Chapter 10
Systematic Development of Product-Service Systems

Margherita Peruzzini and Stefan Wiesner

Abstract Main problems occurring in Product-Service Systems (PSSs), are due to an inadequate requirements analysis and lack of a strong PSS conceptual design. Problems vary from exceeding budgets, to missing functionalities, unsuccessful market launch, or even project abortion. Furthermore, the special characteristics of a PSS have to be considered already at an early stage of the development process. Requirements Engineering (RE) and design methodology as well as supporting Information and Communication Technologies (ICT) need to establish a common perception of the targeted PSS. At the same time, the inner complexity of PSS leaves requirements analysis, design activities and development tasks fragmented among many disciplines and sometimes conflicting, unstable, unknowable or not fully defined. In this context, a concurrent, transdisciplinary and collaborative design of PSS is required to create feasible and successful solutions. The objective of this chapter is to present a structured approach to face the specific challenges of PSS development in detail, to elaborate a general framework that features a systematic approach for PSS development, and to consider the effects of changes in specific product and service design on a systematic PSS development process.

Keywords Product-service systems (PSSs) · Requirements engineering (RE) · Design methods · Servitization

M. Peruzzini (✉)
University of Modena and Reggio Emilia, Modena, Italy
e-mail: margherita.peruzzini@unimore.it

S. Wiesner
BIBA University, Bremen, Germany
e-mail: wie@biba.uni-bremen.de

J. Stjepandić et al. (eds.), *Systems Engineering in Research and Industrial Practice*,
https://doi.org/10.1007/978-3-030-33312-6_10

10.1 Introduction

Design and development of Product-Service Systems (PSS) is a complex process, mainly due to the transdisciplinarity of the activities to be carried out and the cross-disciplinarity of input and output data. As a consequence, a concurrent, transdisciplinary and collaborative approach is required to promote the exchange of engineering knowledge about user and system requirements, design specifications and processing instructions between different stakeholders. In particular, this chapter provides and discusses a general overview of the design and requirements engineering (RE) methodologies, and proposes a high-level integrated model for P-S lifecycle management. Furthermore, a systematic PSS development framework is offered to manufacturing enterprises to determine their own position on the PSS maturity scenario and to plan future developments. It is based on a set of models determining the different aspects of servitization [1]. An Innovation Potential model describes the level of novelty of potential servitization approaches and gives the enterprise and its partners an impression of the general options for product service combinations, the required development work and the expected uncertainty. An Opportunity Potential model identifies the opportunities for innovative combinations of products and services and formulates a challenge to ask for ideas and trigger idea generation. Finally, a Search Areas model helps manufacturing enterprises to look beyond well-known domains that are already served ("looking out-of the box") and do this in an efficient way.

10.2 Design Methods for PSS

Usually manufacturing companies have well-defined and structured product development processes, but they lack a sufficiently definition of the service development processes as found in traditional service companies. Therefore, they are poorly equipped with appropriate approaches, methodologies and tools for supporting the development of PSSs in an efficient way. With manufacturer business models increasingly being extended to include the service phase of the product lifecycle, this poses a significant issue.

In literature several methodologies have been proposed to design a Product-Service System along its entire lifecycle [2]. Some of them are very theoretical and hard to implement in practice, while others are very specific and have a limited applicability range. The most significant existing PSS design methods to be applied to manufacturing contexts usually come from service design and engineering. Some of them are characterized by a transdisciplinary approach. The most significant are listed below:

- UML 2.0 model: this approach allows to concurrently conduct a systematic technical-services design and the corresponding product design process, as proposed by Aurich et al. [3];

- Model-based approach to allow Industrial PSS (namely IPSS) design modelling: it fosters functional behavior and modelling of PSS artefacts (Welp et al. [4]);
- Service Computer-Aided Design (CAD): this is able to support decision-making evaluation through the concepts design, prompting different alternatives scenarios, but it needs a structured Integrating Service CAD Lifecycle simulation (namely ISCL) to also allow a quantitative and probabilistic PSS design as suggested by Komoto and Tomiyama [5]. It is transdisciplinary because it merges technical issues and economical sciences;
- Software tools for designing service activity and products concurrently: these usually have to be adopted in a collaborative way from the early phase of PSS design. Different simulation tools have been recently proposed [6, 7], including service availability prediction [8]. They all consider a variety of aspects with a transdisciplinary perspective;
- Service Engineering based on Structured Analysis and Design Technique (SADT): SADT represents the PSS by its technical specifications by fully describing the object-service system, considering the different combinations of the two main aspects of total core products, from system architecture (i.e., hardware and service support system) to business issues (i.e., markets, risks, partnerships, business chains, agreements, sales and distribution) [8–12]. All mentioned studies report valuable examples of transdisciplinarity, thanks to the combination of technical and business aspects;
- Knowledge-sharing network: the traditional knowledge management systems adopted in the majority of manufacturing industries are product-oriented and can hardly be applied to PSSs, due to the wide set of competences required. In this context, the role of Web 2.0 technologies is a key factor in managing knowledge along the PSS life cycle in a transdisciplinary way [13];
- Lifecycle-oriented PSS approach: it implies a new definition of the product lifecycle phases, from ideation to delivery, use, and disposal in order to be in line with the service proposal. Indeed, the PSS lifecycle approach involves both the product and the service development in order to align designers and providers in delivering of the same solution that is more sustainable along all the lifetime [14–17];
- Layer-based Development Methodology for PSS: it is a new development methodology for Industrial Product-Service Systems (IPS2) that aims to integrate both products and services into a unique solution for creating innovative business models able to generate an added value for the customer. Such a method involves several models to support early PSS development phases, i.e. the PSS planning and requirements engineering [18].

On the basis of industrial case studies, a list of design guidelines have been defined in the current state of the art:

- Requirement Elicitation (RE) is a crucial method to identify the main requirements according to the target market. Indeed, offering PSS instead of a traditional product requires additional competencies to identify the service functionalities to enhance the product, and a better understanding of the customer requirements which must be reached [19]. RE has to be properly addressed for PSS by adopting user-centered approaches;

- Design Structure Matrix (DSM) can be used to define the main PSS functions, combined with Business Use Case (BUC) analysis, which defines the use-case model and a goal-oriented set of interactions between external actors and the involved system [20]. It is a transdisciplinary engineering tool since it merges technical, economical and social aspects;
- Serious Games are useful to investigate the PSS lifecycle and human-system interactions during the design stages [21];
- Quality Functional Deployment (QFD) technique allows mapping the customer needs with the PSS functions in order to elicit the final PSS requirements for the solution to be developed [22];
- UML 2.0 can be used to easily carry out PSS process modelling to define the main activities to achieve the process tasks, identify the enterprise's ability in capturing, sharing and transferring the involved knowledge. The main common techniques for process modelling come from static models, focusing on the information flow (i.e., UML, Petri-Nets, flowcharting, IDEF0, etc.) and dynamic models for process evaluation (i.e., Event-Process Chain), to provide a high-level technical view.

The combination of RE, DSM, QFD and Serious Games can assure a robust user-centred design process and the definition of reliable design specifications. Furthermore, the combination of process modelling techniques assures a deep process analysis to achieve a comprehensive mapping of PSS tangible and intangible assets.

However, existing research focuses on the description of PSS technical solutions, embedded technologies to enable product-related services, methods of data acquisition and elaboration, and software interfaces. However, the focus on technology often neglects the final customer needs. Applying a rigorous User-Centered Design (UCD) approach to PSS has the potential to create customer-oriented and adaptable services and, finally, more effective and efficient models to identify usability problems at the different stages of PSS design. A first example of an integrated transdisciplinary method to support technical and business issues in PSS design process is presented by Peruzzini et al. [23], which tries to provide a QFD approach that drives a designer along both axes of evaluation. Such a method is based on collaboration and knowledge exchange between practice and science, in particular between technical and social sciences (attention to social sustainability, human wellbeing, ergonomics, work organization, etc.).

In order to achieve a transdisciplinary view, UCD techniques (such as interview, questionnaires and role-playing) allow for directly involving users into the design process. In particular, role-playing highlights the PSS users' needs and tasks to be created to satisfy them by directly considering a set of "personas" representing sample users [24]. Role-playing is performed by experts in the specific PSS domain, who play as characters into the real context of use simulating the actions and moods of the consumers. Personas are widely used in the investigation of user experience as fictional characters representing different user types and experiences. Indeed, the combination of user-centered issues has been proven to benefit the final PSS by including the user needs and demands from the early design stages. It is an important issue to develop successful PSS, but it is still not deepened in existing methodologies. In particular, the analysis of human-system interaction is fundamental to predict the relationship that will be created between the user and the PSS and to optimize both product and service functions to have a higher business impact. In order to do that, functional prototypes enhanced with interaction features should be created. In this direction, Mengoni and Peruzzini provide a methodological design framework to support the design of PSS by adopting a UCD approach to involve end-users during the different stages of PSS development, using interactive virtual prototypes [25].

By adopting these methods to industrial cases, some problems still occur. They are mainly related to the definition and evaluation of the user experience generated by the PSS, the complexity to predict the PSS behaviour during the design stage, and finally the assessment of the human-system interaction. When interactive prototypes are used, by exploiting different software tools (i.e., CAD modelling, virtual prototyping developing platform, system simulation) to involve sample end-users to test PSS usability and performance, time and cost for PSS prototyping and optimization can be reduced. However, the creation of proper PSS simulation is still a hard task mainly due to system integration, proper interfacing with service provider platform(s), and reliable data analysis and service behaviour prediction on small test data samples.

10.3 Requirements Engineering for PSS

Understanding the customer and other affected stakeholders expectations, i.e. their underlying needs, and linking information from all phases of the product-service lifecycle to the development process is a prerequisite for successful solution engineering [26–28]. Inadequate RE is a main source for failure of development projects and leads to exceeding budgets, missing functionalities or even the abortion of the project [1]. Often the relevance of appropriate requirements is underestimated, which in turn leads to errors in the requirements specification, not to mention foregoing completeness, consistency, verifiability etc. of requirements. Such errors are mostly discovered late in the development process, thus substantially contributing to higher costs in order to compensate for and correct the errors [29].

Requirements are used to define the needs of stakeholders, such as organizations or individuals along with their environment and specify what a solution must provide to satisfy those needs. Their record, documentation and management are the main objectives of RE. It is “a process, in which the needs of one or many stakeholders and their environment are determined to find the solution for a specific problem” [30].

In traditional development approaches, mainly from the manufacturing domain, RE is seen as a discrete development phase. This has substantial disadvantages when dealing with the increasing complexity, dynamics and time constraints of PSS. Change requests in later phases will not be included in the requirements specification, so it is often insufficiently documented which parts of the original specification are really implemented at the end and which not. Thus, it is difficult to use the requirements specification for change management and testing. Furthermore, each development project will have its own RE phase, with no focus on requirements re-usability [31], increasing the overall development time. An example illustrating this challenge is that in traditional RE scenarios for simple products, the stakeholders are generally aware of their needs. Often, a specific functionality is requested and the product development is based on formalized requirements through a single enterprise. In contrary, a Product-Service System is expected to solve a particular customer problem without prescribing a specific functionality, allowing alternative usage. In addition, cross-linking with other systems and integration into the system environment increases the complexity of the system development even more. System integration leads to a fuzzy problem description which again influences the RE process [32].

PSS require temporary collaboration of different stakeholders in Systems Engineering, which increases the complexity of the RE process. Besides the customer and user of the system, actors like the project manager, product designers, software developers, service engineers, marketing experts, suppliers, quality assurance and many more have to be involved, often being spatially distributed. This induces a change in RE from a quasi-stable and simple environment to a more complex and dynamic variation, making the RE process more challenging, due to both different cultural issues, but also organizational issues like organization of meetings [33] and conflicts as well as interdependencies have to be assessed for a larger number of requirements.

10.3.1 RE Process for Products, Services and IT

In order to select or develop a suitable RE approach for PSS, it is necessary to identify the specific characteristics of such systems. The analysis of some widely used definitions of PSS found in the literature [34–36] reveals some characteristics that seem to be specific for PSS in general. These characteristics are listed below:

- Integration of product and service shares, including software components
- Mutual planning, development, provision and use of product and service shares
- Fulfilling an end user need by delivering value in use
- Provided by either a single company or by an alliance of companies
- Dynamic adoption of changing customer demands and provider abilities
- Enabling innovative function-, availability- or result-oriented business models.

As far as product development, RE approaches have already been implemented with a high degree of formalization. Structured fundamental models exist that provide a general development procedure including RE. However, they focus almost exclusively on requirements development as the main process, which is only conducted at the beginning of the development approach, e.g. by specifying the product requirements document [37]. Sometimes, also aspects of requirements management are adopted, but without explicit instructions for implementation [38].

The following fundamental contributions regarding product engineering approaches have been taken into account:

- RE for PSS—A State of the Art Analysis [39];
- Vorgehenszyklus für die Lösungssuche (i.e., Procedure Cycle for the Solution Search) [40];
- Engineering Design [37];
- Product Design and Development [41];
- Erfassen und Handhaben von Produktanforderungen (i.e., Capture and Manage Product Requirements) [42];
- Entwicklung technischer Produkte (i.e., Development of Technical Products) [43];
- Collaborative Product Design [44];
- New Product Development [45].

The analysis of the future development environment is commonly discussed among the analysed approaches. Possible influences on the product development are identified and the "overall objective of the development" is established. The stakeholders are identified in order to elicit the requirements. The elicitation of requirements is addressed in product engineering approaches; however, procedures for the elicitation of requirements for product-related services are not described. Moreover, the authors state that there are weaknesses in the derivation of requirements from the customer's value chain processes, and cross-domain knowledge is not considered [46]. The necessity of requirements translation of initial stakeholder's requirements to design requirements—concretized and in the language of the developers—is mentioned in the analysed approaches. However, procedures for the concretization of requirements are not mentioned explicitly. Quality Function Deployment is applied in product engineering, but the authors state that "it cannot be employed for new development or for the derivation of design requirements from customer requirements" [39]. Furthermore, the procedures described are not applicable to services due to focus on quantification of requirements. In source [42], the author analysed the approach of [37] and comes to the conclusion that the translation of initial requirements to design requirements is lacking the provision of concrete methods and procedures. However, the concretization is supported by a guideline of characteristic product features helping the developer to elicit the requirements systematically. The guideline is adaptable to varying problems. To derive the priority, the requirements are differentiated between wish and demand [42].

The product engineering approaches provide procedures for identification and resolution of conflicts (e.g. influence matrices). However, the proposed procedures are domain-specific and do not discover conflicts between requirements of different domains. Negotiation with stakeholders and developers is commonly mentioned to resolve conflicts. In source [43], the author proposes two methods to solve a conflict—either to find a compromise between the conflicting goals or to avoid the conflict by changing the concept. Procedures for the documentation of requirements are provided by product engineering approaches. However, cross-domain documentation is not considered. Influence or link matrices are used to trace the requirements to its origin. Interdependencies between requirements of different domains are not captured. Change management is not described in detail; only the necessity of it is observed [39]. Ahrens [42] argues that procedures are provided to structure requirements lists either by thematic affliction or alternatively by the product structure in the methodology of Pahl und Beitz [37]. Requirements validation is mentioned in the analysed approaches but not discussed in detail. Generally, validation is done through evaluation of design drawings by the customer. The authors stated that "the customer plays a central role during the entire development process" [46]. However, the integration of the customer and stakeholders is restricted to the early stages of the development—the requirements elicitation and agreement. The dimension of collaboration between domains is not covered by the state-of-the-art research. Modularization, specification of interfaces and the re-use of modules for different products are mentioned in the product engineering approaches [39]. The authors argued that manufacturers are attempting to reduce costs by "increasing the use of the same parts, or modules, across different products" [45]. Procedures are not explicitly mentioned. Liu et al. described the state-of-the-art on collaborative engineering design systems utilized in the collaborative design process—web-based CAD systems, and propose a new system including tools to resolve conflicts [44]. The web-based collaborative engineering design systems enable the developers from different locations and businesses to share and integrate design models, e.g. via collaborative modelling or multimedia tools such as online chat and meetings. Collaboration during the first phases of the product development is not mentioned explicitly. The utilization of collaborative networks is not mentioned in the analysed methodologies.

In the service area, numerous models for the systematic development of services have been proposed [47, 48]. However, no systematic procedures for the effective implementation of RE in industry have been established yet, because the characteristics of a service (e.g. its complexity, pose greater challenges). Thus, Service Engineering procedures do not integrate a holistic RE until now, but focus more on methods like "trial and error" [49].

The following literature discussing the service engineering state-of-the-art and approaches has been analysed:

- RE for PSS—A State of the Art Analysis [39];
- Design and Management Service Processes [50];
- Key Concepts for New Service Development [51];
- Ein Rahmenkonzept für die Entwicklung von Dienstleistungen (i.e., A Framework for the Development of Services) [47];
- Dienstleistungsproduktion (i.e., Service Production) [52];
- Anforderungsanalyse für produktbegleitende Dienstleistungen (i.e., Requirements Analysis for Product-related Services) [38];
- Service Engineering [47];
- Collaborative Service Engineering [53].

The elicitation process in service engineering comprises the tasks of identifying essential information—e.g. service ideas, possible customers and their expectations, and the sources of the requirements—and determining the goals, chances and risks.

The procedures are service-domain specific; cross-domain knowledge is not considered. Furthermore, no precise methods for the elicitation are provided. To this end, van Husen analysed the elicitation process of conventional service engineering methodologies in detail and comes to the conclusion that procedures for the requirements elicitation are described on a relatively general level [38]. The initial requirements are concretized "by assigning them to quantifiable attributes related to the implementation" [39] and classified into three dimensions—potential, process and result. Consequently, the activities and resources needed for the development as well as the result of the service provision can be derived. According to van Husen [38], only Ramasway provided a detailed design process procedure including the activities of prioritization of requirements according to their importance, specification of attributes which are required to fulfil the needs, and creating a link between the attributes and requirements [50]. The identification and resolution of conflicts is not described explicitly in the service engineering approaches. In the analysed approaches it is suggested to use the procedures known from software and product engineering. The approaches analysed by Berkovich et al. [39] provided a set of procedures to document requirements in natural language without giving detailed information about creating a requirements specification. Traceability of requirements and change management are only mentioned briefly according to the authors. It is argued that the validation of the requirements is described as a comparison of the service concept with the initial stakeholder requirements. The validation is not discussed in detail.

Furthermore, the customer requirements are captured by the elicitation procedures. However, the procedures are only vaguely mentioned. From the analysis it can be derived that the customer and stakeholders are actively involved in the RE processes. Explicit procedures for the collaboration are not described. The modularization of services and re-use is recognized by the approaches analysed by Berkovich et al. [39].

A third aspect to consider in PSS is the IT system. For the development of software systems, standard procedures have been increasingly established. Besides generic process models, specific methods for RE exist. According to the scope and risk of the project, a suitable development model can be selected. In direct comparison with product and service development, RE is integrated deeper and more comprehensive into software development [38]. The following literature discussing the software engineering state-of-the-art and approaches has been investigated:

- RE for PSS—A State of the Art Analysis [39];
- Requirements Engineering Framework [54];
- Requirements Engineering Process [55];
- Requirements Engineering Process [56];
- A Generic Process for Requirements Engineering [57];
- Engineering and Managing Software Requirements [58];
- Requirements Engineering [57];
- Collaboration in Distributed Software Development [59];
- Collaboration in Software Engineering [60].

Requirements and their sources are identified in the elicitation phase and customer-integration is emphasized in the software engineering approaches. Consequently, the focus is laid upon the software domain—interdisciplinary requirements are not considered. Similar to the considered product engineering approaches, the necessity of requirements concretization is recognized in the software engineering approaches. The procedures are not described explicitly and are not suitable for the development of new products or services. The procedures provided for the identification of conflicts focuses solely on the software domain; interdisciplinary conflicts are not discovered. Negotiation with stakeholders is suggested to resolve conflicts and find a compromise [39]. The description of requirements, changes and responsibilities are specified and documented. Model-based requirements documentation is commonly used in software engineering; however, Berkovich et al. [39] state that "there are no procedures and models for the representation of requirements on services, nor for the relationship between the requirements of different domains". Traceability procedures are provided, specifying the affiliation of the requirement towards the different layers of concretization (e.g. a requirement assigned to a component), and linking the design requirements to the initial requirements. The authors describe the utilization of traceability in detail [57]. Interdisciplinary traceability is not considered. It is recognized that requirements can change during any lifecycle phase and changes have to be captured and analysed to "check them for their feasibility by determining their costs and impacts on other requirements, to prepare them for further stages of development, as well as to ensure appropriate documentation" [39]. Change management is described as important during the whole lifecycle including the use phase of the product [57].

Finally, requirements validation is an important part of the RE process to check the requirements for ambiguity and falsity. The design requirements are validated against the initial (stakeholder) requirements to determine the fulfilment of the stakeholder needs. Validation procedures are discussed in detail in the software engineering approaches. The validation focuses solely on software engineering [39]. Customer integration is restricted to the requirements definition stage; integration in other phases such as the utilization of the software is not explicitly mentioned. Modularization is recognized in software engineering approaches. Li et al. state that "requirements encapsulation means organizing requirements into a set of clusters along with external interfaces such that each cluster can be ultimately implemented by a functional module" [61]. Moreover, Lanubile described the state-of-the-art of collaboration in the software domain, focusing on the collaboration between software engineers [59]. Collaboration in software engineering is taking place in various ways during the whole lifecycle of the development, e.g. collaboration with stakeholders to elicit requirements, identification of errors and collaborative working on the software design. The author mentioned knowledge centres as web-enabled tools to share knowledge and argued that "the quality of programmers is the most important factor in software work" and thus, developers are hired regardless of their location. Competencies from non-software domains are not explicitly mentioned. Furthermore, software companies outsource development work to programmers in low-cost countries to reduce development costs. The collaborative environment presented by Lanubile and Whitehead referred solely to the software domain [59, 60]. Nevertheless, collaborative networks are created.

10.3.2 RE Approaches for PSS

RE for PSS has to be conducted for a growing number of tangible and intangible components from a variety of distributed, multi-disciplinary stakeholders. Consequently, only robust and transdisciplinary RE approaches that can deal with the complexity of PSS, its openness and dynamics are suitable. Due the inherent complexity, the direct involvement of the end user and information exchange between the different stakeholders has to be enabled during RE. Thus, the domain specific formalisms and tools have to be made interoperable or substitutable. In this context, the following literature discussing the state-of-the-art and approaches of integrated product and service development has been analysed:

- RE for PSS—A State of the Art Analysis [39];
- Integrated Product and Service Engineering versus Design for Environment [62];
- Solution Approach in the Hybrid Product Development [49];
- Life Cycle Management of Product-Service Systems [63];
- Framework for Development of Product-Service Systems [64];
- Systematic Translation of Customer-specific IT Solutions in Integrated Product-Service Building Blocks with the SCORE Method [65];
- Review of PSS Design Methodologies [66];
- State-of-the-art in Product-Service Systems [35];
- PSS Design [67];
- Developing new product service systems [68].

According to Berkovich et al. [39], the literature about PSS development and design discusses the process of development only abstractly without going into detail. Firstly, the "organizational conditions are created in order to enable an integrated development of services and hardware/software". The stakeholder needs—in regard to products and services—are identified. Concrete techniques or methods are not mentioned. During the concretization the initial requirements are analysed and assigned to the respective domains; a requirements model representing the product structure is created and updated during the entire development. The development process focuses on the single components of the PSS. However, concrete procedures for the translation of initial requirements to domain-specific requirements are not provided. They argue that identification and resolution of conflicts is only mentioned briefly in the selected approaches.

Procedures for the documentation of requirements are provided; model based requirements documentation is not applicable to PSS due to missing procedures and models for the representation of service requirements. Furthermore, there are no procedures to capture the interdisciplinary relationship between requirements. Change management is not deepened in detail in any of the approaches [39]. The validation of requirements "is not discussed in detail" in the analysed approaches. The importance of customer integration in all lifecycle phases is recognized. However, specific methods or procedures are not specified. In addition, modularization is widely mentioned in literature to create standardized solutions [65]. Integration of the involved domains is neither supported in the first phases of the development (e.g. elicitation) nor in later phases such as requirements validation. The necessity of integration is recognized; concrete procedures are not examined [66]. Due to the nature of integrated products and services the necessity of additional competencies is recognized. Aurich et al. noted the "importance of extended value creation networks" [63]. Additionally, the authors argue that the importance of collaboration is only mentioned and "not detailed enough to understand the uniqueness of this process [the collaboration between stakeholders] and how to implement it in real-time" [66].

Engineering of PSS, in contrast to a centralized development process for simple products, requires the orchestration of distributed products, services and business processes for a common purpose. Therefore, organizational, technical and managerial interoperability is a prerequisite for the realization of the system. The RE methodologies of the product, service and software disciplines focus on the respective domain, neither consider the methodologies interdisciplinary requirements nor are interfaces for the handling of interdisciplinary requirements specified. In addition, procedures and methods are solely applicable to the respective domains, making it impossible to apply them to other domains, let alone PSS as a whole. The elicitation procedures in the product domain focus on technical requirements. The methods used to elicit requirements such as checklist are not suited for the elicitation of service requirements. As the concretization of requirements is mainly done by assigning quantifiable attributes, this is not applicable to the intangible part of the EP. Collaboration and integration of development processes with other business partners are not explicitly mentioned. In general, the lack of an interdisciplinary view and thus missing interfaces towards other domains, as well as the insufficient requirements documentation complicate the adoption of product engineering methodologies.

The service engineering methodologies display weaknesses—the procedures provided focus solely on the service domain; interdisciplinary requirements are not considered. The service engineering methodologies are not detailed enough to be used as a basis for PSS development. For example, for the identification of conflicts only a reference to the already existing methods and procedures of the software and product engineering is made. Indeed, the software engineering methodologies do not consider other domains and interdisciplinary collaboration. The procedures described for the prioritization of requirements are not suitable for the development of new products or services. Furthermore, the representation of service requirements is not possible with the provided procedures and modelling techniques. Collaboration is strictly within the software domain, e.g. through networks of companies spread worldwide.

The integrated approaches state the necessity of cross-domain knowledge, interfaces and interdisciplinary requirements. However, the RE methodologies of the integrated products and services are too vague and do not provide the procedures necessary in order to realize a PSS. The procedures are not explained in detail or similarly to service engineering, procedures of other domains are referenced.

To summarize, the adoption of existing requirements engineering methodologies of the product, software and service domain to the development of PSS seems to not be possible as they do not fulfil the requirements for a successful realization of such a complex solution. Especially the lack of a holistic, interdisciplinary view and the corresponding interfaces must be highlighted. A holistic view of the development process of the PSS is necessary. Thus, integration of the development processes of the individual components is mandatory. Missing interfaces to other domains make it difficult to apply domain-specific requirements engineering methodologies to the respective component of the PSS. Moreover, the methodologies of the product domain do not cover all the lifecycle phases required to realize a PSS. For example, change management is not intended after the product has been realized.

The requirements engineering methodologies of the individual domains do not cover the collaboration across domains and the integration of development processes. The requirements engineering methodologies of integrated products and services cover all phases of the PSS, however they do not provide the required integration interfaces. The selection of collaborative business partners depending on the configuration of the PSS and formation of business networks is not described in any of the methodologies.

Thus, two main aspects will have to be supported by a suitable transdisciplinary RE framework:

1. Collaboration and interoperability between stakeholders and PSS components from different domains, especially products, services and software;
2. Management of unstable and unknowable requirements, taking into account information from all PSS life cycle phases.

Integrating PSS components from different domains, like manufacturing and service, requires collaboration between previously separated stakeholders. These stakeholders needed for the realization of PSS typically have their own specific development methodology, standards and even “language”. Thus, the “translation” of requirements between domains needs to be supported by the RE framework to enable a common understanding of the PSS. In order to be adaptable to changing requirements throughout the lifecycle, the value chains have to be flexible even in the PSS usage phase. To support interoperability between PSS components from different domains, new methodologies are needed, also to describe emergent system behaviour. Conflicting, unstable and unknowable requirements have to be identified across the different PSS domains. Methods and tools need to anticipate dynamic changes over the PSS lifecycle and environment. The interdependencies between the tangible product and intangible service components as well as between the stakeholders have to be described. In such a way, the PSS specification and stakeholder information needs could be comprehensively identified.

10.4 Roles in PSS Engineering

Pahl and Beitz defined a number of roles along the product lifecycle, from product origination to disposal or recycling [37]. The following roles are relevant for the product engineering process:

- The *Market/Customer* delivers information about the requirements and constraints in order to generate and select product ideas and create a requirements specification. Furthermore, he/she is the user of the product and gives feedback about product quality.
- The *Product Planner* defines the product portfolio of the manufacturer according to the information from the market (market pull) and the available technology (technology push). The aim of the strategic product planning is the development contract, specified by requirements and justified by a promising business plan.
- The *Product Designer* is responsible to specify the to-be product according to the customer requirements within the necessary documents for prototyping and production. He/she may also be responsible to create and review prototypes.
- The *Production Planner* allocates the necessary employees, materials and production capacity in order to realize the product portfolio created by the Product Planner and Designer. Thus, he/she plans the production and manufacturing processes for the OEM.
- The *Suppliers* deliver the necessary materials, components and missing competencies to realize the product portfolio together with the OEM.
- The *Product Development Team* is comprised of representatives from several of the roles defined above and deals with the coordination of the product development process. Therefore it is responsible for the project management for specific product lines and information exchange between the actors.

Moving to Service Engineering, Freitag et al. described a schematic role model for Service Engineering, with the different role owners in seven phases, from the idea contributor in the ideation phase to the service facilitator during market launch [69]. The role model aligns the specific, function-oriented roles from Scheithauer et al. to the service lifecycle [70]. As an example the owner of the role “Service Manager” is responsible for a set of tasks in the role model. He/she has to decide quickly on proposed service ideas on a strategic level, then allocate resources for the service development project and control the execution of the decision. This role can be fulfilled by an individual person as well as a team or department. Furthermore, the PSS Engineering process is characterized by the inclusion of competences in the form or various actors during the development phases [71]. During a PSS project, the involved actors are determined, and development teams are established and assigned to several PSS specific roles that can be found in literature [3, 71–74].

The application of agile development methods like scrum leads to the definition of additional roles for the model:

- *SE Project Manager*: Comprehensive and frequent communication with customers about Service Engineering results; monitoring of the project's economy regarding development efforts and added value for customer/revenues;
- *SE Project Team*: Shared aim of highly flexible reaction to short-term changes of customer demands, even in late development phases;
- *SE Project Moderator*: Control of the group members meeting working standards; taking care of personal relations in an interdisciplinary team;
- The *PSS Provider* is the focal point of all involved stakeholders and is responsible for the whole PSS lifecycle. The tasks of the PSS provider include the coordination and execution of design, development and production of the product, as well as planning and development of complementary services [75];
- The *Production Network* comprises various PSS suppliers who are responsible for provision of materials, parts and components or system modules for the PSS Provider;
- The *Service Network* contains distributors, subsidiaries and service partners, which are mainly material and service specialists. The main task of the Service Network represents is the PSS distribution, which includes the market-specific adaptation of the integrated service shares and the handling of client orders including the individual PSS configuration;
- The *Customer* plays another key role next to the PSS Provider. Especially in the early stages of development, he/she is considered as the initiating part, because demands towards the PSS will be drawn up and implemented based on the determined customer needs;
- The *PSS Project Manager* acts in various phases of the PSS development process and performs management activities. The main tasks of the PSS Project Manager include the establishment of the connection between the PSS project management and the PSS development process. In addition, it is a task to coordinate the PSS actors and their communication and networking over the phases along the development process;
- The *PSS Architect* can be defined as another PSS specific role. The role is characterized by its PSS specific knowledge and the overarching effectiveness in the PSS development process. The duties of a PSS Architect include, among others, the PSS idea generation, documentation and management of PSS concepts and making the link to the PSS project management. Thus, the activities of the PSS Architect also span over several phases of the development process.

All actors involved in the PSS development process need to communicate with each other in different phases for different reasons. According to the respective phases, thus there is a different distribution of tasks, competencies and responsibilities as well as changing communication needs [75].

10.5 Knowledge Management in PSS

The design phase in the lifecycle of products and services is characterized by an intense exchange of engineering knowledge [76]. This even increases if an integrated PSS shall be designed in a collaborative way, where the tangible and intangible components are entangled and dependent on each other [77–79]. Explicit and tacit knowledge for the engineering process, like user and system requirements, sentiments, competences, design specifications or processing instructions has to be exchanged between the involved stakeholders from different domains [80]. To this end, both knowledge from the product side as well as the service side must be shared in an appropriate way, combined and utilized, in order to create an attractive product-service bundle for the customer. On the one hand, it has to be elaborated which process steps are typically conducted in PSS design [79, 81, 82]. On the other hand, the involved stakeholders have to be identified and described as the relevant knowledge sources and targets [79, 80, 83]. Based on the results, the relevant types of knowledge and appropriate exchange mechanisms and standards have to be defined [79, 84, 85].

In the scientific discipline of Knowledge Management (KM), several approaches to capture, develop and apply knowledge effectively during product design have been developed. Knowledge-Based Engineering (KBE) for example is aiming at establishing engineering knowledge models, for application in product design and along the whole product life cycle [86]. First attempts have also been made to include service knowledge into a KM framework for PSS as well. These attempts are however focusing on using service knowledge for product design and service operations only. Furthermore, most approaches have been focusing on explicit formalized knowledge inside an individual organization [87].

In order to identify the knowledge exchanged in PSS design and develop an appropriate Knowledge Management approach, it is necessary to identify the stakeholders involved in the underlying processes, as they are the relevant sources and targets of knowledge. As the kind of stakeholders depends strongly on the products, services or industrial sector, a more generic classification is necessary. Thus role models are applied, which have shown to be useful for orchestrating the contributions of various stakeholders in innovation processes [88].

Aligned with the process models, role models describe a set of roles, which can be assigned to role owners, which can be internal or external stakeholders. Assignment of roles and tasks should thus be related to the underlying processes, organizational structure, and competences of stakeholders, respectively. The roles define the division of work between the stakeholders. They contain one or more tasks and can be assigned to one or more individuals or organizational units. The following sections will describe existing role models for product, service and PSS engineering.

In order to raise awareness, where relevant design knowledge can be found ("knowing who knows") and who might be interested in a specific knowledge asset ("knowing who should know"), it is required to define distinctive roles for a PSS design project. Such roles can be derived from the PSS lifecycle. Possible roles are

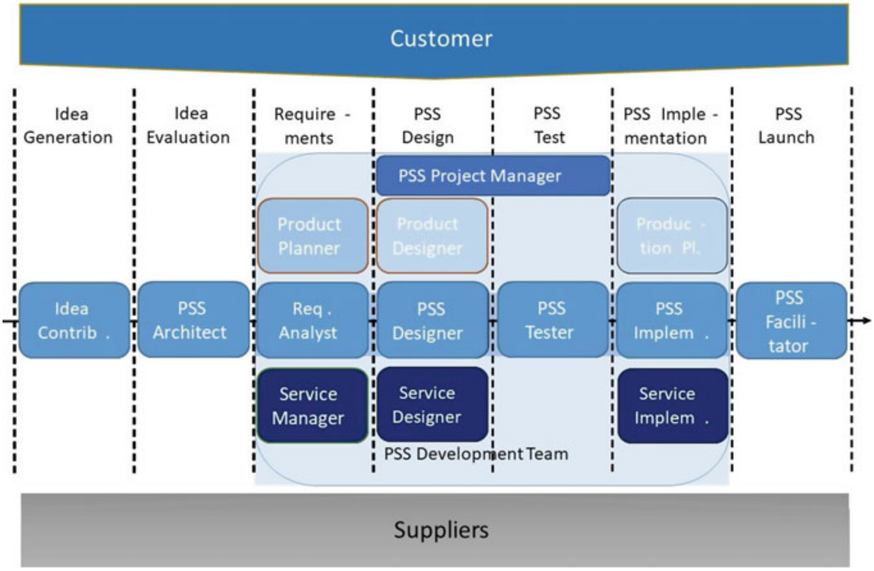

Fig. 10.1 Outline of roles in PSS lifecycle

illustrated in Fig. 10.1. The roles of the PSS Provider (in blue) act as coordinators and are responsible for the execution of design, development and realization of the PSS. The PSS Architect generates documents and manages PSS concepts. The PSS Project Manager coordinates the development team and their communication over the phases along the development process. The PSS Development Team is comprised of representatives from the different domains and deals with the coordination of the product and service development process.

The roles of the Production Network (in orange) are responsible for provision of materials, parts and components or system modules to the PSS Provider. The Product Planner defines the tangible portfolio for the PSS according to the information from PSS Architect. The Product Designer is responsible to specify the product components according to the PSS requirements. The Production Planner plans the production and manufacturing processes for the products. The roles of the Service Network (in green) include the market-specific adaptation of the integrated service shares and the handling of client orders including the individual PSS configuration. The Service Manager conducts comprehensive and frequent communication with customers and the PSS Provider about Service Engineering results; monitoring of the project's economy regarding development efforts and benefit for customer/revenues. The Service Designer reacts flexibly to short-term changes of customer and PSS Provider demands, even in late development phases. The Service Implementer plans the implementation of the services. The Customer plays another key role because demands towards the PSS will be drawn up and implemented based on the determined customer needs. Furthermore, he/she is the user of the PSS and gives feedback about quality. Finally, the Suppliers deliver the necessary materials, components and missing competencies to realize the product and service portfolio together with the PSS Provider.

About knowledge exchanges in PSS, it is worth to consider that PSS engineering is a dynamic process with fluctuating actors [80, 89], where the knowledge residing in individuals has to be combined with knowledge assets that are essential for creating the intended (customer) value and have to be shared between the roles, as centred in the so-called “2nd wave of knowledge management” [90]. While in “traditional” product development knowledge assets are mostly explicit and formalized in the form of documents, specifications and design etc. managed by applications such as CAD, PDM or PLM, during PSS engineering when the intangible aspects come into play knowledge is usually tacit, like skills, know-how, emotions and the like [84]. Explicit knowledge for PSS engineering includes market needs and customer requirements, product specifications and concepts, as well as the detailed product design or model [81]. This knowledge can be formalized in text documents, spread sheets, diagrams, CAD drawings and the like. However, only about 4% of organisational knowledge is formalized [91]. Recent studies on open innovation, e.g. in the form of application of crowdsourcing techniques [92] or implicit feedback leveraging from social media [93], have established the important role of open, crowd-oriented opinion and sentiment in enhancing products and services. This knowledge is mostly informal and unstructured, consisting of individual posts and discussions, ideas, comments and other interactions. Thus, it is difficult to codify and share, as it requires individual interaction to transfer. It is however equally important as knowledge for PSS engineering.

With respect to the ability to merge explicit knowledge from different domains, ontologies are capable in terms of multi-domain knowledge (e.g. Web Ontology Language—OWL [94]). Tacit knowledge, in the form of personal opinions and sentiments regarding PSS, poses extra challenges for the design and implementation of knowledge sharing. The informal nature of the relevant data and the inherent lack of formalization create additional issues [95]. The aspect of sharing explicit, formalized knowledge during PSS design is well covered with concrete approaches and frameworks in literature. Nemoto et al. described a framework to manage PSS design knowledge represented by five elements (core product, need, function, entity and actor) [79]. Zhu et al. and Zhang et al. formalized knowledge from previous PSS cases in a physical and a service model [81, 82]. Furthermore, Baxter et al. defined a KM framework for PSS design process knowledge, manufacturing knowledge, service design and service operations knowledge [31].

Concerning tacit or unstructured knowledge, some approaches can be found in literature, mainly on a conceptual basis. For instance, Bertoni emphasizes the importance of “bottom-up” knowledge sharing in PSS design and suggests Web 2.0 tools such as blogs, wikis or social networks to capture tacit and unstructured knowledge and tap into the “wisdom of crowds” [84]. This idea has been extended by Larsson et al. [77] into the concept of “Engineering 2.0”, applying easy to use technologies for knowledge sharing, while Chirumalla explored the use of Web 2.0 tools for knowledge sharing [78]. Also Natural Language Processing (NLP) has been validly adopted to enable formalization of tacit knowledge, according to a transdisciplinary approach [95]. A balance has to be found that supports a “bottom-up” knowledge sharing without sacrificing an efficient way to search and identify relevant knowledge. Bertoni and Larsson have identified seven barriers for knowledge sharing in PSS design, which have to be overcome: acceptability and self-censorship, commitment and reward, resignation, time loss, awareness, language and models, and trust [96].

Furthermore, it is important that all members of the development team have access to the same knowledge in the right form [97]. For explicit knowledge sharing, SysML seems to be appropriate to be extended for the purpose of modelling and exchanging PSS design knowledge, as it is an established standard for systems engineering. A key advantage lies in its extendibility. The needed meta-model layer is provided by default in the specification of UML itself. Hence, an adaption to domain specific needs can be performed. As it is not feasible for all stakeholders to use a common standard for knowledge representation or work with models from other domains, ontologies can be used to share knowledge across domains. However, modelling a proper ontology can become very complex, in particular if a generic ontological representation of a PSS is envisaged. The ontology needs to be filled with product and service related knowledge from different domains. To define service features and software elements demands for specific expertise not only from the field of product design but informatics, service etc. The interface to the knowledge base has to become user-friendly to ensure an acceptance by the end-user.

10.6 PSS Evaluation in Industrial Contexts

For manufacturing companies, the measurement of PSS performance is a crucial aspect to identify the best solution to provide on the market and able to satisfy the customer needs, improving the product value proposition. Moreover, identifying the best PSS offer allows improving the company business model, its business performance, and thus its revenues. In order to promote PSS performance measurement, two main principles must be taken into account: “what cannot be measured cannot be improved” [98] and the Plan-Do-Check-Act (PDCA) approach [99], since a continuous monitoring is required during the entire P-S lifecycle.

According to the authors’ perspective, PSS performance evaluation involves both the preliminary evaluation and the validation of the scenarios that will be implemented by the company. Indeed, the evaluation is determining whether the process in its entirety can yield an output that meets the desired requirements, while the validation is determining whether the process as implemented can yield an output that meets the specifications with acceptable capability. For validation, the process must be challenged using verified measurement systems. The areas investigated refer to the definition of a set of key performance indicators (KPIs), and the PSS assessment according to the performance achieved referring to the global system sustainability, according to its three main dimensions: economy, environment, and social well-being [100].

In literature, four different kinds of performance measures can be identified: Result indicators (RIs), Performance indicators (PIs), Key performance indicators (KPIs), and Key result indicators (KRIs). (K)RIs quantities the degree the company achieved its defined goals (they are measured over a long time period), while (K)PIs recognises the actions to do or that should be done in the future to increase the current performance and achieve the defined objectives (they usually are evaluated daily or at least weekly). KPIs in particular measure a current or future situation able to encourage the stakeholders to adopt any strategy in order to face up the scenario that arises. According to the aim of this research, KPIs are able to provide the guidelines to drive the company in the right business direction. Indeed, KPIs measuring the company performance regarding a certain business give information to company stakeholders during the decision-making process. Moreover, they are involved to discover what are the non-adding value activities (that approximately represent the 60% of a company's activities) inside a specific business [101]. Therefore, in order to identify the right KPIs to adopt for evaluating a certain business, literature proposes the adoption of the SMART principles, which are Specific, Measurable, Attainable, Realistic, and Time sensitive [102]. KPIs that comply with these five criteria allow companies assessing their real time performance, defining measures early enough before problems occur, and collecting the appropriate KPIs for PSS evaluation during the Design phase. This last one is a crucial aspect in PSS assessment, because the evaluation and validation of a new PSS offer during the design phase allows both reducing the time to market and successfully addressing the customers' needs. It is important to notice again that PSS assessment is different from product assessment, as in PSS, product characteristics and service functionalities influence one another. Currently in literature, few works about performance assessment in PSS exist; thus, it is an open issue. An interesting research was conducted by Mourtzis et al. [103], which classified KPIs with respect to the main PSS Design methods. Those classes involved in this paper are the following: Customers (C), Business (B), and Sustainability (S). Figure 10.2 shows how the KPIs classes refer to the related PSS design methods explained in the previous chapter. They are groups into three classes: B if referring to Business aspects, S if relating to Sustainability, and C if relating to the Customers. The main KPIs involved and the relative classes are listed in Table 10.1. Beyond the advantages that KPIs measurement offers to assess a PSS offer during the Design phase, some weaknesses remain. KPIs measurement demands lot of effort due to a frequent evaluation. For this reason, a critical aspect in the performance measurement system is to compare the value of an indicator with the effort required for its evaluation [104]. Furthermore, the number of indicators should be limited to ensure a meaningful overview of the current situation.

Generally a PSS will provide not only a higher customer satisfaction, but also a great advantage on the sustainability in respect to traditional products [105], according to its three main dimensions: economy, environment, and social well-being [100]. From the environmental viewpoint, PSS provides a more conscious product usage thanks to the service functionalities delivered, increasing resource productivity and a close loop-chain manufacturing. Moreover, because the PSS requires the involvement of different partners and stakeholders, they will deliver a solution able to create

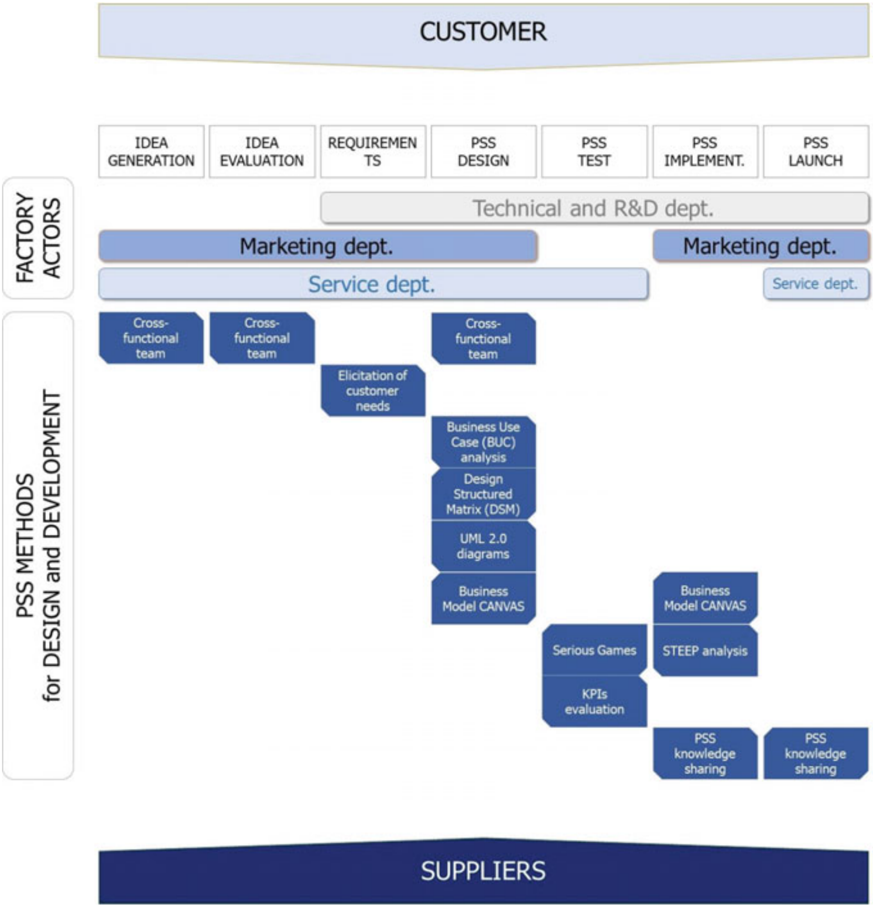

Fig. 10.2 Design and development framework for PSS

a sustainable supply chain, according to the service provided. From the economic viewpoint, PSS is able to create new market potentials and higher profit margins, and can contribute to higher productivity by means of reducing investment costs along the lifetime as well as reducing operating costs for the final users [35]. Finally, from the social viewpoint, PSS can build secure knowledge intensive jobs and contribute to a more geographically balanced distribution of wellbeing.

The PSS and the relative Servitization process extend the responsibility of the PSS provider to the whole lifetime of the product [106]. For this reason, it is required to perform assessment from a lifecycle perspective. In the manufacturing industry, product sustainability is calculated by adopting a lifecycle design approach: it allows quantifying product impacts and providing tangible commercial values in terms of efficiency and costs [107]. They are based on the definition of several indicators to assess the lifecycle performance and support comparative analysis. The technique to

Table 10.1 KPIs list for PSS performance evaluation

KPIs	Class	KPIs	Class
Customer satisfaction	C	Overall equipment effectiveness	B
Acceptability	C	Technical availability	B
Acceptance rate	C	Flexibility	B
Availability for production plan	C	Stability	B
Number of customer needs	C	Machine reliability	B
On-time delivery	C	Service reliability	B
Efficiency	C	Service assurance	B
Quality	B, C	Team qualification	B, C
Customer needs rate	C	Knowledge management	B
Requirement inconsistency	C	PS maintenance efficiency	B
Efficiency of collaboration	C	Development cost	B
Privacy	C	Service delivery costs	B
Product flexibility	C, B	Environmental quality cost function	S, B
Expansion flexibility	C, B	Energy efficiency	S
Sustainable product-service efficiency	S	Lease/reuse	S

support this described lifecycle design approach is defined as Life Cycle Assessment (LCA) according to ISO 14040:2006 (2006) [108] and it allows evaluating the environmental impacts on the product. Moreover, another lifecycle approach to assess the economic impact is Life Cycle Costing (LCC) [109], which has the scope to recognize all the economic impact during the product lifecycle. More recently, also the social impacts have been included in the lifecycle design approach by the so-called Social Life Cycle Assessment (SLCA) [110]. Such methods defined for product assessment could be "extended" and applied also to PSSs. However, the common indicators that assess economic, environmental or social domains separately will not approach and assess PSS sustainability in its complexity and wholeness. Indeed, the sustainability of a system cannot be assessed by the use of a single criterion mainly because of the intrinsic multidimensionality characteristic of sustainability. It is required to generate and assess a unique value that is the combination of all relevant criteria. In literature, some research calculates the sustainability of a PSS without adopting the lifecycle approaches, while other research has proposed to translate the lifecycle design approaches to assess the PSS sustainability demonstrating how to calculate the sustainability impacts of an integrated PSS by considering not only the impacts related to the product realization, usage and dismissing, but involving also the intangible assets and the ecosystem actors, as reported in Table 10.2. It shows several methods and relative indicators developed and used by different researchers in literature, which adopts a transdisciplinary approach. In the columns, the three main lifecycle indicators are identified (i.e. Environmental Impact, Economic Impact, and Social Impact), and also the integrated indicator to calculate the entire Sustainable impact.

Table 10.2 Lifecycle indicators in literature according to a transdisciplinary approach

References	Environmental impact indicator	Economic impact indicator	Social impact indicator	Sustainable impact indicator
[22, 111]	ENI (ENvironmental indicator) measured by Eco-indicator 99 point (EI-99)	ECI (EConomic Indicator) refers to all the lifecycle costs through the equivalent annual cash flow technique (EA)	SOI (SOcial Impact) considers separately Human Health contributions according to EI-99 methodology	SI = ENI + ECI + SOI (each indicator is normalized to obtained a monetary value (€), that is SI)
[112]	Total environmental impact along lifetime	Total lifecycle cost	None	None
[110]	Global warming potential	Lifecycle costs	QALY (measure of well-being)	QALY as a single-score alternative to direct monetisation
[72]	Life span, efficiency of resource consumption, closed cycle efficiency, and potentials for improvement	None	None	None

10.7 An Extensive Design and Development Framework for Industrial PSSs

On the basis of a critical analysis of the existing literature about the different aspects of PSS development, we can conclude that the design and development of successful PSS has to focus on the identification and interpretation of interactions between products and services to fully reflect stakeholder requirements. Design decisions cannot be merely technology-driven or manufacturing-related, but the user needs have to be the main focus. In this context, UCD techniques become a driving force. Thus, in the case of PSS, innovation relies on sharing knowledge between partners from different domains, maintain a common understanding of the design concept derived from customer needs and re-use experiences from other PSS projects; the usage of "downstream" knowledge from later phases of the life cycle and the inclusion of the customer into the design process is important as well. While in a conventional static OEM-supplier relationship contractual obligations set by the leading company define what and how knowledge is shared, such a model is not feasible for the dynamic collaboration required for PSS. Besides the missing lead-time required setting up such an arrangement, there might simply not be a partner powerful enough to impose its standards.

Based on this analysis, it can be stated that:

- Elicitation of customer needs is the key point and the first issues to solve in PSS design and development. It can be done by UCD techniques (ethnography, personas, role-playing);
- Technical analysis of PSS function can be executed by Business Use Case (BUC) analysis, which provides a user-centred investigation of the conceptual PSS model and a mapping of goal-oriented interactions between external actors and the PSS items;
- Design of the PSS functions by Design Structured Matrix (DSM);
- Definition of the PSS process model and system infrastructure by UML 2.0 diagrams and extended SysML models for systems engineering. The PSS meta-model layer is provided by UML, and extension of SysML allows exchanging PSS design knowledge, as it is an established standard. Tacit knowledge sharing can be supported using Web 2.0 tools for the PSS stakeholders on a dedicated social platform;
- Definition of the Business Model and involved stakeholders by CANVAS modelling;
- Identification of the new business strategies and trends by a simplified STEEP analysis;
- Management of the PSS knowledge sharing by formalization of explicit engineering knowledge and flexible exchange of unstructured and tacit knowledge between the stakeholders involved;
- Creation of cross-functional teams to foster knowledge sharing during the design phase, including people coming from the different functions, domains and organisations involved (i.e., stakeholders from product, service, or system integration). For this purpose;
- PSS model validation by Serious Games and hybrid PSS digital mock-ups in order to simulate the human-system interaction;
- Evaluation of PSS performances by proper key performance indicators (KPIs), investigating different areas such as Business, Sustainability and Customers. The latter include also the evaluation of the user experience by tests with samples users. Interactive prototypes can be adopted for this scope: high-fidelity mock-ups that combines realistic visualization (e.g., high quality aesthetic rendering, realistic environments, truthful use cases) with high level of interaction and simulation of the PSS behaviours according to the PSS conceptual model (e.g., movements of product parts according to some interaction with its interface or the service functions, real-time feedback connected to the service delivery).

The overall framework defined to support PSS design and development is represented in Fig. 10.2. It integrates the above-mentioned methods along the P-S lifecycle management (P-SLM) and considers the actors involved, as presented in Fig. 10.1.

In addition, Table 10.3 shows the selected KPIs for an overall evaluation of the PSS performances, according to the three identified areas: Business, Sustainability

Table 10.3 Selected KPIs for PSS evaluation

Area	KPIs	PSS methods for design and development
KPIs for business (B)	Overall equipment effectiveness	
	Technical availability	
	Flexibility	
	Stability	
	Machine reliability	
	Service reliability	IPS^2 models
	Service assurance	IPS^2 models
	Team qualification	
	Knowledge management	Layer-based methodology
	Knowledge sharing	Layer-based methodology
	PS maintenance efficiency	Lifecycle oriented approaches
	Development cost	Lifecycle oriented approaches
	Service delivery costs	Lifecycle oriented approaches
	Environmental quality cost function	Lifecycle oriented approaches
	Quality	Software tools
	Product flexibility	Software tools
	Expansion flexibility	Software tools
KPIs for sustainability (S)	Environmental quality cost function	Lifecycle oriented approaches
	Energy efficiency	Web-based models
	Lease/reuse	Web-based models
	Sustainable product-service efficiency	Lifecycle oriented approaches
	Sustainability assessment (SA)	Life cycle assessment (LCA) and costing (LCC)
KPIs for customer (C)	Customer satisfaction	Usability assessment
	Acceptability	Service engineering
	Acceptance rate	Service engineering
	Availability for production plan	IPS^2 models
	Number of customer needs	Requirements elicitation
	On-time delivery	Service engineering
	Efficiency	Usability assessment
	Quality	Software tools
	Customer needs rate	Requirements elicitation

(continued)

Table 10.3 (continued)

Area	KPIs	PSS methods for design and development
	Requirement inconsistency	Requirements elicitation
	Efficiency of collaboration	IPS^2 models
	Privacy	IPS^2 models
	Product flexibility	Software tools
	Expansion flexibility	Software tools
	Team qualification	
	Effectiveness	Usability assessment

and Customers. In respect to the state of the art, Business indicators have been extended to consider also knowledge management and knowledge sharing among all the stakeholders, while Sustainability indicators have been expanded according to the last researches in this field (as cited in Table 10.2) and Customers indicators have been integrated with usability, value and interaction indicators.

10.8 Summary and Outlook

This chapter discussed the most useful methods for PSS Requirement Elicitation (RE), design and development and proposes a structured way to manage such a complex and transdisciplinary process, with a special attention to transdisciplinary methods. The most significant RE approaches for PSS and the most successful design methods are presented, and numerous examples of transdisciplinary methods from recent literature are discussed. After that, the chapter focuses on knowledge management, which is a key aspect in PSS due to the increased complexity and the higher quantity of data and knowledge exchanged during PSS design and develop. The discussion highlights how to identify the stakeholders involved in the underlying processes, as relevant sources and targets of knowledge, and how to choose the best role models for orchestrating the contributions of various stakeholders in the innovation process. Finally, the measurement of PSS performance is presented as a crucial point to identify the greatest solution to provide on the market and able to satisfy the customer needs, improving the product value proposition. A set of Key Performance Indicators (KPIs) suitable for PSS are defined according to the reference area: business, sustainability, and customer. This overview about design methodologies and evaluation strategies allows having a high-level view and useful tools to develop a systematic PSS development framework, in order to help manufacturing enterprises to determine their own position on the PSS maturity scenario and to plan future actions toward servitization, according to knowledge exchange and transdisciplinary view [113].

References

1. Hauksdóttir D, Mortensen NH, Nielsen PE (2013) Identification of a reusable requirements structure for embedded products in a dynamic market environment. Comput Ind 64(4):351–362. https://doi.org/10.1016/j.compind.2012.10.008
2. Garetti M, Rosa P, Terzi S (2012) Life cycle simulation for the design of product-service systems. Comput Ind 63(4):361–369
3. Aurich JC, Fuchs C, Wagenknecht C (2006) Life cycle oriented design of technical product-service systems. J Clean Prod 14(17):1480–1494. https://doi.org/10.1016/j.jclepro.2006.01.019
4. Welp EG, Meier H, Sadek T, Sadek K (2008) Modelling approach for the integrated development of industrial product- service systems. In: Proceeding of 41st CIRP conference on manufacturing systems
5. Komoto H, Tomiyama T (2008) Integration of a service CAD and a life cycle simulator. CIRP Ann—Manuf Technol 57(1):9–12
6. Shimomura Y, Hara T, Arai T (2009) A unified representation scheme for effective PSS development. In: CIRP Ann—Manuf Technol, pp 379–382
7. Marilungo E, Coscia E, Quaglia A, Peruzzini M, Germani M (2016) Open innovation for ideating and designing new product service systems. In: Proceeding of 8th CIRP conference on industrial product service systems, Bergamo, Italy
8. Sakao T, Shimomura Y (2006) Service engineering: a novel engineering discipline for producers to increase value combining service and product. In: Huisingh D et al. (eds) J Clean Prod 15(6):590–604
9. Tomiyama A (2005) A design methodology of services. In: Samuel A, Lewis W (eds) DS 35: proceedings ICED 05, the 15th international conference on engineering design. Melbourn, Barton, pp 1970–2014
10. Komoto H, Tomiyama T (2009) Systematic generation of PSS concepts using a service CAD tool. In: Sakao T, Lindahl M (eds) Introduction to product/service-system design. Springer, London, pp 71–92
11. Sakao T, Shimomura Y, Sundin E, Comstock M (2009) Modeling design objects in CAD system for service/product engineering. Comput Aided Des 41(41):197–213
12. Shimomura Y, Arai T (2009) Service engineering—methods and tools for effective PSS development. In: Sakao T, Lindahl M (eds) Introduction to product/service-system design. Springer, London, pp 113–136
13. Chirumalla K, Bertoni A, Ericson A, Isaksson O (2013) Knowledge-sharing network for product-service system development: is it atypical? The philosopher's stone for sustainability. Springer, Berlin, pp 109–114. https://doi.org/10.1007/978-3-642-32847-3_18
14. Matzen D, Tan A, Andreasen MM (2005) Product/service-systems: proposal for models and terminology. In: Meerkamm H (eds) Design for X: Beiträge zum 16. Symposium. Erlangen: Lehrstuhl für Konstruktionstechnik, pp 27–38
15. Matzen D (2009) A systematic approach to service oriented product development. Stokkemarke: Scandinavian Digital Printing A/S, Ph.D. thesis
16. Tan A (2010) Service-oriented product development strategies. Scandinavian Digital Printing A/S, Stokkemarke
17. Peruzzini M, Marilungo E, Germani (2014b) Functional and ecosystem requirements to design sustainable P-S. In: Advances in transdisciplinary engineering, volume 1: moving integrated product development to service clouds in the global economy, proceedings 21st ISPE Inc. international conference on concurrent engineering (CE2014), pp 768–777. https://doi.org/10.3233/978-1-61499-440-4-768
18. Müller P (2013) Integrated engineering of products and services—layer-based development methodology for product-service systems. Fraunhofer IRB, Stuttgart
19. Miller D, Hope Q, Eisenstat R, Foote N, Galbraith J (2002) The problem of solutions: balancing clients and capabilities. Bus Horiz 45(2):3–12. https://doi.org/10.1016/S0007-6813(02)00181-7

20. Peruzzini M, Germani M, Favi C (2012) Shift from PLM to SLM: a method to support business requirements elicitation for service innovation in Product Lifecycle Management. In: Proceedings of IFIP advances in information and communication technology 388 AICT. Springer, New York, pp 111–123, https://doi.org/10.1007/978-3-642-35758-9_10
21. Wiesner S, Peruzzini M, Doumeingts G, Thoben KD (2012) Requirements engineering for servitization in manufacturing service ecosystems (MSEE). In: 4th CIRP IPS2 conference, Japan
22. Peruzzini M, Marilungo E, Germani M (2014a) A QFD-based methodology to support product-service design in manufacturing industry. In: Proceedings of 2014 international conference on engineering, technology and innovation: engineering responsible innovation in products and services ICE 2014, Bergamo, Italy, 23–25 June pp 1–7. https://doi.org/10.1109/ice.2014.6871572
23. Peruzzini M, Marilungo E, Germani M (2015) Structured requirements elicitation for product-service system. Int J Agile Syst Manag 8(3/4):189–218
24. Peruzzini M, Marilungo E (2016) User-centred approach for product- service design using virtual mock-ups. In: Proceedings 14th international design conference, DESIGN 2016, Cavtat, Dubrovnik, Croatia, 16–19 May 2016, in DS 84, pp 1805–1814
25. Mengoni M, Peruzzini M (2016) How to support the design of user-oriented product-related services. Proceedings 18th international conference on human-computer interaction, HCI international 2016, Toronto, Canada, 17–22 July 2016, in distributed, ambient and pervasive interactions, Lecture Notes in Computer Science, vol 9749, pp 103–110. https://doi.org/10.1007/978-3-319-39862-4_10
26. Nilsson P, Fagerström B (2006) Managing stakeholder requirements in a product modelling system. Comput Ind 57(2):167–177. https://doi.org/10.1016/j.compind.2005.06.003
27. Elgh F (2007) Modelling and management of manufacturing requirements in design automation systems. In: Loureiro G, Curran R (eds) Complex Syst Concurr Eng. Springer, London, pp 321–328
28. Rouse WB, Sage AP (2009) Handbook of systems engineering and management, 2nd edn. Wiley Series in Systems Engineering and Management. Wiley, Hoboken, NJ
29. Boehm B, Basili VR (2001) Top 10 list [software development]. Computer 34(1):135–137. https://doi.org/10.1109/2.962984
30. Nuseibeh B, Easterbrook S (2000) Requirements engineering. In: Finkelstein A (ed) ICSE'00 Proceedings of the conference on the future of software engineering, Limerick, Ireland, pp 35–46. https://doi.org/10.1145/336512.336523
31. Baxter D, Gao J, Case K, Harding J, Young B, Cochrane S, Dani S (2008) A framework to integrate design knowledge reuse and requirements management in engineering design. Robot Comput-Integr Manuf 24(4):585–593. https://doi.org/10.1016/j.rcim.2007.07.010
32. Laporti V, Borges MRS, Braganholo VP (2007) A collaborative approach to requirements elicitation. In: 11th international conference on computer supported cooperative work in design, Melbourne, Australia, pp 734–739. https://doi.org/10.1109/cscwd.2007.4281527
33. Azadegan A, Papamichail KN, Sampaio P (2013) Applying collaborative process design to user requirements elicitation: a case study. Comput Ind 64(7):798–812. https://doi.org/10.1016/j.compind.2013.05.001
34. Goedkoop MJ, van Halen CJG, Te Riele HRM, Rommens PJM (1999) Product service systems—ecological and economic basics
35. Baines TS, Lightfoot HW, Evans S, Neely A, Greenough R, Peppard J, Roy R, Shehab E, Braganza A, Tiwari A, Alcock JR, Angus JP, Bastl M, Cousens A, Irving P, Johnson M, Kingston J, Lockett H, Martinez V, Michele P, Tranfield D, Walton IM, Wilson H (2007) State-of-the-art in product-service systems. Proc Inst Mech Eng, Part B: J Eng Manuf 221(10):1543–1552. https://doi.org/10.1243/09544054JEM858
36. Meier H, Roy R, Seliger G (2010) Industrial product-service systems—IPS2 CIRP annals. Manuf Technol 59(2):607–627. https://doi.org/10.1016/j.cirp.2010.05.004
37. Pahl G, Beitz W (2007) Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung; Methoden und Anwendung, 7th edn. Springer-Lehrbuch

38. van Husen C (2007) Anforderungsanalyse für produktbegleitende Dienstleistungen. IPA-IAO-Forschung und Praxis, Nr. 458. Jost-Jetter, Heimsheim
39. Berkovich M, Leimeister JM, Krcmar H (2011) Requirements engineering für product service systems. Wirtschaftsinf 53(6):357–370. https://doi.org/10.1007/s11576-011-0301-3
40. Ehrlenspiel K (2007) Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit, 3rd edn. Hanser, München
41. Ulrich KT, Eppinger SD (2012) Product design and development, 5th edn. McGraw-Hill, New York
42. Ahrens G (2000) Das Erfassen und Handhaben von Produktanforderungen: Methodische Voraussetzungen und Anwendung in der Praxis
43. Lindemann U (2009) Methodische Entwicklung technischer Produkte. Springer, Berlin, Heidelberg
44. Liu X, Raorane S, Leu MC (2007) A web-based intelligent collaborative system for engineering design. In: Li WD, McMahon C, Ong SK, Nee AYC (eds) Collaborative product design and manufacturing methodologies and applications. Springer Series in Advanced Manufacturing. Springer, London, pp. 37–58
45. Murthy DNP, Rausand M, Østerås T (2008) Product reliability: specification and performance. Springer Series in Reliability Engineering. Springer, London
46. Berkovich M, Leimeister JM, Krcmar H (2009) Suitability of product development methods for hybrid products as bundles of classic products, software and service elements. In: ASME 2009 international design engineering technical conferences and computers and information in engineering conference, San Diego, California, USA, August 30–September 2, pp 885–894. https://doi.org/10.1115/detc2009-86939
47. Bullinger HJ, Scheer AW, Schneider K (2006) Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen, 2nd edn. Springer, Berlin
48. Bullinger HJ, Schreiner P (2006) Service Engineering: Ein Rahmenkonzept für die systematische Entwicklung von Dienstleistungen. In: Bullinger HJ, Scheer AW (eds) Service engineering. Springer, Berlin/Heidelberg, pp 53–84
49. Spath D, Demuß L (2006) Entwicklung hybrider Produkte - Gestaltung materieller und immaterieller Leistungsbündel. In: Bullinger HJ, Scheer AW (eds) Service engineering. Springer, Berlin/Heidelberg, pp 463–502
50. Ramaswamy R (1996) Design and management of service processes. Engineering process improvement series. Addison-Wesley Pub. Co., Reading, MA
51. Edvardsson B, Olsson J (1996) Key concepts for new service development. Serv Ind J 16(2):140–164. https://doi.org/10.1080/02642069600000019
52. Frietzsche U, Maleri R (2006) Dienstleistungsproduktion. In: Bullinger H-J, Scheer A-W (eds) Service engineering. Springer, Berlin/Heidelberg, pp 195–225
53. Kersten W, Kern EM, Zink T (2006) Collaborative service engineering. In: Bullinger HJ, Scheer AW (eds) Service engineering. Springer, Berlin/Heidelberg, pp 341–357. https://doi.org/10.1007/3-540-29473-2_14
54. Pohl K (2008) Requirements Engineering: Grundlagen, Prinzipien, Techniken, 2nd edn. Dpunkt-Verl, Heidelberg
55. Kotonya G, Sommerville I (1998) Requeriments engineering processes and techniques. Wiley, New York
56. van Lamsweerde A (2009) Requirements engineering: from system goals to UML models to software specifications. Wiley, Chichester, England, Hoboken, NJ
57. Hull E, Jackson K, Dick J (2005) Requirements engineering, 2nd edn. Springer, London
58. Aurum A, Wohlin C (2005) Engineering and managing software requirements. Springer, Berlin, London
59. Lanubile F (2009) Collaboration in distributed software development. In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Naor M, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, de Lucia A, Ferrucci F (eds) Software engineering. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp 174–193

60. Whitehead J (2007) Collaboration in software engineering: a roadmap. In: Future of software engineering, Minneapolis, MN, USA, pp 214–225. https://doi.org/10.1109/fose.2007.4
61. Li Z, Rahman QA, Ferrari R, Madhavji NH (2009) Does requirements clustering lead to modular design? In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Naor M, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, Glinz M, Heymans P (eds) Requirements engineering: foundation for software quality. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp 233–239
62. Lindahl M, Sundin E, Sakao T, Shimomura Y (2007) Integrated product and service engineering versus design for environment—a comparison and evaluation of advantages and disadvantages. In: Takata S, Umeda Y (eds) Advances in life cycle engineering for sustainable manufacturing businesses. Springer, London, pp 137–142
63. Aurich JC, Schweitzer E, Fuchs C (2007) Life cycle management of industrial product-service systems. In: Takata S, Umeda Y (eds) Advances in life cycle engineering for sustainable manufacturing businesses. Springer, London, pp 171–176
64. Elnadi M, Shehab E (2015) Main enablers and factors for successful implementation of lean in product-service systems. Int J Agile Syst Manag 8(3–4):332–354
65. Böhmann T, Langer P, Schermann M (2008) Systematische Überführung von kundenspezifischen IT-Lösungen in integrierte Produkt-Dienstleistungsbausteine mit der SCORE-Methode Wirtschaftsinf 50(3):196–207. https://doi.org/10.1365/s11576-008-0047-8
66. Vasantha GVA, Roy R, Lelah A, Brissaud D (2012) A review of product–service systems design methodologies. J Eng Des 23(9):635–659. https://doi.org/10.1080/09544828.2011.639712
67. Maussang N, Sakao T, Zwolinski P, Brissaud D, (2007) A model for designing product-service systems using functional analysis and agent based model. In: 16th international conference on engineering design (ICED'07), Paris, France
68. Morelli N (2006) Developing new product service systems (PSS): methodologies and operational tools. J Clean Prod 14(17):1495–1501. https://doi.org/10.1016/j.jclepro.2006.01.023
69. Freitag M, Kremer D, Hirsch M, Zelm M (2013) An approach to standardise a service life cycle management. In: Zelm M, van Sinderen M, Ferraira Pires L, Doumeingts G (eds) Enterprise interoperability. Wiley, Chichester, pp 115–126
70. Scheithauer G, Kett H, Kaiser J, Hackner S, Hu H, Wirtz G (2010) Business modeling for service engineering. In: Shin SY, Ossowski S, Schumacher M, Palakal MJ, Hung CC (eds) the 2010 ACM symposium, Sierre, Switzerland, p 118. https://doi.org/10.1145/1774088.1774113
71. Schweitzer E, Fiekers C, Möhrer J (2010) Realisierung investiver Produkt-Service Systeme. In: Aurich JC, Clement MH (eds) Produkt-Service Systeme. Springer, Berlin, Heidelberg, pp 95–116
72. Mont O (2002) Clarifying the concept of product-service system. J Clean Prod 10(3):237–245. https://doi.org/10.1016/S0959-6526(01)00039-7
73. Lindow K, Müller P, Stark R (2011) NEW job roles in global engineering-from education to industrial deployment. In: Proceedings of the 18th international conference on engineering design (ICED 11), impacting society through engineering design, vol 8: design education. Lyngby/Copenhagen, Denmark, pp 205–215
74. Abramovici M, Aidi Y, Jin F, Göbel JC (2012) Lifecycle management von Hybriden Leistungsbündeln. In: Meier H, Uhlmann E (eds) Integrierte Industrielle Sach- und Dienstleistungen. Springer, Berlin, Heidelberg, pp 265–284
75. Müller P, Stark RR, Fraunhofer IP (2014). Integrated engineering of products and services: layer-based development methodology for product-service systems. Fraunhofer Verlag
76. Clarkson J, Eckert C (2005) Design process improvement: a review of current practice. Springer, London
77. Larsson A, Ericson Å, Larsson T, Isaksson O, Bertoni M (2010) Engineering 2.0: exploring lightweight technologies for the virtual enterprise. In: Randall D, Salembier P (eds) From CSCW to web 2.0: European developments in collaborative design. Computer supported cooperative work. Springer, London, pp 173–191

78. Chirumalla K (2013) Managing knowledge for product-service system innovation: the role of web 2.0 technologies. Res Technol Manag 56(2):45–53. https://doi.org/10.5437/08956308x5602045
79. Nemoto Y, Akasaka F, Shimomura Y (2015) A framework for managing and utilizing product–service system design knowledge. Prod Plan Control 26(14–15):1278–1289. https://doi.org/10.1080/09537287.2015.1033493
80. Cedergren SI, Elfving SW, Eriksson J, Parida V (2012) Analysis of the industrial product-service systems (IPS2) literature: a systematic review. In: 2012 IEEE 6th international conference on management of innovation & technology (ICMIT 2012), Bali, Indonesia, pp 733–740. https://doi.org/10.1109/icmit.2012.6225897
81. Zhang D, Hu D, Xu Y, Zhang H (2012) A framework for design knowledge management and reuse for product-service systems in construction machinery industry. Comput Ind 63(4):328–337. https://doi.org/10.1016/j.compind.2012.02.008
82. Zhu H, Gao J, Cai Q (2015) A product-service system using requirement analysis and knowledge management technologies. Kybernetes 44(5):823–842. https://doi.org/10.1108/K-11-2014-0244
83. Wewior J (2015) Role-play based assessment of IPS2-specific intellectual capital. Proc CIRP 30:415–420. https://doi.org/10.1016/j.procir.2015.02.127
84. Bertoni M (2010) Bottom-up knowledge sharing in PSS design. A classification framework. In: Marjanović D (eds) Design 2010: DS 60: proceedings of the 11th international design conference, Dubrovnik, Croatia, Zagreb, pp 1461–1470
85. Romero D, Rabelo RJ, Molina A (2012) On the management of virtual enterprise's inheritance between virtual manufacturing & service enterprises: supporting "dynamic" product-service business ecosystems. In: 2012 18th international ICE conference on engineering, technology and innovation (ICE), Munich, Germany, pp 1–11. https://doi.org/10.1109/ice.2012.6297695
86. Stjepandić J, Verhagen WJC, Liese H, Bermell-Garcia P (2015) Knowledge-based engineering, In: Stjepandic J et al. (eds) Concurrent engineering in the 21st century: foundations, developments and challenges. Springer International, Switzerland, pp 255–286
87. Baxter D, Roy R, Doultsinou A, Gao J, Kalta M (2009) A knowledge management framework to support product-service systems design. Int J Comput Integr Manuf 22(12):1073–1088. https://doi.org/10.1080/09511920903207464
88. Kremer D, Leyh J (2010) Mit Technologietreiber-Rollen neue Technologien schneller in Produkte umsetzen B. Dworschak, & A. Karapidis, Professional Traing Facts, pp 49–74
89. Nergard H, Ericson A (2012) Changes in present product design—opportunities for industrial oriented research. In: 2012 IEEE 3rd international conference on cognitive infocommunications (CogInfoCom), Kosice, Slovakia, pp 499–503. https://doi.org/10.1109/coginfocom.2012.6422032
90. Mouritsen J, Larsen HT (2005) The 2nd wave of knowledge management: the management control of knowledge resources through intellectual capital information. Manag Account Res 16(3):371–394. https://doi.org/10.1016/j.mar.2005.06.006
91. Bell S (2006) Lean enterprise systems: using IT for continuous improvement. Wiley Series in Systems Engineering and Management. Wiley-Interscience, Hoboken, NJ
92. van den Ende J, Frederiksen L, Prencipe A (2015) The front end of innovation: organizing search for ideas. J Prod Innov Manag 32(4):482–487. https://doi.org/10.1111/jpim.12213
93. Budak C, Agrawal D, El Abbadi A (2011) Structural trend analysis for online social networks. Proc VLDB Endow 4(10):646–656. https://doi.org/10.14778/2021017.2021022
94. McGuinness DL, van Harmelen F (2004) OWL web ontology language overview. W3C Recommendation 10(10)
95. Nawroth C, Schmedding M, Brocks H, Kaufmann M, Fuchs M, Hemmje M (2015) Towards cloud-based knowledge capturing based on natural language processing. Procedia Comput Sci 68(2015):206–216
96. Bertoni M, Larsson AC (2010) Coping with the knowledge sharing barriers in product service systems design. In: Horváth I, Mandorli F, Rusak Z (eds) Tools and methods of competitive engineering: proceedings of the eighth international symposium on tools and methods of

competitive engineering—TMCE 2010, 12–16 April 2008, Ancona, Italy. Delft University of Technology, Delft, Netherlands, pp 903–914

97. Nonaka I, Toyama R, Konno N (2000) SECI, Ba and leadership: a unified model of dynamic knowledge creation. Long Range Plan 33(1):5–34. https://doi.org/10.1016/S0024-6301(99)00115-6
98. Gries B, Restrepo J (2011) KPI measurement in engineering design: a case study. In: Proceeding of international conference on engineering design, ICED11, vol 1, pp 531–537
99. Deming WE (1992) Quality, productivity and competitive position, MitCenter for Advanced Engineering Study, place
100. Adams WM (2006) The future of sustainability: re-thinking environment and development in the twenty-first century. In: Technical report, IUCN renowned thinkers meeting
101. Dombrowski U, Schmidtchen K, Ebentreich D (2013) Ballanced key performance indicators in product development. Int J Mater, Mech Manuf 1(1):27–31
102. Abramovici M, Jin F, Dang HB (2013) An indicator framework for monitoring IPS2 in the use phase. In: Product-service integration for sustainable solutions, pp 311–322. https://doi.org/10.1007/978-3-642-30820-8_27
103. Mourtzis D, Doukas M, Fotia S (2015) Performance indicators for the evaluation of product-service systems design: a review. In: Proceeding of IFIP WG 5.7 international conference, Tokyo, Japan, vol 460. https://doi.org/10.1007/978-3-319-22759-7_68
104. Kerzner HR (2013) Project management metrics, KPIs, and dashboards: a guide to measuring and monitoring project performance. Proj Manag J 43(2):102. https://doi.org/10.1002/pmj.21263
105. McAloone TC, Mougaard K, Restrepo J, Knudsen S (2010) Eco-innovation in the value chain. In: Proceedings of international design conference, Dubrovinik, Croatia
106. Aurich JC, Mannweiler E, Schweitzer E (2010) How to design and offer services successfully. Proc CIRP J Manuf Sci Technol 2(3):136–143. https://doi.org/10.1016/j.cirpj.2010.03.002
107. Jeswiet J (2009) A definition for life cycle engineering. In: Proceedings of 36th international seminar on manufacturing systems, Saarbrucken, Germany
108. ISO 14040:2006 (2006) Environmental management—life cycle assessment—principles and framework
109. Woodward DG (1997) Life cycle costing—theory, information acquisition and application. J Proj Manag 15(6):335–344
110. Weidema B (2006) The integration of economic and social aspects in life cycle impact assessment. Int J Life Cycle Assess 11(1):89–96. https://doi.org/10.1065/lca2006.04.016
111. Peruzzini M, Marilungo E, Germani M (2013b) Product-service sustainability assessment in virtual manufacturing enterprises. IFIP International Federation for Information Processing AICT 408, Proceedings 14th IFIP WG 5.5 working conference on virtual enterprises, PRO-VE 2013, Dresden (Germany), 30 September–2 October 2013, Camarinha-Matos LM, Scherer RJ (eds), pp 13–21. ISSN 1868-4238, ISBN 978-3-642-40542-6. https://doi.org/10.1007/978-3-642-40543-3
112. Kwak M, Kim H (2013) Economic and environmental impacts of product service lifetime: a life-cycle perspective. In: Proceedings of 5th CIRP international conference on industrial product-service systems, Bochum, Germany. https://doi.org/10.1007/978-3-642-30820-8_16
113. Wognum N, Bil C, Elgh F, Peruzzini M, Stjepandić J, Verhagen WJC (2019) Transdisciplinary systems engineering: implications, challenges and research agenda. Int J Agile Syst Manag 12(1):58–89