Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 47 (2016) 18 – 23

www.elsevier.com/locate/procedia

Product-Service Systems across Life Cycle

Servitization Methodology in ICT Service System Design

Shigeru Hosono[a]*, Eriko Numata[a], Yoshiki Shimomura[b]

[a]*NEC Corporation, 1753, Shimonumabe, Nakahara-ku, Kawasaki 211-8666, Japan*
[b]*Tokyo Metropolitan University, 6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan*

* Corresponding author. Tel.: +81-44-431-7032. *E-mail address:* s-hosono@bu.jp.nec.com

Abstract

This paper proposes steps of conceptual service design by assimilation of service modeling methods into conventional design practices of ICT systems integrators. The service models are simplified into four forms: persona-scenario, requirement-function-resource tree, Actors' Map, and service blueprint. The input and output data of each model are formalized with design principles, enabling data chain between the models. The relationships between the data of design deliverables, such as plan, functional design, and detailed design documents, are also defined clearly in accordance with their design procedures. As the results, roles for tasks are clarified, and tasks and deliverables are systematized into seven design steps. The steps help to change the mindset of system integrators from product-oriented to service-oriented and assist servitize their business.

Peer-review under responsibility of the scientific committee of the 8th Product-Service Systems across Life Cycle

KeyWords: Servitization, Service Design, Design Process

1. Introduction

ICT service providers have provided systems integration services to their clients. Systems integrators have played a key role in the service delivery; they implement software components, build on-premises ICT system infrastructures with hardware products, and support system troubles in operation. The clients expect the high degree of perfection of the system infrastructure as the results of services. However, their expectation has been shifted to their valuable experiences through ICT service lifecycle as such infrastructures have become easily integrated with software and hardware functions of cloud services: infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and software-as-a-service (SaaS). This change accelerates the trend from 'own' to 'use' of ICT system infrastructure. To keep up with new value proposition of ICT services to their clients, systems integrators will extend their role from implementation and integration to design of early stage of development – conceptual design of ICT service system. They need to redesign of traditional in-house ICT system infrastructures and propose new valuable design of web services on cloud to their clients, assembling arbitrary software/hardware functions of cloud services. This design task requires expertise of specifying the source of values in a service system lifecycle and embedding it into conceptual service system design. To accomplish their new roles, modeling methods for services such as requirement description, quality function deployment and system model become necessities. However, disparities between service models and their conventional deliverables failed their practices of service design.

2. Related Work

2.1. Service Modeling

To design a service from various aspects, a number of modeling methods has been proposed; persona [1][2], customer journey map [3], storyboard [4], laddering [5], touch point, experience map [6], Actors' Map, view model [7], scope model [7] and service blueprint [8]. The modeling methods help to depict expected users, their behavior and service functions to meet their requirements.

Peer-review under responsibility of the scientific committee of the 8th Product-Service Systems across Life Cycle
doi:10.1016/j.procir.2016.03.058

However, each model does not specify its elements and the use of the models have relied on interpretation of system integrators, whose roles are plan, design, integration, test and deployment of ICT service platforms. In addition, these models have not coherency as each model was established individually, and the valid relationships between the models have not been considered thoroughly yet.

2.2. Modeling Task

The data of service models are developed and shared by system integrators and operators through service development and delivery process. However, the design and use scenarios have not been associated with their tasks. These gaps have surfaced as their conventional tasks are rather rigid following the traditional waterfall software development process. Service modeling in the early phase of development goes back and forth to build consensus among system integrators, who are in charge of different phases, i.e. plan, design or implementation. Hence, agile development approach can fit to provide disciplines for such incremental modeling procedures.

SEMAT is an approach in the software engineering to visualize such iterative progress and prompt next actions for practitioners [9][10][11]. SEMAT defines the kernel to make the essence of agile software development concrete and to identify the areas of a software endeavor that a development team must be mindful of and assess for progress and health. The kernel consists of three discrete areas of concerns; ‘customer’, ‘solution’ and ‘endeavor’. The concern about customer includes everything to do with the actual use and exploitation of the software system to be produced. The concern about solution comprises everything related to the specification and development of the software system. The concern about endeavor covers everything related to the development team and the way they do their work. Each area of concern consists of ‘Alphas’ (abstract-level progress health attribute) to understand, monitor, direct and control for progress and health of endeavors. SEMAT Kernel provides basic alphas: ‘opportunity’, ‘stakeholders’, ‘requirements’, ‘software system’, ‘team’, ‘work’, and ‘way of working’. Each alpha stipulates states and a checklist to help professionals understand the progress and health of software development. The checklist can be represented as alpha cards and used in development. By aligning the alpha cards through a service lifecycle, the development team understands the position in the development process. The alpha cards can help developers to understand the current state - where the development team is and where they should do next. Hence, the development can progress in a balanced and cohesive manner.

However, development process in business makes progress by the milestones of document development and review as the status of documents is an observable indicator. Therefore the lack of bridges between state and document management becomes the cause that state and task management in an agile manner has been introduced to service development sites at a snail’s pace.

To increase the adaptability of service modeling to their conventional practices, the next section discusses a simplified service modeling and assimilation approach.

3. Service Modeling in Business

3.1. Requirements for Service Modeling in Practice

The elements of each service model need to be specified, and relationships between data of service models of different perspective need to be specified though data is transformed to another terminology in an adjacent service model.

To provide systematic service models, we propose a set of modeling, which covers 5W1H (when, where, who, what, which, how). ‘Persona-Scenario’ model depicts what the target customer need. ‘RFR Tree’ obtains the needs from Persona-Scenario and depicts which functions and resources will be provided. ‘Actors’ Map’ obtains defined functions and resources from the RFR Tree and depicts who/where are going to provide them in a service system. The last modeling, ‘Service Blueprint’, obtains what, which, who, where of the target and depict when and how to provide them.

3.2. Persona-Scenario

Persona-Scenario depicts a hypothetical users targeting customers segment. The model depicts the users’ profiles such as age, family, lifestyle, motivation and values of their life, and expected behavior in their daily life. The users’ needs are described in the profiles and scenarios.

3.3. Requirement-Function-Resource Tree

RFR (requirement-function-resource) Tree depicts users’ and customers’ wants comprehensively, and clarifies what kinds of functions, channels and resources should be delivered. The users’ wants are obtained from the needs specified in the Persona-Scenario. RFR Tree structures what will be implemented to meet the wants by three steps: service functions (contents), service channels (channels), and service resources (resources). RFR Tree can visualize complete functions and resources for users’ or customers’ wants. RFR Tree visualizes complete translation of users’ and customers’ wants into functions, and of functions into resources (Fig.1).

While deployment users’ and customers’ wants into functions, (1) the wants are transcribed to functions for fulfilling them, and (2) the functions are transcribed to resources for implementing them. This step-by-stem procedure can form a structure of RFR tree including all possibilities of functions and resources for delivery. Then, the design team considers constrains of development period and operational costs, and prioritizes the functions and resources in the tree. After the functions and resources to deliver are identified, the design team considers which actor is going to provide the group of resources. Then, resources are modularized in terms of actors i.e. resource ownership.

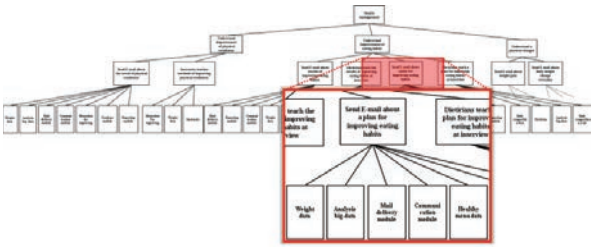

Fig. 1. Example of RFR Tree.

3.4. Actors' Map

Actors' Map depicts the relationships among stakeholders and ICT modules, i.e. software, hardware and networking, for functional delivery of services. The actors can be obtained from the modules in the RFR Tree. An actor can be represented a branch of RFR Tree; the relationships between content, channel and resource. The relationship of functional delivery between actors is represented as an arrow. One or more actors may belong to a single stakeholder. For instance, modularized software functions are represented as an actor, and it is allocated to a delivery division, which operate the module. In the same way, another software and hardware functions are modularized as an actor and it can be allocated to the same division. This procedure completes when comprehensive relationships between stakeholders and function modules are depicted. Then, system integrators can reconsider other possibilities of design; the functions in the actor can be delivered from another actor with views of data dependency or impact upon qualities, such as performance and security, and then, finalize their best structures.

When ICT service providers are to operate and maintain a service system, they need to acquire a number of resources to operate the system. The ICT provider can have partnerships with other stakeholders, which have competency in elements of service systems, to deliver comprehensive functions and resources. For example, a health care service system can be delivered in a collaboration of an ICT service provider and a health care firm (Fig. 2). In this manner, lack of technology or skills in the original actor can be substituted through another actor in modeling procedures, and ideal relationships between actors can be obtained.

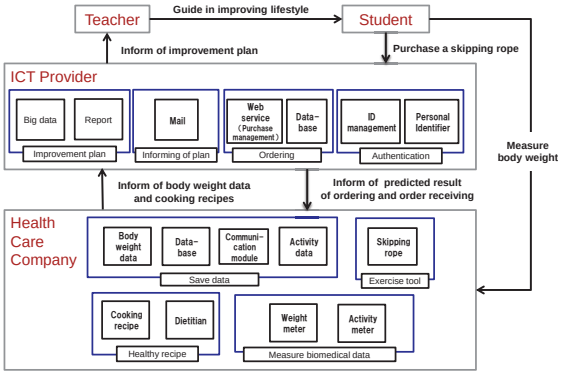

Fig. 2. Example of Actors' Map.

3.5. Service Blueprint

Service Blueprint depicts a service delivery process. It determines collaborations between actors in the delivery phase, visualizing risks for continuous service operation (Fig. 3). When system engineers develop a service blueprint, they start by transcribing actors from an Actors' Map and transcribing contents and channels from the RFR Tree. The actors, contents and channels are aligned with users' scenarios from a persona-scenario model. The top lane shows user's behavior and each lane depicts user's phases of service encounters: (1) access, (2) check-in, (3) diagnosis, (4) service delivery, (5) check-out, (6) follow-up, and flows of functional deliveries and behavior in each phase.

The modelling procedures are as follows:

1. The top lane is tagged for 'User'.
2. Users' actions based on scenarios are depicted in the top lane chronologically, starting from the left.
3. The second to the last lanes are tagged for Actors of front-end to back-end toward the user, and each actor's actions are drawn up chronologically.
4. The relationships of functional delivery between actions in different lanes are depicted as arrows.
5. The relationships of functional delivery between actions in the same lane are depicted as arrows, indicating work flows of the actors.

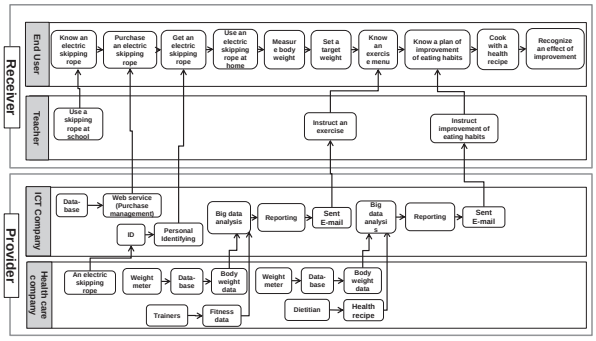

Fig. 3. Example of Service Blueprint.

4. Servitization Methodology

4.1. Design Process

To introduce the service modeling in service development, the flow of design process needs to be shared among practitioners, as they can comprehend the whole design process. Persona-scenario helps design target users' behavior. RFR Tree helps specify requirements and functions. Actors' Map helps to determine entities to deliver the functions. Service Blueprint helps to design a delivery process of functions (Fig. 4).

The comprehensive steps of conceptual design are streamlined as follows:

1. Observe user's As-Is behavior and difficulties.
2. Depict user's To-Be behavior with Experience Map.
3. Ideate solutions to realize the To-Be behavior.
4. Determine a portrait of target user and its requirements with Persona-Scenario.

5. Transcribe the requirements into functions and resources to be delivered with a RFR Tree (‘what’, ‘which’).
6. Determine appropriate entities to provide the functions and functional relationship between them with an Actors’ Map. (‘who’, ‘whom’)
7. Depict a delivery process of functions with a Service Blueprint. (‘when’, ‘how’)
8. Document requirements, functions to deliver, stakeholders to be involved and delivery plans from the data in Persona-Scenario, RFR Tree, Actor Map and Service Blueprint diagrams.

In accordance with their design procedures, relationships between the data of design deliverables, such as plan, functional design, and detailed design documents, are clarified. With this, roles for tasks are clarified, and tasks and deliverables can be systematized into seven design steps.

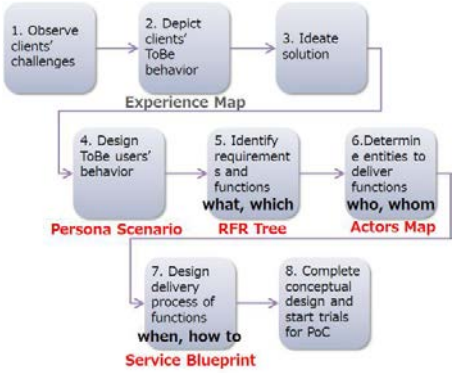

Fig. 4. Design Process.

4.2. Role, Task and Documentation in Design Process

By assimilating the model into conventional design practices of ICT system integrators, the design process is formalized with the following seven steps.

Step1. Hypothesis of customer segment and business volume

The design targets are (a) business segment, (b) list of all requirements, (c) list of service functions, (d) unit price. The deliverable is business proposal (Word). The owner is the planning division. Reviewer is the responsible person for service business.

Step2. Validation of customer segment and business volume

The design targets are (a) list of requirements (updated), (b) list of service functions (updated), (c) unit price (updated). Intermediate deliverables are (a) workflow description sheets (Excel), (2) customer hearing sheets (Excel), (3) validation sheets (Excel). The deliverables are a service business proposal (Word). Owner is the planning division. The reviewers are (a) responsible person for service business, (b) sales person in the service business division

Step3. Hypothesis of service systems’ structure

The design targets are (a) definitions of requirements, functions for the requirements, and resources for the functions, (b) definitions of service system structure.

The intermediate deliverables are (a) workflow description sheets (Excel), (b) customer hearing sheets (Excel), (c) validation sheets (Excel). The deliverables are the service business proposal (Word). The owner is the planning division. Reviewers are (a) responsible person for service business, (b) sales person in the service business division.

Step4. Validation of service systems structure

The design targets are (a) visual of constraints from costs or resources. The intermediate deliverables are (a) **RFR Tree** sheet (customer side) (Excel), (b) **Actors’ Map** sheet (customer side) (Excel), (c) Service ideas (Excel). The deliverables are an executive summary of service system (Power Point). Owner is the Planning division. The reviewers are (a) responsible person for service business, (b) sales person in the service business division

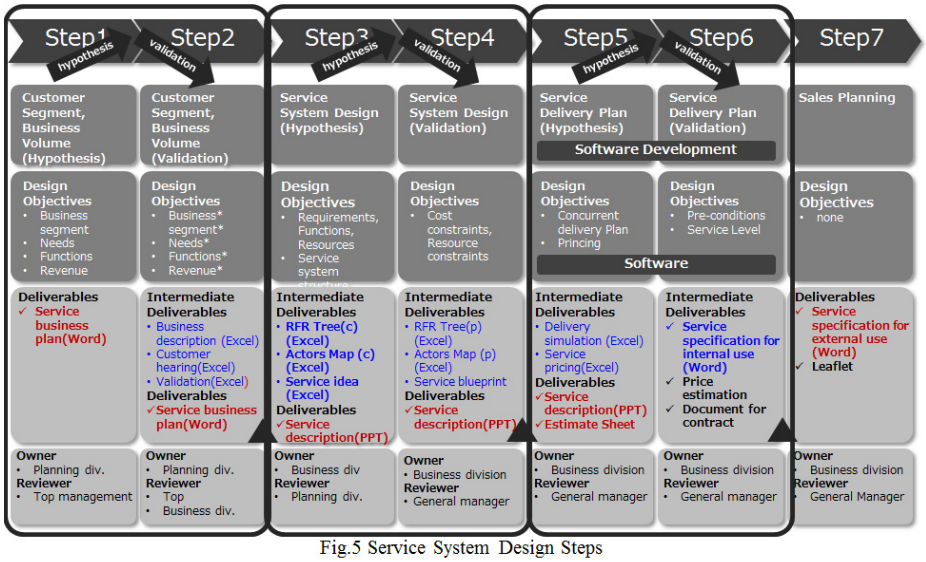

Fig.5 Service System Design Steps

Step5. Hypothesis of service delivery plan

The design targets are (a) Service delivery planning, (b) Service pricing. The intermediate deliverables are (a) Requirement-Function-Resource sheet (customer side) (Excel), (b) Actors' Map (customer side) (Excel), (c) Service blueprint. The deliverables is a service description (power point). The owner is sales person of business division. Reviewer is the general manager of business division.

Step6. Validation of service delivery plan

The design targets are (a) prior conditions, (b) service level definitions for SLA. The deliverables are (a) service specification for internal use (Word), (b) estimated value, (c) service contract. The owner is the sales person of the business division. The reviewer is the general manager of the business division.

Step7. Sales Preparation

The deliverables are (a) service specification for external use (Word), and (b) a leaflet. The owner is the business division (sales, promotion). The reviewer is a general manager.

4.3. Tasks for Documentation

To help such iterative considerations, we extend the kernel alpha. The alphas are stakeholder, requirements, service system platform, and way of working. Each alpha consists of six states with deliverables: models as intermediate documents and final documents for reviews (Appendix Table 1). The states with documents are redefinitions of the seven steps. Monitoring and controlling design process with the alpha allows back-and-forth progress under the rigid seven steps.

5. Case Study

The steps 1-4 were applied to design a new BtoBtoC service. The service is targeted for small and medium business - a store operation, such as a supermarket group, a hotel group, and a hospital group under the same brand. Step 1 and 2 were conducted by a member of a staff division, which has responsibility of whole service business in business divisions. Step 3 and following steps were led by integrators of a business division. Following the steps with modeling, they obtained a price management module, and introduced as a new actor to attain their needs for controlling vacancy in branches against profitability (Fig. 6). The tasks stipulated in each state of Alpha (Table 1) guided the project members which service modelling they should do at the moment and how much they should depict the model precisely for the next state. The design procedures were trial and errors initially, but the seven steps and alphas could indicate their next actions in their gradual design work and fasten to reach the design solution.

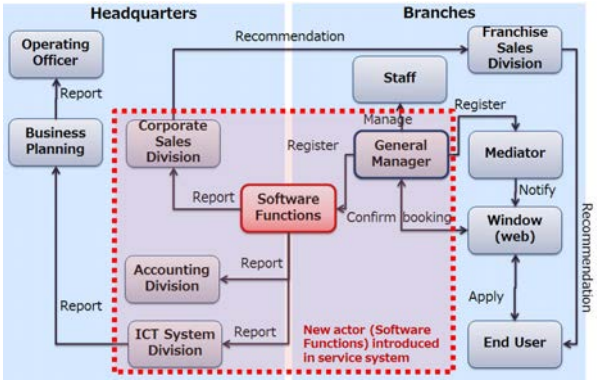

Fig. 6. Actors' Map for Franchise Management.

6. Discussion

The service models are simplified into four forms: Persona-scenario, RFR Tree, Actors' Map, and Service Blueprint. The input and output data of each model are formalized. The models and their relationships are clear and easy to understand for system integrators. They could make progress of service system design without an omission or oversight in the modeling steps. The models are also assimilated to the seven steps, which are the line that extends their conventional development practices. Hence, the system integrators can easily conduct service modeling under the steps: step1 of users' value design to identify needs, step 2 of value design validation to identify wants, step 3 of conceptual system design to determine functions, step 4 of system design validation to determine resources, step 5 of delivery plan design to design delivery process, step 6 of plan validation to fix business plan, and step 7 of preparation of promotion and sales. The kernel alpha of Table 1 worked successfully to control the design procedure, though the system integrators have to refer the table every single step. To make this design methodology more practical, the tasks and stakes of Alpha can be integrated project management tools, which the service integrators have been conventionally used.

7. Conclusion

To help inexperience systems engineers to do advanced design tasks in early development phases, this paper presents steps of conceptual service design based empirical analysis of assimilation of service modeling methods into conventional design practices of systems engineers. The steps are applied to development of a web service on cloud - management service for chain stores. The results showed that the steps help planners, managers and systems engineers to collaborate and design minimum structure of a web service with high user experiences systematically. The tasks and documentation defined in the seven-step gives practical disciplines for service practitioners. Hence, the seven-step design methodologies enhance servitization for ICT service providers shifting product-oriented system integration to experience-oriented service business. Future study will include implementation of the approaches into project management tools to govern design practices in a rather practical manner.

References

[1] Cooper, A., 1999, The Inmates Are Running the Asylum, SAMS.

[2] Pruitt, J., Adlin, T., 2006, The Persona Lifecycle : Keeping People in Mind Throughout Product Design, Morgan Kaufmann.

[3] A Richardson - HBR Blog Network, posted, 2010, Using customer journey maps to improve customer experience, http://www.iimagineservicedesign.com/wp-content/uploads/2015/09/Using-Customer-Journey-Maps-to-Improve-Customer-Experience.pdf

[4] Using visualization techniques for enhancing public participation in planning and design: process, implementation, and evaluation, K Al-Kodmany - Landscape and urban planning, 1999 - Elsevier

[5] Laddering theory, method, analysis, and interpretation, TJ Reynolds, J Gutman - Journal of advertising research, 1988 - universalrg.org

[6] Multilevel service design: from customer value constellation to service experience blueprinting, L Patrício, RP Fisk, L Constantine - Journal of Service Research, 2011 - jsr.sagepub.com

[7] Service Engineering: a novel engineering discipline for producers to increase value combining service and product, T Sakao, Y Shimomura - Journal of Cleaner Production, 2007 - Elsevier

[8] GL Shostack - 1984 - lcis.com.tw - 1984 - lcis.com.twservice blueprint, http://www.lcis.com.tw/paper_store./paper_store/servicesThatDeliver-20141291771178.pdf

[9] Jacobson, I., Ng, P.W., McMahon, P., Spence, I., Lidman, S., The Essence of Software Engineering: Applying the SEMAT Kernel, Addison-Wesley; 2013.

[10] Jacobson, I., Huang, S., Kajko-Mattsson, M., McMahon, P., Seymour, E., SEMAT: Three-Year Vision, Programming and Computer Software 38:1; 2012, p. 1-12.

[11] The Object Management Group (OMG), Documents Associated With Essence - Kernel And Language For Software Engineering Methods 1.0 - Beta 1; 2013, http://www.omg.org/spec/Essence/1.0/Beta1/PDF.

Appendix:

Table 1: Kernel Alphas for Service System Design

Concern	Kernel Alphas	State / Checklist 1	State / Checklist 2	State / Checklist 3	State / Checklist 4	State / Checklist 5	State / Checklist 6
Customer	Stakeholders	Recognized: - Stakeholders have been identified - There is agreement on stakeholder groups to be represented - Responsibilities of stakeholder representative defined	Represented: - Stakeholder representatives appointed - Stakeholder representative agreed to take on responsibilities and authorized - Collaboration approach agreed - Representatives respect team way of working	Involved: - Stakeholder representative carry out responsibilities - Stakeholder representative provide feedback and take part in decisions in timely way	In Agreement: - Stakeholder representatives agree their input is valued and respected by the team - Stakeholder representatives agree with priorities	Satisfied for Deployment: - Stakeholder representatives provide feedback on system from their stakeholder group perspective - Stakeholder representatives confirm system ready for deployment	Satisfied in Use: - System has met or exceeded minimal stakeholder expectations - Stakeholder needs and expectations are being met
	Document	- ***RFR Tree*** *(intermediate)*	- ***Actors' Map*** *(intermediate)*	- ***Actors' Map*** *(intermediate)*	- ***Actors' Map*** *(intermediate)*		- **Service Business Plan**
Solution	Requirements	Conceived: - The need for a new system is clear - Users are identified. - Initial sponsors are identified	Bounded: - Success criteria are clear - Mechanisms for handling requirements are agreed on - Constraints and assumptions are identified	Coherent: - The big picture is clear and shared by all involved - Important usage scenarios are explained - Priorities are clear - Conflicts are addressed	Acceptable: - Requirements described a solution acceptable to the stakeholders - The rate of change to agreed-on requirements is low - Value is clear	Addressed: - Enough requirements are implemented for the system to be acceptable - Stakeholders agree the system is worth making operational	Fulfilled: - The system fully satisfies the requirements and the need - There are no outstanding requirement items preventing completion
	Document	- ***RFR Tree*** *(intermediate)*	- ***Actors' Map*** *(intermediate)*	- ***Actors' Map*** *(intermediate)*	- ***RFR Tree*** *(intermediate)*	- ***Actors' Map*** *(intermediate)*	- **Service Specification**
	Service System Platform	Architecture Selected: - Architecture selected that address key technical risks - Criteria for selecting architecture agreed - Buy, build, and reuse decisions made	Demonstrable: - Key architecture characteristics demonstrated - Relevant stakeholders agree architecture is appropriate - Critical interface and system configurations exercised	Usable: - System is usable and has desired quality characteristics - System can be operated by users - Functionality and performance have been tested and accepted - Defect level acceptable.	Ready: - User documentation available - Stakeholder representatives accept system - Stakeholder representative want to make system operational	Operational: - System in use in operational environment - System available to intended users -- System supported to agreed service levels	Retired: - System no longer supported - Updates to system will no longer be produced - System has been replaced or discontinued
	Document	- ***Actors' Map*** *(intermediate)*	- ***Service Blueprint*** *(intermediate)*	- **Service Specification**	- **Service Specification**	- **Service Operation Manual**	
Endeavour	Way of Working	Principles Established: - Principles and constraints established - Principles and constraints committed to - Practices and tools agreed to - Context learn operates in understood	Foundation Established: - Key practices and tools ready - Gaps that exist between practices and tools analysed and understood - Capability gaps analysed and understood - Selected practices, and tools integrated	In Use: - Use of practices and tools regularly inspected - Practices and tools being adapted and supported by team - Procedures in place to handle feedback	In Place: - All members of the team are using the way of working - All members have access to practices and tools to do their work - Whole team involved in inspection and adaptation of way of working	Working Well - Way of working is working well for team - Team members are making progress as planned - Team naturally applies practices without thinking about them - Tools naturally support way of working	Retired: - Way of working no longer in use by team - Lessons learned are shared for future use
	Document						- **Service Project Review**