% of the causes. The experience that "20% of team members work on the team actively, but 80% of team members work on the team inactively" is an example of the Pareto principle. With this statement, this method encourages all members of the design team to participate in PSS design. In addition, there is usually no project manager in agile development. Therefore, in order to design PSS with agile development, all team members are required to grasp the progress. PSS design cards enable to do this by showing it tangibly by card. In this study, PSS design cards were developed with these reasons. Designers can spread the cards to the desk actually, which encourages members to participate in discussions more actively. Namely, PSS design cards make it possible to discuss the current state of the PSS design process and set up the next team tasks more smoothly through work assignments and collaboration among team members. This is also the common foundation necessary for PSS design teams and is considered to be the basis of work. In addition, since proposed method works as a common foundation of PSS design, it can be used with other PSS design methods.

### 5.2. Validity and future works

In the case application, the proposed method associated development tasks in actual support services that have been arranged in advance with the checklist of the PSS design cards. The application results

demonstrated that tasks in the development of support services can be represented by the scheme of the proposed method. However, practitioners who applied the proposed method to the case pointed out that the tasks listed in the PSS design cards might not fully cover the actual development tasks comprehensively. It is considered that the tasks and the priority of alpha states to reach in PSS design vary depending on the type of PSS business model for which designers aim. Moreover, these tasks can differ depending on how each company shifts to PSS providers. Therefore, future works should find ways to customise the tasks based on the type of PSS business model.

## 6. Conclusion

In this paper, the authors proposed a PSS design management method for PSS designers using PSS design cards. In addition, the application of the proposed method shows that the tasks in actual PSS design process were expressed and organised by the proposed tools and method. On the other hand, it was pointed out that the tasks may not be comprehensive and valid. Therefore, future works should determine how to make the tasks comprehensive and valid.

## Acknowledgment

## References

Akasaka, F., Nemoto, Y. and Shimomura, Y. (2013), "Product-Service Systems Design focusing on System Aspect: Total Value Creation for Various Stakeholders", *Proceedings of CIRP, International Product-Service Systems Conference 2013, CIRP, Bochum, Germany*, pp. 371-382. https://doi.org/10.1007/978-3-642-30820-8_32

Baines, T.S., Lightfoot, H.W., Evans, S., Neely, A., Greenough, R. et al. (2007), "State-of-the-art in product-service systems", *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* Vol. 221, No. 10, pp. 1543-1552.

Jacobson, I. (2012), *Essence – Kernel and Language for Software Engineering Method.* [online] SEMAT. Available at: www.semat.org/documents/20181/27952/SEMAT_submission_v11.pdf/8d819915-7dfe-4173-9209-42396c7ece18 (accessed 12.05.2017).

Jacobson, I., Ng, P.W., McMahon, P., Spence, I. and Lidman, S. (2013), *The Essence of Software Engineering: Applying the SEMAT Kernel*, Addison-Wesley, New York, USA.

Kawakita, J. (1967), *Thinking method – for creativity development (in Japanese)*, Chuko shinsho, Tokyo, Japan.

Meier, H., Funke, B. and Boßlau, M., (2011), "Flexible Resource Planning in the Context of Dynamic IPS2 Business Models", *Proceedings of the 3rd CIRP International Product-Service Systems Conference 2011, CIRP, Braunschweig, Germany*, pp. 165-170. https://doi.org/10.1007/978-3-642-19689-8_30

Morelli, N. (2006), "Developing new product service systems (PSS): methodologies and operational tools", *Journal of Cleaner Production*, Vol. 14 No. 17, pp. 1495-1501. https://doi.org/10.1016/j.jclepro.2006.01.023

Neely, A. (2009), "Exploring the financial consequences of the servitization of manufacturing", *Operations Management Research*, Vol. 1 No. 2, pp. 103-118. https://doi.org/10.1007/s12063-009-0015-5

Ostaeyen, J., Neels, B. and Duflou, J. (2011), "Design of a Product-Service Systems Business Model: Strategic Analysis and Option Generation", *Proceedings of the 3rd CIRP International Product-Service Systems Conference 2011, CIRP, Braunschweig, Germany*, pp. 147-152. https://doi.org/10.1007/978-3-642-19689-8_27

Pareto, V. (1971), *Manual of political economy*, A.M. Kelley, New York, USA.

Paulk, M.C. (2001), "Extreme programming from a CMM perspective", *IEEE Software,* Vol. 18, No. 6, pp. 19-26. https://doi.org/10.1109/52.965798

Rese, M., Meier, H., Gesing, J. and Boßrau, M. (2012), "An Ontology of Business Models for Industrial Product-Service Systems", *Proceedings of the 4th CIRP International Product-Service Systems Conference 2012, Tokyo, Japan*, pp. 191-196. https://doi.org/10.1007/978-3-642-32847-3_32

Rosenberg, D. and Stephens, M. (2007), *Use Case Driven Object Modeling with UML Theory and Practice*, Apress. https://doi.org/10.1007/978-1-4302-0369-8

Schwaber, K. (1997), "SCRUM Development Process", In: Sutherland, J., Casanave, C., Miller, J., Patel, P. and Hollowell, G. (Eds.), *Business Object Design and Implementation,* Springer, London, pp. 117-134. https://doi.org/10.1007/978-1-4471-0947-1_11

Wiesner, S., Winkler, M., Eschenbächer, J. and Thoben, K.D. (2013), "Strategies for Extended Product Business Models in Manufacturing Service Ecosystems", *Proceedings of the 5th CIRP International Product-Service Systems, Bochum, Germany,* pp. 239-250. https://doi.org/10.1007/978-3-642-30820-8_21

Shogo Minato, Master course student
Tokyo Metropolitan University, System Design
Asahigaoka 6-6, Hino-shi, 191-0065 Tokyo, Japan
Email: minato-shogo1@ed.tmu.ac.jp