Ref. Ares(2016)5477490 - 21/09/2016

D2.1
Requirements Engineering Framework for Symbiotic P-S Systems (M4)

Document Owner:	BIBA
Contributors:	POLIMI, IAO, NTUA, INNO, IVLAB
Dissemination:	PUBLIC
Contributing to:	T2.1
Date:	2015-06-30
Revision:	1.0

Version history

No.	Date	Notes and Comments
0.1	2015-05-04	Deliverable structure and ToC
0.2	2015-05-11	First content
0.3	2015-06-05	Extended State-of-the-Art content
0.4	2015-06-19	Outline of RE Approach
1.0	2015-06-30	Full Version

Deliverable Peer Review Summary

ID	Comments	Addressed (✔) Answered (A)
1.		
2.		

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

This deliverable is part of PSYMBIOSYS WP2.1 “Collaborative Requirements Engineering Framework for Symbiotic Product-Service Systems”. It is the first deliverable of the work packageand defines a novel RE framework for specification of PSS in collaborative environments. According to the DoA, D2.1 “Requirements Engineering Framework for Symbiotic P-S Systems” has the following content:

“Cross-linking and integration of products and services require new RE approaches and tools. This task is based on the needs for a collaborative approach of all stakeholders involved in the entire product life cycle and aims to develop a RE framework that enables a symbiotic specification of PSS in a collaborative way along the whole value chain. In such a networked environment, information exchange between the stakeholders and domains is the key success factor. Therefore, the framework will feature pre-defined structures, formats and interfaces for communication to avoid information loss and delays.” (PSYMBIOSYS Consortium 2014)

The deliverable is organized in five distinct chapters, describing the role of RE in PSS engineering, analyzing the State-of-the-Art or RE approaches for products, services, software and product-service bundles and finally developing the RE framework for specification of PSS in collaborative environments. This framework, which is the main outcome of task 2.1, lays the foundation of the requirements analysis and up-take in task 2.2. The framework is designed to support the challenging requirement engineering process by delivering a set of tools that will allow the different stakeholder in involved in the process to communicate and define the end user requirements. The framework is specifically designed to meet the need of end-user that are not always aware of his/her needs for a complex system like a PSS, the framework is based on new innovative approaches for requirements elicitation, analyses, specification and evaluation. The methods which are therefore explored include among others Storytelling and Serious Gaming.

The structure of the deliverable is outlined below:

Chapter 1: Introduction to the topic, objectives, structure of the deliverable and approach.
Chapter 2: Description of the role of RE in SE, PSS components and stakeholders.
Chapter 3: State-of-the-Art of RE approaches and gap analysis.
Chapter 4: Development of the Collaborative RE framework for symbiotic PSS.
Chapter 5: Summary and Conclusions for the project.

1 INTRODUCTION

The underlying paradigm of the PSYMBIOSYS project is that *"services are becoming (…) [an] integral part of manufacturing industries' offer, able to differentiate the product value proposition, to take a significant part of the revenue stream and in some most disruptive cases even to substitute the traditional selling of products with more innovative service-oriented business models."* (PSYMBIOSYS Consortium 2014).

The project addresses five main dichotomies or tussles, which are preventing the adoption of service innovation by manufacturing companies in the EU (PSYMBIOSYS Consortium 2014):

1. The existence of walled gardens of **Product and Service** competencies,
2. the mono-directional transfer of information between **Design and Manufacturing** and other phases of the life-cycle in PLM systems,
3. the mediation between **Knowledge and Sentiment** to consider both for the design of innovative Product-Service Solutions (PSS),
4. the integration of **Service-Oriented and Event-Driven** architectures for new ICT systems, and
5. the balancing of **Business and Innovation** to reconcile profitable product Business Models with service innovation.

The aim of PSYMBIOSYS is to improve the competitiveness of European manufacturing enterprises by helping to overcome the five tussles mentioned above and reduce the time-to-market for innovative Product-Service Systems. To reach this goal, ten research objectives have been defined for the project (PSYMBIOSYS Consortium 2014):

- To collect and analyse the **requirements** of the end users towards PSYMBIOSYS results,
- to develop a **symbiotic design-manufacturing multi-directional collaboration**,
- to develop a **symbiotic product-service lifecycles concurrent engineering**,
- to develop a **symbiotic knowledge-sentiment harmonic innovation**,
- to implement a **symbiotic service oriented – event driven secure IT infrastructure**,
- to conceptualize a **symbiotic business-innovation strategy**,
- to integrate **IT platforms and tools for manufacturing intelligence**,
- to design **innovative value chain systems**, to experiment in **industrial value chains**, and
- to implement an **innovation and impact strategy**.

In order to achieve the project objectives and to overcome the five tussles for service innovation in manufacturing industries, a set of new models, methods, processes and tools will have to be developed in PSYMBIOSYS. These results ought to be aligned to the requirements of the project's end users' as well as with the manufacturing industry as a whole. Understanding the industrial partners and other affected stakeholders expectations, i.e. their underlying needs, and linking information from all phases of the product-service life-cycle to the development process is a prerequisite for successful solution engineering (Rouse, Sage 2009; Elgh 2007; Nilsson, Fagerström 2006). Inadequate Requirements Engineering (RE) is a main source for failure of development projects and leads to exceeding budgets, missing functionalities or even the abortion of the project (Hauksdóttir et al. 2013a). Often the relevance of appropriate requirements is underestimated, which in turn leads to errors in the requirements specification, not to mention the disregards of 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 (Boehm, Basili 2001). Therefore, RE is an important part of PSYMBIOSYS, organized in work package 2. This deliverable comprises the outcome of task 2.1, the PSYMBIOSYS framework for Requirements Engineering.

1.1 Objective of the Deliverable

The first two objectives of WP 2 are to develop a RE framework for specification of PSS in collaborative environments and to take up user scenarios and business requirements for symbiotic PSS engineering. The objective of this deliverable D2.1 is to present the initial results of WP 2 on a RE framework that enables the specification of the correct requirements for PSYMBIOSYS developments, reducing the risk of failure for PSYMBIOSYS as well as supporting the project end users when specifying requirements for innovative PSS.

1.2 Structure of the Deliverable

The deliverable comprises the following sub-sections, according to the objectives described in section 1.1 (see Figure 1):

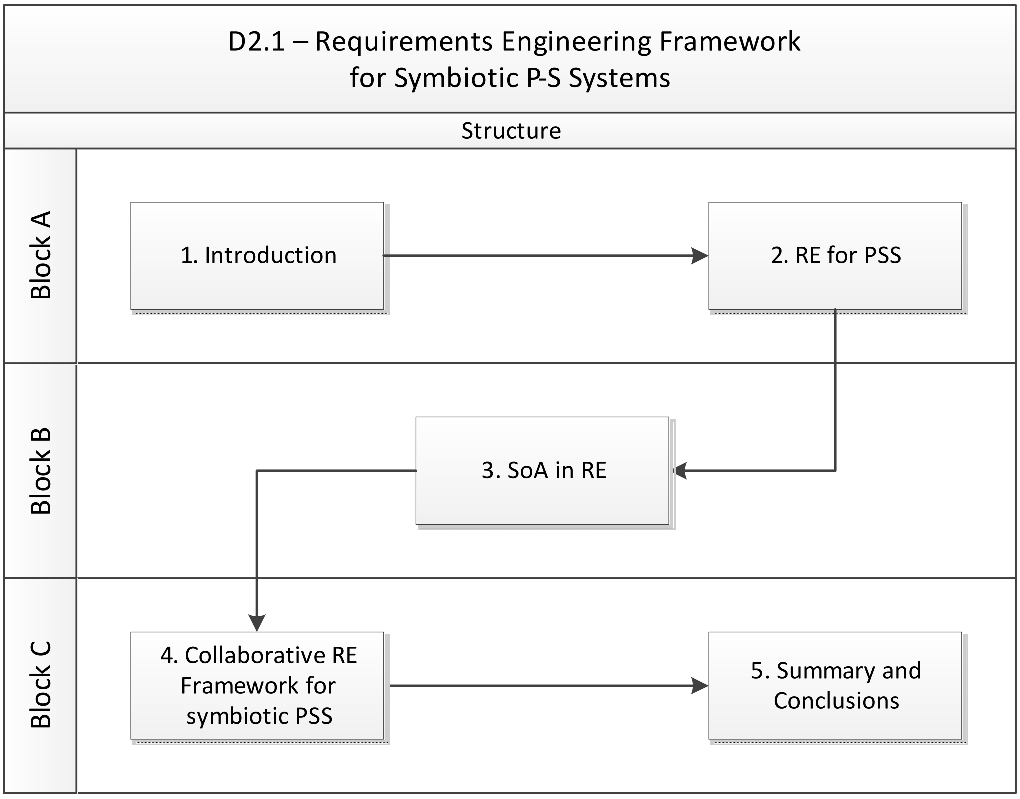

Figure 1: Structure of D2.1

Block A has the function to introduce the PSYMBIOSYS Requirements Engineering approach and integrate the topic into the project context. Taking into account the innovative nature of the project and the characteristics of PSS, the process of Requirements Engineering is presented. It consists of chapters 1 and 2 and presents the approach in the scientific context.

Block B contains the State-of-the-Art in Requirements Engineering for products, services, software and product-service bundles. Based on a literature review, gaps and requirements for the Collaborative RE Framework for symbiotic PSS are identified.

Block C describes the Collaborative RE Framework for symbiotic PSS derived from the State-of-the-Art analysis in chapter 3. Suitable methods and tools for requirements development and management are given. Finally, the results are summarized and conclusions for further research is given.

1.3 Approach for Framework development

For the development of the framework, we combine literature review and action research in a methodological approach. In a first step, the relevant characteristics of PSS and the RE process are identified. Secondly, there are several existing approaches that might serve as a good basis for the framework, thus these will be assessed for their suitability for the usage in PSS. The main assessment criteria are to what extent the existing framework, approaches and methodologies are transferable to PSS as well as to what extent it appears likely that their re-use or adaption will contribute to improved quality of RE. Thirdly, based upon this analysis, existing tools and framework for PSS can be further developed or changed in order to address these points.

For the literature review, scientific papers were accessed through suitable portals (Google Scholar, Elsevier, Researchgate etc.) searching for key words (Product-Service, PSS, Extended Products, collaborative networks, Requirements Engineering, dynamic systems, RE + PSS) (Kitchenham et al. 2009). The relevance of the identified papers for this article was based on assessing the abstract, as well as by searching for the combination of RE and PSS in the full papers. The main outcome of the literature research is discussed in chapter 3.

2 Requirements Engineering for Product-Service Systems

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 Requirements Engineering (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"* (Nuseibeh, Easterbrook 2000).

2.1 The Role of RE in Systems Engineering

This section summarizes the findings of the literature research related to Requirements Engineering as a discipline and highlights its growing importance and interrelationships with Systems Engineering and the system life cycle.

2.1.1 Existing approaches

In traditional development approaches, mainly from the manufacturing domain, RE is seen as a discrete development phase. This sequential approach has also been utilized in the first software development approaches, termed "Waterfall Model" (Royce 1970). It has the objective of creating a requirements specification, which serves as a reference for the subsequent development activities, as shown in Figure 2:

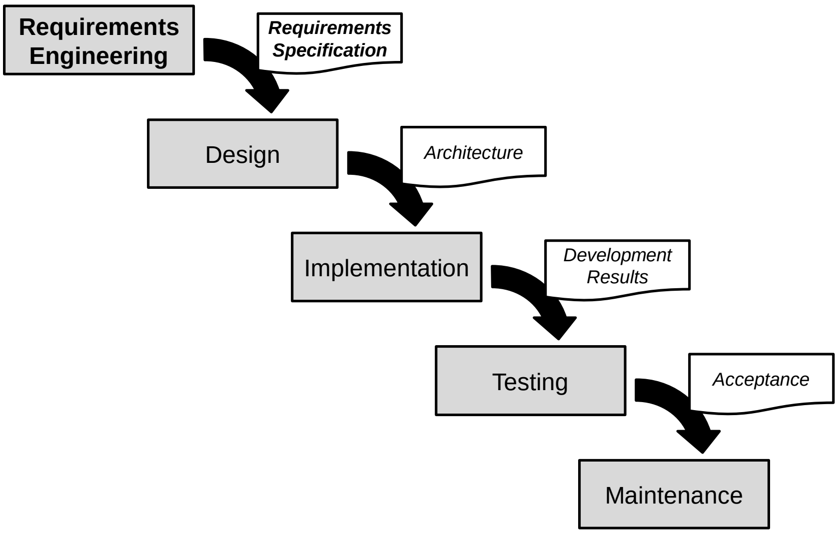

Figure 2: RE in the Waterfall Model

The traditional view of RE as a discrete phase in the development of a system has substantial disadvantages when dealing with the increasing complexity, dynamics and time constraints of current development projects. 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 (Baxter et al. 2008), increasing the overall development time.

Systems Engineering deals with the development of complex solutions, consisting of a large number of components interacting with each other (Fuxin 2005), such as in PSS. Thus, this implicates additional challenges for RE, which cannot be covered through the Waterfall Model. An example illustrating this challenge is that in traditional RE scenarios for simple products, the stakeholders are in general 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. (Laporti et al. 2007)

Complex systems 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 spatial 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 (Azadegan et al. 2013) and conflicts as well as interdependencies have to be assessed for a larger number of requirements.

However, competition also demands for faster time to market, in spite of increasing system complexity. At the same time, falling prices lead to cost pressure. The system has to fulfill all agreed requirements, notwithstanding shorter development times and cost restrictions. These challenges favour errors in Requirements Engineering for complex systems, leading to risks for the development process. A weak definition of requirements can slow down system development and induce unnecessary costs for design changes (see Figure 3). Incorrect requirements can results in an unsuitable system architecture and implementation can result, so the system may have missing or wrong functionalities.

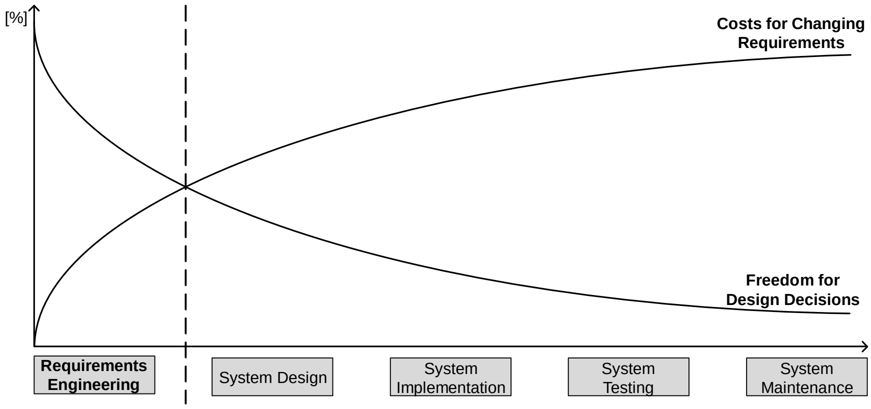

Figure 3: Cost for Changing Requirements vs. Freedom for Design Decisions

Besides having been able to describe the challenges mentioned above, the literature review revealed that approaches to overcome these challenges exist. An approach widely used in Systems Engineering is the “V-Model”. The term describes a family of models used to illustrate the software or system development process and its main steps (Rausch, Broy 2006). Here, RE continues along the development process of a system and secures a consistent and traceable elicitation and management of requirements. In concordance with the principles of concurrent engineering, there is an ongoing interaction between RE and the development phases in Systems Engineering, as shown in Figure 4 below:

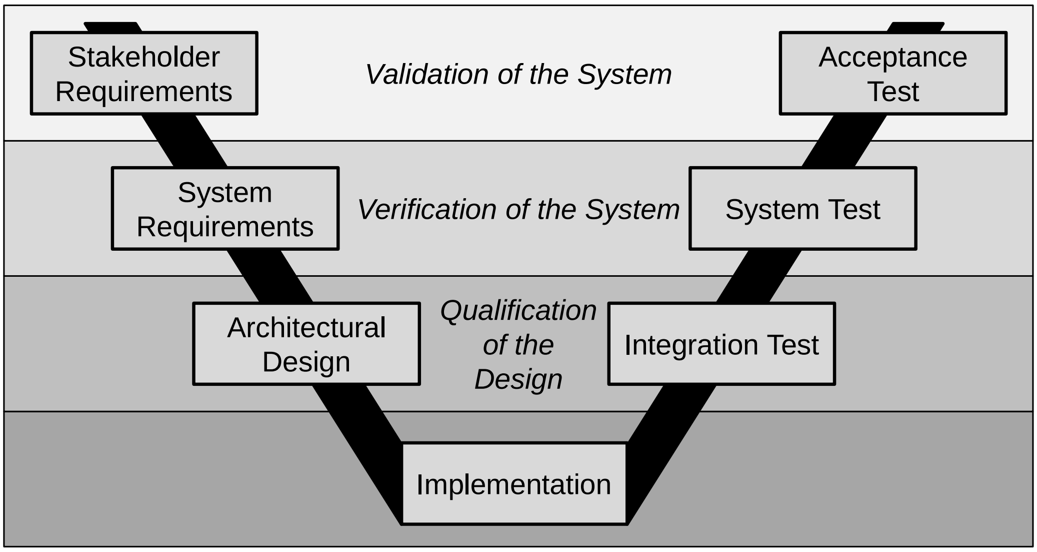

Figure 4: Requirements Engineering in the V-Model, according to (Hull et al. 2005)

The separate layers show the activities performed during the individual phases of system development. The collection of stakeholder requirements and their subsequent decomposition into system requirements and the architectural design of the solution is depicted on the left side of the "V". After system implementation, the right side of the "V" illustrates the test of the development results against the original specification. While the integration test qualifies the correctness of the architectural design, the system test verifies the compliance of the whole system to the technical specification. The final acceptance test validates that the system really meets the needs of the stakeholders. (Hull et al. 2005)

2.2 Characteristics of a PSS

In order to select or develop a suitable RE approach for PSS, it is necessary to identify the specific characteristics of such systems. In the following, some widely used definitions of PSS found in the literature are analysed for these characteristics.

"A Product Service system (PS system) is a ***marketable set of products and services*** *capable of* ***jointly fulfilling a user's need****. The PS system is* ***provided by either a single company or by an alliance of companies****. It can enclose products (or just one) plus additional services. It can enclose a service plus an additional product. And* ***product and service can be equally important for the function fulfilment****. The researcher's need and aim determine the level of hierarchy, system boundaries and the system element's relations."* (Goedkoop 1999)

"A PSS is an ***integrated product and service offering*** *that delivers* ***value in use****. A PSS offers the opportunity to* ***decouple economic success from material consumption*** *and hence reduce the environmental impact of economic activity. The PSS logic is premised on utilizing the knowledge of the designer-manufacturer to both increase value as an output and decrease material and other costs as an input to a system."* (Baines et al. 2007)

"*'An Industrial Product-Service System is characterized by the* ***integrated and mutually determined planning, development, provision and use of product and service shares*** *including its immanent* ***software components*** *in Business-to-Business applications and represents a* ***knowledge-intensive socio-technical system****'*

This means in detail:

- *An IPS[2] is an* ***integrated product and service*** *offering that delivers values in industrial applications.*

- *IPS²is a new product understanding consisting of integrated* ***product and service shares****.*
- *IPS²comprises the* ***integrated and mutually determined planning, development, provision and use****.*
- *IPS²includes the* ***dynamic adoption of changing customer demands and provider abilities****.*
- *The partial* ***substitution of product and service shares over the lifecycle*** *is possible.*
- *This integrated understanding leads to* ***new, customer-adjusted solutions****.*

IPS²enable ***innovative function-, availability- or result-oriented business models****.*” (Meier et al. 2010)

The analysis of the above definitions 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

In order to provide an integrated RE approach reflecting these characteristics, it is important to describe the identified components in more detail. This is done in the next section.

2.3 PSS Components

The PSS definitions analysed confirm that the main components to be covered in an integrated approach are tangible product and intangible service shares. However, IT components, such as software, are becoming more and more important for PSS. Therefore, they are described as a separate component below.

2.3.1 Tangible Product

Definition

“A product can be defined as an end item that will satisfy the customer’s needs. Key features of a product include:

- *It can consist of more than one physical entity,*
- *It can be a tangible entity that will be used or consumed by the customer. In addition to any physical product, a software program is also considered to be a tangible product.* […]
- *A product has definable attributes that can be classified under time, cost, quality, functionality, etc.”* (Thoben et al. 2001)

“A product is a tangible commodity manufactured to be sold. It is capable of falling onto your toes and of fulfilling a user’s need.” (Goedkoop 1999)

Benefits that are offered to the customer in the form of physical goods are referred to as tangible products and are characterized by their physical nature (Specht et al. 2002). They have a describable, technical function, which determines the application for a particular purpose or use for specific activities (Volz 1997). According to the degree of integration, it is distinguished between customized and standardized products. Standardized products are usually made without any customer involvement in the form of mass production (Meier et al. 2005). Due to the increasing demand for customer individual solutions, in addition to the standardized product customized products emerged (Hippel 2005). The customer is integrated into the creation process and his individual needs and requirements are taken into account, for example, by providing a customized machine. A tangible product can be described by a product model, which models the product components (Kraemer, Zimmermann 1996). Here, the tangible product is divided into its individual modules as well as in the associated items. The listed parts are in turn the necessary materials and resources to be allocated.

Product Engineering

In the discipline of Product Engineering several development models have been established to support the developers in the tasks of planning future development processes, orientate within the current development processes, and reflect on completed development processes. An ideal development process for products does not exist due to varying and wide-ranging influence factors on the development environment and process, e.g. the nature and complexity of the product being developed, available resources or time (Lindemann et al. 2011).

The planning of development processes is structured into three levels – the strategic planning, the operative planning and the goal level. The level of strategic planning comprises tasks of the project management – rather abstractly planned subprojects and project phases, e.g. in form of flow charts or procedure models for the whole product development process. On the level of operative planning, detailed and concrete steps of a procedure or for the individual product development phases are provided. The goal level contains the results of the operative planning, e.g. requirements lists, product models or design concepts.

The VDI-2221 guideline (VDI, 1993) structures the product development process into four phases on the strategic level:

Figure 5: Phase model of VDI 2221 (According to (VDI, 1993))

In a more recent product development process by (Unger, Eppinger 2011), the authors have added the phases of ‘integration and test’, as well as ‘release’ following the RE process models of software engineering. Tasks of the ‘integration and test’ phase are for example reliability testing, performance testing, obtaining of regulatory approvals or the implementation of design changes (Tingström et al. 2006).

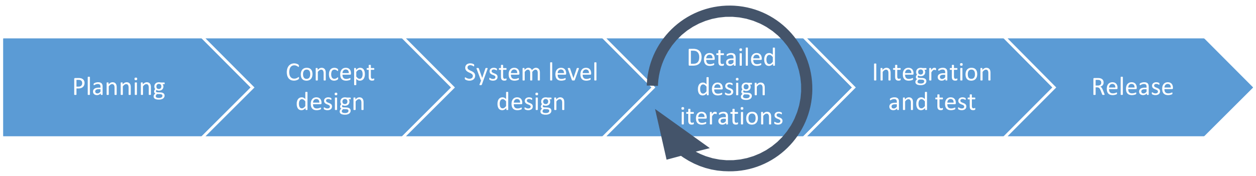

Figure 6: General staged product development process (According to (Unger, Eppinger 2011))

Once a development phase is completed it is “difficult or expensive to go back” (Unger, Eppinger 2011). The authors introduce alternatives to the staged product development processes such as the spiral model due to the advantage of being more flexible and open to feedback.

The phases are assigned to the procedures of the operative level and the results (goal level). The illustrated procedures show a generic product development process based on the outputs at the goal level; the outputs represent increasingly concretized partial models of the product (Lindemann et al. 2011).

The illustrated models comprise the same development phases and thus the same task packages concerning requirements. Consequently, the illustration of the generic staged product development process is sufficient for the identification of requirements engineering tasks embedded into the development process in the discipline of product engineering.

The stages and associated results described in the procedure by (Pahl, Beitz 2007) allow drawing conclusions towards requirements engineering tasks performed during the development process such as the discovering, documenting, verifying and tuning, and maintaining of requirements. During the clarification of the task stage, the market, the business situation and the environment are analyzed. Broadly speaking, information about the requirements and constraints are collected in order to generate and select product ideas and create a requirements specification. On the basis of the specification the requirements are concretized to enable developing of concept variants. The concepts are evaluated against technical and economic criteria, in other words, validated against the requirements and needs. The specification is adapted iteratively during the whole development.

2.3.2 Intangible Services

Definition

*"A service is an **activity or series of activities** of more or less **intangible** nature that normally, but not necessarily, take place in **interactions between the customer and service employees** and / or **physical resources or goods and / or systems** of the service provider, which are provided as solutions to customer problems."* (Grönroos 2007)

*"A service is an **activity (work) done for others** with an economic value and often done on a commercial basis. In this project, we include work done by **human beings as well as by automated systems**."* (Goedkoop 1999)

*"A service is an **activity or benefit** that one party can offer to another that is essentially **intangible** and does not result in the ownership of anything. Its production may or **may not be tied to a physical product**."* (Kotler, Bloom 1984)

Services include the intangible benefits for the customer. According to DIN ISO 9000, a service is an activity at the interface between supplier and customer with the purpose of fulfilling a customer need (DIN EN ISO 9000:2005-12). In this respect, in addition to the intangible nature services are characterized through the direct or indirect involvement of the customer in the process. The customer as an external factor is always part of the development or execution of the service. In addition, the synchronization of service production and service consumption is a characteristic of a service. This means that at the time the service is provided, the satisfaction of the customer needs is fulfilled. (Haller 2015)

The degree of customer integration also distinguishes between standardized or customized services. A standard service is provided with no customer integration, i.e. rendered uniform by predefined customer requirements. This includes insurance, which is offered to customers in a standardized form. The individual services include, among others, consultancy services that are targeted to individual customer needs (Meier et al. 2005).

In contrast to the modeling of tangible products, services cannot be divided in assemblies or individual parts. For a comprehensive description of services, three dimensions are considered which contribute to the modeling of a service. These include resources that are required for implementing the service, such as equipment, human labor, materials, information and intellectual property. After compilation and combination of resources, a process with a particular sequence of activities transfers these resources to the customer. The result of the service is described by the final resource combination. This can emerge, for example, in the form of a product, service sales or as an intellectual property transfer. (Bullinger et al. 2006)

Service Engineering

Service engineering is a complex development process with influences from various disciplines – engineering sciences, business economics, psychology and computer sciences. Services can be developed systematically; standardized procedures as they are common in engineering sciences can be of advantage. Moreover, the efficiency of services can be increased by adoption of component-based development procedures or variant concepts. On the one hand, computer sciences support the development process by providing software for the service development. On the other hand, in most projects of service development changes to existing or the development of new information and communication systems are required. Hence, the development discipline for service engineering is to be established analogous to product and software development. (Bullinger et al. 2006)

For the development of services, (Schneider et al. 2006) identified three different kinds of development process models: sequential phase models, iterative models and prototyping models.

As an exemplary development process of services, the development process model proposed by (Bullinger, Schreiner 2006) is illustrated in (Figure 7). The development process model is both phase-oriented and iterative, and comprises six phases. In the start phase, service ideas are generated. During the analysis, customer requirements as well as requirements involving the provider of the service are captured and represent the basis for the evaluation of the service ideas. In addition, the ideas can be refined if they do not fulfill the desired needs stated through the requirements. Methods are applied to elicit requirements such as surveys, interviews and customer feedback. In the design phase, individual specifications on potential (role models), process (flow charts), goal (product/component models) and market level (marketing concepts) are specified and integrated into a general specification. To create the specifications a variety of methods can be used, e.g. capacity planning to determine the required resources, modeling techniques to design the components of the service and the interaction and interfaces between the service provider and customer. During the preparation phase the resources which are required to realize the service are allocated. Tests of the specifications are carried out to find possible weaknesses (Bullinger, Schreiner 2006).

The model allows for iterative development, additionally the phases are not necessarily executed in the illustrated order. Modifications and changes are an essential element of the development process. Consequently, the model enables returns to previous phases (Bullinger, Schreiner 2006).

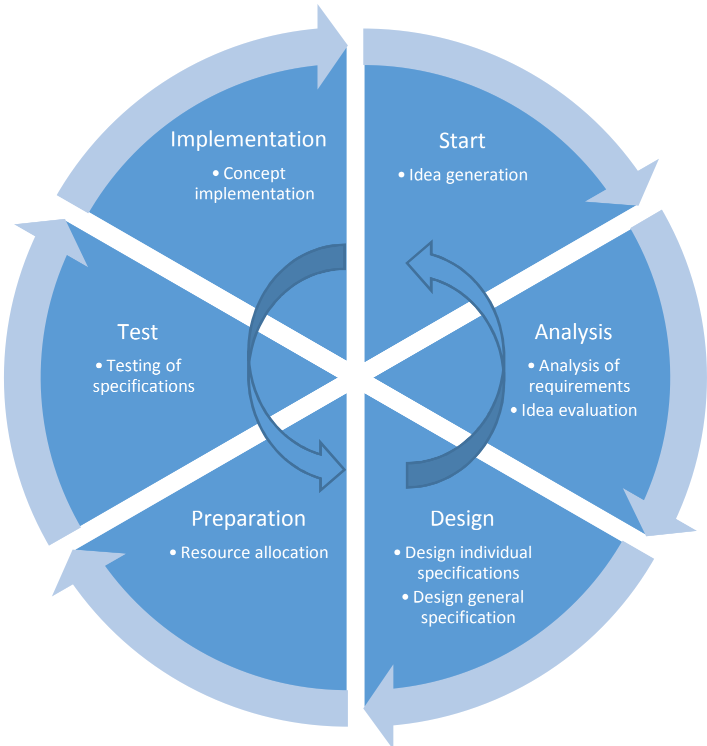

Figure 7: Development model for a systematic service development according to (Bullinger & Schreiner, 2003)

In terms of RE, on the one hand, activities to identify, analyze and validate requirements are provided. The identification of requirements is conducted in the start and analysis phases. The requirements are analyzed in the analysis and design phases by determining whether the requirements are complete in terms of fulfilling the needs during the analysis phase, and by performing various modeling activities during the design phase. Requirements are validated during the analysis and test phases. On the other hand, the requirements are documented in form of specifications and concretized by assigning them to the respective dimension of potential, process, goal or market.

2.3.3 IT Components

Definition

"[Information Technology] *is commonly used as a synonym for* ***computers and computer networks****, but it also encompasses other* ***information distribution technologies*** *such as television and telephones. Several industries are associated with information technology, including computer* ***hardware, software****, electronics, semiconductors,* ***internet****, telecom equipment,* ***engineering****, healthcare, e-commerce and* ***computer services****."* (Chandler, Munday 2011)

"(IT) Applied ***computer systems*** *- both* ***hardware and software****, and often including* ***networking and telecommunications****, usually in the context of a business or other enterprise."* (Howe 1985)

Software Engineering

In *software engineering*, RE is widely recognized as a special discipline; RE "has begun to evolve from its traditional role, as a mere front-end in the software development lifecycle, towards becoming a key focus in the software development process" (Aurum, Wohlin 2005).

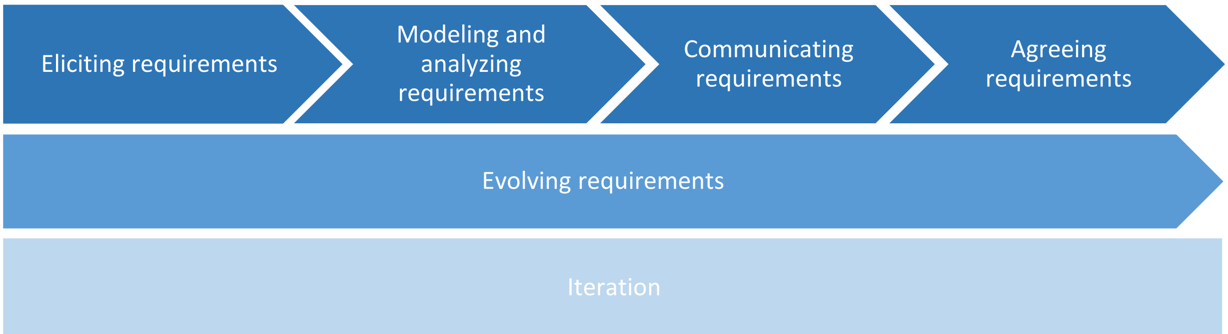

Figure 8: Requirements Engineering (according to (Nuseibeh, Easterbrook 2000))

In (Nuseibeh, Easterbrook 2000), the authors describe the core RE activities as the following:

- Eliciting requirements
- Modeling and analyzing requirements
- Communicating requirements
- Agreeing requirements
- Evolving requirements

The elicitation of requirements is the process of defining the problem that needs to be solved, and furthermore define how the problem is to be solved. One of the first activities is to identify and agree on the boundaries of the solution – the product, defining where it will be deployed in the operational phase. The stakeholders have to be identified, "individuals or organizations who stand to gain or lose from the success or failure [of the product]" (Nuseibeh, Easterbrook 2000). Stakeholders are on the one hand the customers of the product and on the other hand the developers and users. The needs of the stakeholders have to be identified and goals – “objectives a system must meet” (Nuseibeh, Easterbrook 2000) – have to be elicited.

The second core activity, modeling and analyzing of requirements, describes the activities of requirement concretization – resulting in domain specific requirements – and the identification and resolution of conflicts between requirements. The needs and goals of the stakeholders, even of different users, may vary and conflict. They may also be difficult to articulate or be constrained by a variety of factors. Therefore, the consistency of the requirements has to be analyzed to identify conflicts (Berkovich et al. 2011). Additionally, different modeling approaches can be applied to get a better understanding of the products’ environment and gather further information, e.g. enterprise modeling. Enterprise modeling is used to identify the purpose of the product by analyzing the organizational structure and behavior of the enterprise where the product will be used (Nuseibeh, Easterbrook 2000).

Communicating requirements comprises the documentation and management of requirements. Defining a specification language and notation – formal, semi-formal or informal, logic or natural – helps to ensure that the requirements can be “read, analysed, (re-)written, and validated” (Nuseibeh, Easterbrook 2000). The requirements documentation is the basis for the communication with the stakeholders. A crucial part of the RE process is the requirements management, this involves the identification of requirement interdependencies in order to trace the requirements. Ensuring the readability and traceability of the requirements is a key factor.

Agreeing upon requirements with all stakeholders can be a difficult task because of the different viewpoint each stakeholder may have. The stakeholders have different ways of expressing themselves, different goals and priorities. Conflicting requirements are negotiated and resolved. The established requirements are the basis for the validation of the product against the initial requirements (Nuseibeh, Easterbrook 2000).

The fifth core activity, evolving requirements, describes the change of requirements due to changes of customer needs or progressing development of the product. Change has to be recognized “through continued requirements elicitation, re-evaluation of risk, and evaluation of systems in their operational environment” (Nuseibeh, Easterbrook 2000).

2.4 PSS Stakeholders

One of the most important steps in PSS development is determining what the stakeholders actually expect from the system. Thus it is essential to identify all persons and institutions involved in the development, construction and the use of the PSS and document their requirements (Hass et al. 2008). According to (Pohl 2008): *“A Stakeholder of a system is a person or organization, who has a potential interest on the future system and therefore normally makes demands on the system. In doing so, one person can represent the interest of several persons or organizations, that means one person can assume more than one role.”*

The involvement of a stakeholder can be direct, like in the case of the users of the intended PSS, or indirect, such as the investors financing the development. These stakeholders are “*the actual information providers for aims, requirements and boundary conditions, whose management is a factor of success for the satisfaction of the customer needs*” (Rupp 2009).

To carry out the activities of the PSS development process, an integration of external competencies will take place, which participate in the form of various actors (Schweitzer et al. 2010). Within the PSS project management, the PSS-specific actors are determined and compiled in development teams. Also PSS specific roles are defined. The **PSS provider** represents the center of all those involved and acting along the entire PSS development process. The duties of a PSS provider include the coordination and execution of design, development and manufacture of contributions, as well as planning and development of complementary services (Müller et al. 2014). For efficient development of the overall PSS, value networks are formed, within the development of the individual PSS components takes place. The value networks can be divided into production and service networks.

The **production network** comprises various PSS suppliers who are responsible for the provision of materials, parts and components or system modules for the PSS provider (Mont 2002). In the **service network**, distributors, subsidiaries and service partners are summarized, which appear mostly in the form of material and service specialists. The main task of the service network comprises PSS sales, which includes the market-specific adaptation of the integrated service shares and the handling of client orders and the individual PSS configuration (Aurich et al. 2007).

Next to the PSS provider, the **customer** has another key role. He is considered especially in the early stages of development as the initiating actor, because based on the determined customer needs, demands on the PSS are derived and implemented (Schweitzer et al. 2010). A PSS specific role is the PSS project manager. He acts in the development process in various phases and performs management activities. The main tasks of the PSS project manager include the connection establishment between the PSS project management and the development process. In addition, he coordinates the PSS actors and their communication and networking along the development process. (Abramovici et al. 2012)

The **PSS architect**, as another PSS specific role is characterized by its PSS specific knowledge and the overarching effectiveness in PSS development. 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. (Lindow et al. 2011)

All parties involved in the PSS development process must communicate for different reasons in different phases with each other. According to the respective phases, thus there is a different distribution of tasks, competencies and responsibilities as well as a changing communication needs (Müller et al. 2014).

3 STATE-OF-THE-ART IN REQUIREMENTS ENGINEERING

This chapter reflects the requirements engineering methodologies for products, software, services as well as for integrated products and services.

A systematic internet search and literature research has been performed on the basis of ‘Google Scholar’ and ‘Springer Link’ to firstly find literature on the state-of-the-art of requirements engineering in reference to the domains of products, software, services and integrated products and services in order to get a better understanding of the methodologies and terminology of the respective domains. Secondly, frequently referenced materials were included in the literature analysis, and the findings have led to an extended search with on the one hand domain specific keywords, on the other hand the consideration of the collaborative aspect.

3.1 RE for Products

For 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 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 (Pahl, Beitz 2007). Sometimes, also aspects of requirements management are adopted, but without explicit instructions for implementation (van Husen 2007).

The following literature discussing the product engineering state-of-the-art and approaches has been analyzed:

- Requirements Engineering for Product Service Systems – A State of the Art Analysis (Berkovich et al. 2011):
    - Vorgehenszyklus für die Lösungssuche (Ehrlenspiel 2007)
    - Engineering Design (Pahl, Beitz 2007)
    - Methodik zum Entwickeln und Konstruieren technischer System und Produkte (VDI, 1993)
    - Product Design and Development (Ulrich, Eppinger 2012)
    - Erfassen und Handhaben von Produktanforderungen (Ahrens 2000)
- Entwicklung technischer Produkte (Lindemann 2009)
- Collaborative Product Design (Liu et al. 2007)
- New Product Development (Murthy, D. N. P et al. 2008)

The analysis of the future development environment is commonly discussed among the analyzed 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. (Berkovich et al. 2009)

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 analyzed 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” (Berkovich et al. 2011). Furthermore, the procedures described are not applicable to services due to focus on quantification of requirements. In (Ahrens 2000), the author analyzed the approach of (Pahl, Beitz 2007) 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 (Ahrens 2000).

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 (Lindemann 2009), 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 (Berkovich et al. 2011). (Ahrens, 2000) argues that procedures are provided to structure requirements lists either by thematic affliction or alternatively by the product structure in the methodology of (Pahl, Beitz 2007). Requirements validation is mentioned in the analyzed approaches but not discussed in detail. Generally, validation is done through evaluation of design drawings by the customer.

In (Berkovich et al. 2009), the authors state that “the customer plays a central role during the entire development process”. 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 (Berkovich et al. 2011). In (Murthy, D. N. P et al. 2008), the authors argue that manufactures are attempting to reduce costs by “increasing the use of the same parts, or modules, across different products”. Procedures are not explicitly mentioned.

(Liu et al. 2007) describe 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. The web-based collaborative engineering design systems enable the developers from different locations and businesses to share and integrate design models, e.g. the collaborative modeling 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 analyzed methodologies.

3.2 RE for Services

In the service area, models for the systematic development of services have been created (Bullinger et al. 2006). However, no systematic procedures for the implementation of RE have been established, 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” (Spath, Demuß 2006).

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

- Requirements Engineering for Product Service Systems – A State of the Art Analysis (Berkovich et al. 2011):
    - Design and Management Service Processes (Ramaswamy 1996)

  - Key Concepts for New Service Development (Edvardsson, Olsson 1996)
  - Ein Rahmenkonzept für die Entwicklung von Dienstleistungen (Bullinger, Schreiner 2006)
  - Dienstleistungsproduktion (Frietzsche, Maleri 2006)
  - Anforderungsanalyse für produktbegleitende Dienstleistungen (van Husen 2007)
- Service Engineering (Bullinger, Scheer 2006)
- Collaborative Service Engineering (Kersten et al. 2006)

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 2007) analyzed 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.

The initial requirements are concretized “by assigning them to quantifiable attributes related to the implementation” (Berkovich et al. 2011) 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 2007), only (Ramaswamy 1996) provides a detailed design process procedure including the activities of prioritization of requirements according to their importance, specification of attributes which are required to fulfill the needs, and creating a link between the attributes and requirements.

The identification and resolution of conflicts is not described explicitly in the service engineering approaches. In the analyzed approaches it is suggested to use the procedures known from software and product engineering. The approaches analyzed by (Berkovich et al. 2011) provide 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.

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 analyzed by (Berkovich et al. 2011).

3.3 RE for IT

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 (van Husen 2007).

The following literature discussing the software engineering state-of-the-art and approaches has been analyzed:

- Requirements Engineering for Product Service Systems – A State of the Art Analysis (Berkovich et al. 2011):
  - Requirements Engineering Framework (Pohl 2008)

   - Requirements Engineering Process (Kotonya, Sommerville 1998)
   - Requirements Engineering Process (van Lamsweerde 2009)
   - A Generic Process for Requirements Engineering (Hull et al. 2005)
   - Engineering and Managing Software Requirements (Aurum, Wohlin 2005)
- Requirements Engineering (Hull et al. 2005)
- Collaboration in Distributed Software Development (Lanubile 2009)
- Collaboration in Software Engineering (Whitehead 2007)

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 analyzed 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 (Berkovich et al. 2011).

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. 2011) 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. In (Hull et al. 2005), the authors describe the utilization of traceability in detail. Interdisciplinary traceability is not considered. It is recognized that requirements can change during any lifecycle phase and changes have to be captured and analyzed 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" (Berkovich et al. 2011). Change management is described as important during the whole lifecycle including the use phase of the product (Hull et al. 2005).

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 fulfillment of the stakeholder needs. Validation procedures are discussed in detail in the software engineering approaches. The validation focuses solely on software engineering (Berkovich et al. 2011). 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. In (Li et al. 2009), the authors 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".

In (Lanubile 2009), the author describes the state-of-the-art of collaboration in the software domain, focusing on the collaboration between software engineers. 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 mentions knowledge centers as web-enabled tools to share knowledge. The necessity of additional competencies is mentioned by (Lanubile 2009). He argues that "the quality of programmers is the most important factor in software work" and thus, developers are hired regardless 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 described by (Lanubile 2009) and (Whitehead 2007) refers solely to the software domain. Nevertheless, collaborative networks are created.

3.4 RE Approaches for Product-Service Systems

Deducted from the findings in chapter 2, 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 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.

The following literature discussing the state-of-the-art and approaches of integrated product and service development has been analyzed:

- Requirements Engineering for Product Service Systems – A State of the Art Analysis (Berkovich et al. 2011):
  - Integrated Product and Service Engineering versus Design for Environment (Lindahl et al. 2007)
  - Vorgehensmodell der hybriden Produktentwicklung (Spath, Demuß 2006)
  - Life Cycle Management of Product-Service Systems (Aurich et al. 2007)
  - Rahmenkonzept zur Entwicklung von Product-Service Systems (Botta 2007)
  - Systematische Überführung von kundenspezifischen IT-Lösungen in integrierte Produkt-Dienstleistungsbausteine mit der SCORE-Methode (Böhmann et al. 2008)
- Review of PSS Design methodologies (Vasantha, Gokula Vijaykumar Annamalai et al. 2012)
- State-of-the-art in product-service systems (Baines et al. 2007)
- PSS Design (Maussang et al. 2007)
- Developing new product service systems (Morelli 2006)

According to (Berkovich et al. 2011), 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 analyzed 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 (Berkovich et al. 2011). They argue that identification and resolution of conflicts is only mentioned briefly in the analyzed 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 described in detail in any of the approaches (Berkovich et al. 2011). The validation of requirements "is not discussed in detail" in the analyzed approaches. The importance of customer integration in all lifecycle phases is recognized. However, specific methods or procedures are not specified. Modularization is widely mentioned in literature, e.g. in (Böhmann et al. 2008) the authors discuss the modularization of standardized solutions.

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 described (Vasantha, Gokula Vijaykumar Annamalai et al. 2012).

Due to the nature of integrated products and services the necessity of additional competencies is recognized. (Aurich et al. 2007) “have noted the importance of extended value creation networks”. 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” (Vasantha, Gokula Vijaykumar Annamalai et al. 2012).

3.5 Assessment of existing RE Approaches and Methods

The literature describes the complexity and the challenges of RE for PSS. The temporary involvement of stakeholders in different parts of a PSS lifecycle contributes to an even more challenging and dynamic process. Current methods, approaches and tools for Systems Engineering, in particular for Requirements Engineering, such as stated in section 2.1.1 and **Error! Reference source not found.**, do not provide full support for these new challenges. Current RE approaches are not able to handle the large number of different and conflicting requirements without exponentially increasing time and cost, as contradictions and interdependencies have to be assessed for a large number of requirements in various domains (Jarke et al. 2011).

An assessment of the State-of-the-Art for RE identifies the following strength and weaknesses:

RE Task	Criterion	State-of-the-art research on product engineering methodologies	State-of-the-art research on software engineering methodologies	State-of-the-art research on service engineering	State-of-the-art research on methodologies for integrated P&S
Elicitation		●●◐○	●●◐○	●◐○○	●◐○○
	Procedures to identify stakeholders are provided.	●	●	●	●
	Procedures to identify the customer's value chain are provided.	◐	◐	○	○
	Procedures for the elicitation of requirements are provided.	●	●	◐	○
	Interdisciplinary requirements are considered in the procedures.	○	○	○	◐
Concretization		●○○	●○○	●○○	◐○○
	Procedures to translate the initial requirements into design specific requirements are provided.	◐	◐	◐	○
	Procedures to prioritize requirements are provided.	◐	◐	◐	○
	Interdisciplinary requirements are considered in the procedures.	○	○	○	◐
Conflicts		●○	●○	◐○	◐○
	Procedures for the identification and resolution between requirements are provided.	●	●	◐	
	Procedures for the identification and resolution of requirements across different domains are provided.	○	○	○	○
Documentation		●●○○	●●◐○	●◐○○	●◐○○
	Procedures to document requirements are provided, e.g. requirement lists.	●	●	◐	●
	Procedures to trace design requirements to initial stakeholders' requirements are provided.	●	●	◐	◐
	Procedures for change management are provided.	○	◐	◐	○
	The traceability procedures enable cross-domain tracing. Change management procedures are considered for the utilization phase.	○	○	○	○
Validation		◐○	●○	◐○	◐○
	Procedures for validation of the design requirements against the initial stakeholder requirements are provided.		●	◐	◐
	The procedures consider multiple domains.	○	○	○	○
Customer & Stakeholder Integration		●◐○	●●○	●●○	●●○
	Procedures for elicitation provide methods to integrate customer requirements – even if indirectly through marketing or distribution.	●	●	◐	●
	The customers and stakeholders are actively integrated in the requirements elicitation.	◐	●	●	●
	Procedures and methods for collaboration with the customer and stakeholders are provided for all lifecycle phases of the EP.	○	○	◐	○
Modula-rization		◐○	●○	●○	◐○
	Modularization is supported.	◐	●	●	
	Modularization and interfaces for cross-domain modules are supported.	○	○	○	○
Collabora-tion across domains			◐○		●○
	Procedures or methods for interdisciplinary development are mentioned.		◐		●
	Procedures for interdisciplinary development between product engineering, software engineering and service engineering are described explicitly; interfaces are specified.		○		○
Collaborative Environment		●○○○	●◐○○		●●◐○
	The necessity of additional competencies to realize a product is recognized. Other business partners contribute to the development in a cooperative way.	◐	◐		●
	Methods or procedures to integrate and share processes, knowledge and information are provided to enable collaboration between two or more business partners.	◐	◐		◐
	Additionally, methods or procedures are provided to identify and select possible collaboration partners.	○	○		◐
	Consideration of collaborative networks such as the virtual enterprise etc.	○	◐		◐

Figure 9: Analysis of RE State-of-the-Art

The next section will carry out a more detailed analysis and elaborate the gaps. The requirements derived will form the boundaries of our framework.

3.6 Gaps and Requirements for the RE Framework

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, the 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 the 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 the basis for the 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.

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 modeling 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 sum it up, 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 fulfill the requirements for a successful realization of such a complex solution. Especially the lack of a holistic, interdisciplinary view and the corresponding interfaces should 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 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 life-cycle, 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 behavior. 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 life-cycle 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.

In the next step we are therefore looking at how existing approaches contribute to that and which gaps we can identify. To manage the complexity of PSS, a modular approach would lead to a better usage of resources. Based on current approaches, this can be realized by strengthening the reusability of components (Alter 2012) and the collaboration between process steps (Blanchard 2004). Thus, a RE framework would have to allow the specification of the PSS components from different domains together with their interaction, as well as the modular reuse of requirements from previous PSS project, which could be based on the work of (Hauksdóttir et al. 2013b).

In order to include the distributed stakeholders from different domains, participatory approaches without the need for specific PSS models can provide a common basis to dynamically exchange requirements. A framework for the specification of the targeted PSS on different levels in SysML and UML is provided by (Penzenstadler, Eckhardt 2012), which is suitable to exchange information between the technical design departments of stakeholders, if they agree to adhere to a common standard. For non-technical stakeholders, such as the end user however, it might be too difficult to use. A way to transform requirements between natural language and formal models is offered by (Wiesner et al. 2014), which might be a good extension to bridge this gap.

Considering the approaches identified above, information exchange between different disciplines can be addressed by either implementing a common standard for all stakeholders instead of the domain specific models, or using natural language for information exchange. However, it might not be feasible to implement a common standard for all the stakeholders and the use of natural language might produce ambiguity problems. Thus, a translation might be needed between domain specific models on lower levels and a common model on higher levels. Summarizing the findings, the following requirements for an RE framework for PSS can be described:

- Complexity: The RE framework has to be able to handle a large number of interoperable product, service and IT components that create the desired functionality through emergent behaviour.
- Distributed stakeholders: The RE framework has to be applicable for a large number of stakeholders that are typically separated spatially and organizationally. This involves the user, who defines the scope and purpose of the PSS, as well as specialized partners with distinct processes, which develop the individual system components.
- Different disciplines: The RE framework has to be usable in multiple disciplines with own formalisms and tools. Information and requirements exchange between the disciplines has to be supported in order to create a common view of the targeted system.

Based on the requirements identified in the analysis above, three main objectives for the RE framework can be defined:

1. The development of a reusable requirements structure for PSS, which allows the definition of modular components, which can be combined to the overall system. To support modular design and interoperability of PSS, standardized abstractions and architectures are needed. This also refers to verification and validation of the system at the design stage.
2. Definition of a PSS content model that is formal enough to describe the targeted system unambiguously and in a verifiable way, but has the flexibility to include non-formal inputs. The different stakeholder needs and changing requirements must be managed by RE processes. As the functionalities of a PSS emerges from the cumulative interactions of the product and service components, RE methods and tools have to be able to manage emergent effects with predictable results.
3. The inclusion of the most relevant domain specific models into the PSS content model, e.g. using semantic mediation. It needs to integrate mechanical and service engineering models for the formal description of requirements, which have to be mapped to the PSS elements and communicated between stakeholders from different disciplines.

4 Collaborative RE Framework for Symbiotic PSS

A PSS RE framework can be outlined based on the requirements and objectives identified in the previous section. The framework will cover the problem domain and the solution domain as two distinguished areas in the PSS development process. The problem domain describes the needs and business goals as well as their formulation into stakeholder requirements for the PSS, without prescribing the technical means to fulfil them. The business vision of the PSS end user is formalized in business requirements, mostly given in natural language. They will be documented e.g. in Business Process Model and Notation (BPMN) or data flow diagrams that show the difference between the "as-is" and "to-be" business processes. The stakeholder requirements are derived from the differences between the two situations and further statements of need in a non-technical way, like use scenarios.

In the solution domain, system requirements are derived, which describe the functionalities of the PSS and the architectural design used to achieve the system requirements. This can be done by developing system models, describing the targeted PSS functionality and documenting system requirements. These capture the solution for the customer in technical terms and allow to estimate the cost and schedule required to build the PSS. The architectural design of the PSS identifies the different product and service components and shows how they work together to meet the system requirements. These models can be documented in SysML or domain specific notation.

Furthermore, a translation between the different domain specific models must be established, as it might not be feasible to introduce SysML or another common notation for all stakeholders. The envisioned framework covers this in two different ways: For the semi-automatic translation between non-formal (natural language) and formalized notations, Natural Language Processing (Wiesner et al. 2014) can be implemented. Transformation between the different formal domain specific models can be achieved using methods such as semantic mediation, which enable conversion between model using ontologies. In this way, domain barriers can be greatly reduce or fully removed.

4.1 Structure of the RE framework

The RE framework can be divided into sequential requirements development activities and cross-sectional requirements management tasks, as illustrated in Figure 10:

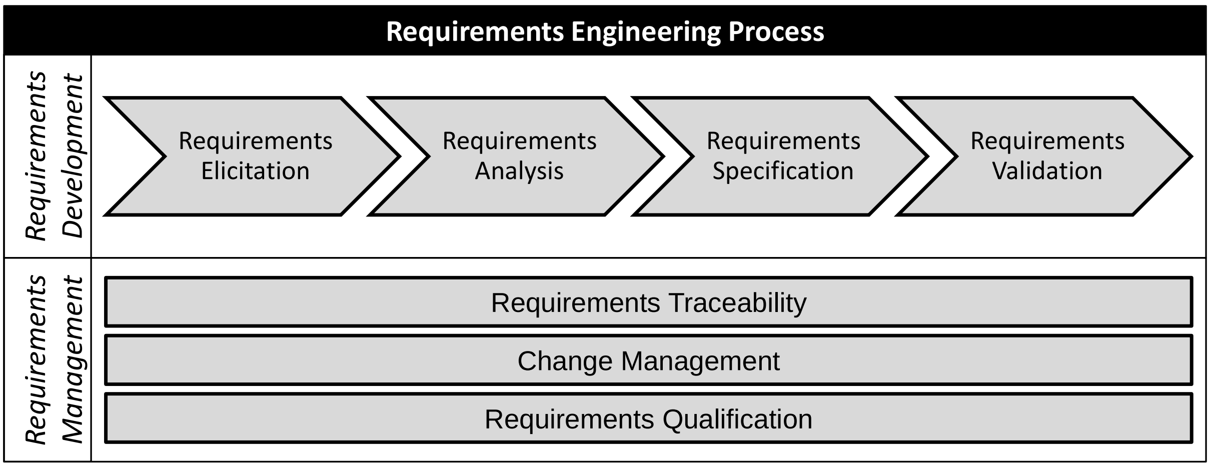

Figure 10: RE Framework

The following sections will describe the activities involved in discovering, documenting and maintaining a set of requirements for the development of a PSS in the RE framework for the identified stakeholders and domains.

4.2 Requirements Development

In requirements development, stakeholder and system requirements are elicited, analyzed, and established. Therefore, requirements for the PSS are identified, requirements expression and conflicts are checked, the requirements are documented in a specification and validated with the stakeholders (CMMI Product Team 2010). The underlying steps of elicitation, analysis, specification and validation are detailed below.

4.2.1 Elicitation

Requirements elicitation determines the relevant requirements for the PSS that fulfils the goals of the stakeholders. Thus, all the stakeholder identified for the PSS have to be consulted during requirements elicitation and many of the requirements can vary or even be opposite and contradictory. Various techniques are traditionally used to capture the requirements from them, such as scenario exploration, interviews, questionnaires and many more (Laporti et al. 2007). For PSS as a complex system, the ideas to solve a totally new problem might be unclear. Thus, requirements elicitation has to improve precision, accuracy and variety of requirements. In the RE framework for PSS, this is supported through the following innovative approaches:

As described in the previous chapter, it is still a challenge within the requirement elicitation to define the requirements clear, correct and complete. One reason for this is that those delivering the requirements and those developing the products and services based on these requirements have different perceptions of terminology and understanding of wording. In addition, in most cases, the requirements are for future products and services for which the customer might not have a clear idea on how the product or service will look like (Gausepohl 2008). This leads to uncertainties and inconsistencies in the requirements, as well as to misunderstanding of what is actually meant with a specific requirement. Thus, in order to reduce the risk of capturing false requirements for the Product-Service System, end user requirements capturing can use the story telling method. This method is not new, actually it has been used for some decades for this purpose (Ribeiro et al. 2014). “Storytelling is a flexible design research method with a broad range of applications, associated processes and variations” (Vink 2015). It is based on communication between the stakeholders. It allows the participants to develop the story commonly (group storytelling) and thus discuss the different perspective as it is carried out. A narrative story telling helps to put the requirement in a specific context clarifying to the reader how they are to be understood. In addition, a different advantage for the users are that it is a natural way of communicate among human beings, and thus reduces the barriers to articulate what they think they need using their natural language and by generating a common understanding.

Bearing in mind that a PSS is something new for several SMEs and we cannot foresee exactly what services a customer would request in the future, it is essential for the sustainability of the solution that it can handle the dynamicity of the system. This requires that the requirements are defined in such a way that they are to some extent flexible. This flexibility needs to be covered already from the beginning, and it requires that for future PSS ideas are generated. This is a very crucial but challenging step in the early stage of innovation projects. This phase is often called the “fuzzy front-end of innovation”. Even though this stadium of an innovation project is loosely structured, the generation of good and realizable ideas is mandatory. The question for the requirements elicitation process is therefore how to support this? The idea generation process seems to have many similarities with processes based on serious games. This approach has been used in previous projects and also in other fields. Furthermore, communication is of high relevance. Therefore we were looking at methods that support the communication and collaboration. This is a typical advantage of multi-player games. The final criteria for why incorporating the use of games in elicitation is the possibility to use game mechanics as motivation factors as well as to provide the user with immediate feedback on how the decision or a change in a variable will impact on the PSS configuration. The latter will be used for the validation step.

4.2.2 Analysis

In requirements analysis, the quality of the elicited requirements is evaluated, if it is *"necessary, verifiable and reachable"* (Ebert 2012). Hidden or latent interrelations between requirements or missing assumptions during analysis have to be discovered to remove obstacles to requirements satisfaction. Also, ambiguity, inconsistency or incompleteness have to be avoided or removed in a second round of elicitation. In the RE framework for PSS, this is supported through the innovative modelling approaches (Shen et al. 2004):

Business modelling is essential for developing PSS. Several modelling methods can be used, which are all well documented in the literature. In our framework we have analysed the different methods by looking at how these fits into the other methods we are using, like storytelling and serious gaming. Thus, a very suitable method is to use a scenario based method again, which would be in line the storytelling approach. For this we have therefore selected the take-up of the as-is scenario (analysing the current running processes) and then deduct the to-be scenario. The gap analysis between the as-is and to-be is then the basis for the requirements.

Industrial companies are often used to processes and to identify involved stakeholders. As described above, in order to capture the right requirements it is essential that these very different stakeholders communicate with each other (through the story telling) and that they understand each other. An old method that helps in no only identifying the human stakeholders, but also system restrictions and other mechanisms (including machinery, software etc.) is IDEF0. "IDEF0 is a standard modelling method used to establish function models, which has already been accepted by most experts and end-users in this field. It was derived from a well-established graphical language, the Structured Analysis and Design Technique (SADT), and has only two types of graphic notation, the activity box and boundary/interface arrow. Diagrams are formed based on the Inputs-Controls-Outputs- Mechanisms (ICOM) Code and there are strict syntax and semantic rules, which ensure that the model is described precisely" (Shen et al. 2004). We have therefore combined the storytelling with the IDEF0 generation in a workshop setting, in which the requirements engineer could establish the as-is and to-be scenarios. In parallel, the process modelling was done and all stakeholders and restrictions were captured. This can be mapped into a game, and in a second iteration the users can play with the modelled to be scenario in order to validate the relevance of each requirement.

4.2.3 Specification

The requirements specification provides the developers with a complete description of the functionality of the PSS to be developed. Therefore, also non-functional requirements and constraints are included. A hierarchical structure can provide sections and subsections for different levels of requirements, such as in the approach by (Hauksdóttir et al. 2013a). In the RE framework, the hierarchy is based on the modeling approach used for PSS development, e.g. usage scenarios or data flow diagrams.

The RE framework features a specific requirements data sheet, which is used to document the stakeholder requirements in a format compatible to establish RE tools. The PSS requirements sheet summarizes all information necessary for PSS development. This includes information on the single PSS components. To avoid ambiguity, basic information like ID no., requirements title and description are recorded. Furthermore, the associated PSS component is assigned, together with the source of the requirement. In order to represent the interactions between the PSS components, information on the connections between the components is recorded as well to serve as reference for requirements dependencies later. This targets the involved disciplines and also the individual actors. Documentation has to establish the connection from the single requirement towards the whole PSS project, including information on the degree and moment of fulfilment.

4.2.4 Validation

It has to be confirmed that the requirements specification meets the needs of the customer and other identified stakeholders. The completeness and correctness of the determined requirements is checked during requirements validation to ensure that the documented requirements accurately express the stakeholder's needs (Hull et al. 2005). Therefore, the RE framework involves the PSS stakeholders in the review of the requirements during validation.

In our framework, we use serious gaming as a part of how to collect the requirements. We collect the data and store them in a DB, but interesting from the validation point of view is the feedback mechanism within the game play. Using a simulation based game, mirroring the as is and to be scenarios of the pilots, the user can play with different configurations and see the outcome. This will provide a base for assessing the relevance and the importance of specific variables and will be used to validate the requirements in a playful way.

4.3 Requirements Management

Requirements engineering has a strong interdependency with management in order to control the PSS development and validate the final product. Thus, managing requirements of the development project and PSS components and ensuring alignment between those requirements and the project plan and work results is the purpose of requirements management. This includes tracing of requirements during the different phases of the PSS life-cycle, change management and qualification of the development results against the requirements input (CMMI Product Team 2010).

4.3.1 Standards and Interfaces for Requirements Traceability and Change Management

In order to understand how the requirements and the PSS design are connected and transformed into each other, lower-level requirements have to be linked with the higher-level requirements they originate from, so that each requirement can be traced to its information source (Wynn et al. 2011). Requirements traceability enables both assessing the effect of changes of stakeholder requirements to PSS development as well as to check if every PSS component is linked to a specific stakeholder requirement. Thus, all requirements can be linked to lower layers and qualified, which is important to assure that they are met by the PSS. The progress of PSS development can be monitored and the impact of changed requirement tested in this way.

For complex systems like PSS, it is hard to avoid a change of requirements. Changing environment or stakeholders may induce changes all along the life-cycle and impact the development process (Lim, Finkelstein 2011). To ensure that such modified requirement are fed back into PSS development, a change management process has to be established (Huang et al. 2011). The RE framework will ensure requirements traceability and change management in the following way.

4.3.2 PSS Testing for Requirements Qualification

Finally, it has to be evaluated if the PSS complies with the requirements specification or not (Project Management Institute 2013). This confirmation that the PSS fully satisfies the documented requirements is conducted in requirements qualification. Deviations from requirements can be detected e.g. by requirement reviews, design inspections, component tests and trials, which has to start early in order to avoid late design changes and rebuilds (Hull et al. 2005). The RE framework will qualify requirements in a similar way to the V-model depicted in Figure 4, first testing the individual components functions, then the integrated system and finally the fulfilment of stakeholder requirements.

5 Summary and Conclusions

The state-of-the-art of requirements engineering and its importance for the realization of integrated products and services – in particular PSS – has been investigated in this deliverable. The nature of PSS – being multidisciplinary and developed in a collaborative environment – poses certain challenges to the development and consequently the requirements engineering as it is the key factor for the successful realization. The characteristics of PSS and the RE phases analyzed in Chapter 2 build the basis for the analysis. Within the literature review, existing requirements engineering methodologies are analyzed concerning the characteristics. To this end, deficits and gaps in the existing methodologies can be evaluated to identify further research needs.

The analysis and classification show that the existing RE methodologies of the 'classic' disciplines – product engineering, software engineering and service engineering – are not fully applicable to PSS due to the lack of a holistic view of the lifecycle phases and the system such a complex solution represents. The methodologies of an integrated product- and service-discipline are still at an initial stage. Procedures and methods are vague and not sufficiently detailed. As an important part of the PSS, it is thus necessary to include service engineering to a higher degree. The discipline of Service Engineering is still object to significant research and in its infancy. Future research will have to dive deeper into the topic of comprehensive integration of the domain-specific methods and techniques to provide a tool-set which is required to identify and maintain requirements of the respective domains. Further, clear definitions of interfaces between the domains need to be devised. As collaboration and stakeholder integration are important factors for the successful realization of the EP, these areas could also be possible research directions.

The collaborative RE Framework for PSS drafts a first outline of an integrated RE approach to be applied in the PSS area. Based on the literature review, it offers suitable methods for requirements development and requirements management, which will be further explored during the course of the project. A first application of the framework is made in T2.2, where the end user requirements towards the PSYMBIOSYS developments are determined. Using the practical feedback of the industrial project partners, the framework can be detailed and tailored to the requirements of the industry. Its evolution and implementation will be reported in the upcoming WP2 deliverables.

6 References

Abramovici, Michael; Aidi, Youssef; Jin, Feng; Göbel, Jens-Christian (2012): Lifecycle Management von Hybriden Leistungsbündeln. In Horst Meier, Eckart Uhlmann (Eds.): Integrierte Industrielle Sach- und Dienstleistungen. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 265–284.

Ahrens, Gritt (2000): Das Erfassen und Handhaben von Produktanforderungen. Methodische Voraussetzungen und Anwendung in der Praxis.

Alter, Steven (2012): Metamodel for Service Analysis and Design Based on an Operational View of Service and Service Systems. In *Service Science* 4 (3), pp. 218–235. DOI: 10.1287/serv.1120.0020.

Aurich, J. C.; Schweitzer, E.; Fuchs, C. (2007): Life Cycle Management of Industrial Product-Service Systems. In Shozo Takata, Yasushi Umeda (Eds.): Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses. London: Springer London, pp. 171–176.

Aurum, Aybüke; Wohlin, Claes (2005): Engineering and managing software requirements. Berlin, London: Springer.

Azadegan, Aida; Papamichail, K. Nadia; Sampaio, Pedro (2013): Applying collaborative process design to user requirements elicitation: A case study. In *Computers in Industry* 64 (7), pp. 798–812. DOI: 10.1016/j.compind.2013.05.001.

Baines, T. S.; Lightfoot, H. W.; Evans, S.; Neely, A.; Greenough, R.; Peppard, J. et al. (2007): State-of-the-art in product-service systems. In *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 221 (10), pp. 1543–1552. DOI: 10.1243/09544054JEM858.

Baxter, David; Gao, James; Case, Keith; Harding, Jenny; Young, Bob; Cochrane, Sean; Dani, Shilpa (2008): A framework to integrate design knowledge reuse and requirements management in engineering design. In *Robotics and Computer-Integrated Manufacturing* 24 (4), pp. 585–593. DOI: 10.1016/j.rcim.2007.07.010.

Berkovich, Marina; Leimeister, Jan Marco; Krcmar, Helmut (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, 2009, pp. 885–894.

Berkovich, Marina; Leimeister, Jan Marco; Krcmar, Helmut (2011): Requirements Engineering für Product Service Systems. In *Wirtschaftsinf* 53 (6), pp. 357–370. DOI: 10.1007/s11576-011-0301-3.

Blanchard, Benjamin S. (2004): System engineering management. 3rd ed. Hoboken, NJ: John Wiley.

Boehm, B.; Basili, V. R. (2001): Top 10 list [software development]. In *Computer* 34 (1), pp. 135–137. DOI: 10.1109/2.962984.

Böhmann, Tilo; Langer, Philipp; Schermann, Michael (2008): Systematische Überführung von kundenspezifischen IT-Lösungen in integrierte Produkt-Dienstleistungsbausteine mit der SCORE-Methode. In *Wirtschaftsinf* 50 (3), pp. 196–207. DOI: 10.1365/s11576-008-0047-8.

Botta, Christian (2007): Rahmenkonzept zur Entwicklung von Product-Service Systems. Product-Service Systems Engeneering. 1. Aufl. Lohmar [u.a.]: Eul-Verl.

Bullinger, H.-J; Scheer, August-Wilhelm; Schneider, Kristof (2006): Service Engineering. Entwicklung und Gestaltung innovativer Dienstleistungen. 2., vollständig überarbeitete und erw. Aufl. Berlin: Springer.

Bullinger, Hans-Jörg; Scheer, August-Wilhelm (Eds.) (2006): Service Engineering. Berlin/Heidelberg: Springer-Verlag.

Bullinger, Hans-Jörg; Schreiner, Peter (2006): Service Engineering: Ein Rahmenkonzept für die systematische Entwicklung von Dienstleistungen. In Hans-Jörg Bullinger, August-Wilhelm Scheer (Eds.): Service Engineering. Berlin/Heidelberg: Springer-Verlag, pp. 53–84.

Chandler, Daniel; Munday, Rod (2011): A Dictionary of Media and Communication: Oxford University Press.

CMMI Product Team (2010): CMMI for Development (CMMI-DEV). Technical Report CMU/SEI-2006-TR-008.

Ebert, Christof (2012): Systematisches Requirements Engineering. Anforderungen ermitteln, spezifizieren, analysieren und verwalten. 4., überarb. Aufl. Heidelberg: Dpunkt-Verl.

Edvardsson, Bo; Olsson, Jan (1996): Key Concepts for New Service Development. In *The Service Industries Journal* 16 (2), pp. 140–164. DOI: 10.1080/02642069600000019.

Ehrlenspiel, Klaus (2007): Integrierte Produktentwicklung. Denkabläufe, Methodeneinsatz, Zusammenarbeit. 3., aktualis. Aufl. München [u.a.]: Hanser.

Elgh, Fredrik (2007): Modelling and Management of Manufacturing Requirements in Design Automation Systems. In Geilson Loureiro, Richard Curran (Eds.): Complex Systems Concurrent Engineering. London: Springer London, pp. 321–328.

Frietzsche, Ursula; Maleri, Rudolf (2006): Dienstleistungsproduktion. In Hans-Jörg Bullinger, August-Wilhelm Scheer (Eds.): Service Engineering. Berlin/Heidelberg: Springer-Verlag, pp. 195–225.

Fuxin, Freddy (2005): Configurable product views based on geometry user requirements. In *Computer-Aided Design* 37 (9), pp. 957–966. DOI: 10.1016/j.cad.2004.09.022.

Gausepohl, Kimberly Ann (2008): Investigation of storytelling as a requirements elicitation method for medical devices.

Goedkoop, M. J. (1999): Product service systems, ecological and economic basics. [The Hague], Zoetermeer: [Ministry of Housing, Spatial Planning and the Environment, Communications Directorate]; Distributiecentrum VROM [distr.] (Publikatiereeks produktenbeleid, nr. 1999/36).

Grönroos, Christian (2007): Service management and marketing. Customer management in service competition. 3rd ed. Chichester, West Sussex, England, Hoboken, NJ: J. Wiley & Sons.

Haller, Sabine (2015): Dienstleistungsmanagement. Grundlagen - Konzepte - Instrumente. 6., vollst. überarb. u. erw. Aufl. Wiesbaden: Gabler (SpringerLink : Bücher).

Hass, Kathleen B.; Wessels, Don J.; Brennan, Kevin (2008): Getting it right. Business requirement analysis tools and techniques. Vienna, VA: Management Concepts (Business analysis essential library).

Hauksdóttir, Dagný; Mortensen, Niels Henrik; Nielsen, Poul Erik (2013a): Identification of a reusable requirements structure for embedded products in a dynamic market environment. In *Computers in Industry* 64 (4), pp. 351–362. DOI: 10.1016/j.compind.2012.10.008.

Hauksdóttir, Dagný; Mortensen, Niels Henrik; Nielsen, Poul Erik (2013b): Identification of a reusable requirements structure for embedded products in a dynamic market environment. In *Computers in Industry* 64 (4), pp. 351–362. DOI: 10.1016/j.compind.2012.10.008.

Hippel, Eric von (2005): Democratizing innovation. Cambridge, Mass.: MIT Press.

Howe, Denis (1985): Free On-line Dictionary of Computing. Available online at http://foldoc.org/information%20technology, updated on 9/25/2015, checked on 9/25/2015.

Huang, Hong-Zhong; Li, Yanfeng; Liu, Wenhai; Liu, Yu; Wang, Zhonglai (2011): Evaluation and decision of products conceptual design schemes based on customer requirements. In *J Mech Sci Technol* 25 (9), pp. 2413–2425. DOI: 10.1007/s12206-011-0525-6.

Hull, Elizabeth; Jackson, Ken; Dick, Jeremy (2005): Requirements engineering. 2nd ed. London: Springer.

Jarke, Matthias; Loucopoulos, Pericles; Lyytinen, Kalle; Mylopoulos, John; Robinson, William (2011): The brave new world of design requirements. In *Information Systems* 36 (7), pp. 992–1008. DOI: 10.1016/j.is.2011.04.003.

Kersten, Wolfgang; Kern, Eva-Maria; Zink, Thomas (2006): Collaborative Service Engineering. In Hans-Jörg Bullinger, August-Wilhelm Scheer (Eds.): Service Engineering. Berlin/Heidelberg: Springer-Verlag, pp. 341–357.

Kitchenham, Barbara; Pearl Brereton, O.; Budgen, David; Turner, Mark; Bailey, John; Linkman, Stephen (2009): Systematic literature reviews in software engineering – A systematic literature review. In *Information and Software Technology* 51 (1), pp. 7–15. DOI: 10.1016/j.infsof.2008.09.009.

Kotler, Philip; Bloom, Paul N. (1984): Marketing professional services. Englewood Cliffs, NJ: Prentice-Hall.

Kotonya, Gerald; Sommerville, Ian (1998): Requeriments Engineering Processes and Techniques. New York: Wiley and Sons.

Kraemer, Wolfgang; Zimmermann, Volker (1996): Public Service Engineering — Planung und Realisierung innovativer Verwaltungsprodukte -. In August-Wilhelm Scheer, J. Hagemeyer (Eds.): Rechnungswesen und EDV. 17. Saarbrücker Arbeitstagung 1996. Heidelberg: Physica-Verlag HD, pp. 555–580.

Lanubile, Filippo (2009): Collaboration in Distributed Software Development. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Eds.): Software Engineering, vol. 5413. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 174–193.

Laporti, Viviane; Borges, Marcos R. S.; Braganholo, Vanessa P. (2007): A Collaborative Approach to Requirements Elicitation. In : 11th International Conference on Computer Supported Cooperative Work in Design. Melbourne, Australia, pp. 734–739.

Li, Zude; Rahman, Quazi A.; Ferrari, Remo; Madhavji, Nazim H. (2009): Does Requirements Clustering Lead to Modular Design? In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Eds.): Requirements Engineering: Foundation for Software Quality, vol. 5512. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 233–239.

Lim, Soo Ling; Finkelstein, Anthony (2011): Anticipating Change in Requirements Engineering. In Paris Avgeriou, John Grundy, Jon G. Hall, Patricia Lago, Ivan Mistrík (Eds.): Relating Software Requirements and Architectures. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 17–34.

Lindahl, Mattias; Sundin, Erik; Sakao, Tomohiko; Shimomura, Yoshiki (2007): Integrated Product and Service Engineering versus Design for Environment — A Comparison and Evaluation of Advantages and Disadvantages. In Shozo Takata, Yasushi Umeda (Eds.): Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses. London: Springer London, pp. 137–142.

Lindemann, Christian; Hippler, Kurt-Robert; Koch, Rainer (2011): Requirements Engineering: Ein Ansatz auch für die klassische Produktentwicklung? Fachtagung des GI-Fachbereichs Softwaretechnik, 21.-25.02.2011, Karlsruhe. In Ralf H. Reussner, Alexander Pretschner, Stefan Jähnichen (Eds.): Software Engineering 2011 - Workshopband (inkl. Doktorandensymposium), Fachtagung des GI-Fachbereichs Softwaretechnik, 21.-25.02.2011, Karlsruhe: GI (LNI, 184), pp. 205–216. Available online at http://subs.emis.de/LNI/Proceedings/Proceedings184/article6333.html.

Lindemann, Udo (2009): Methodische Entwicklung technischer Produkte. Berlin, Heidelberg: Springer Berlin Heidelberg.

Lindow, Kai; Müller, Patrick; Stark, Rainer; others (2011): NEW JOB ROLES IN GLOBAL ENGINEERING-FROM EDUCATION TO INDUSTRIAL DEPLOYMENT. In : DS 68-8: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 8: Design Education, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011.

Liu, Xiaoqing; Raorane, Samir; Leu, Ming C. (2007): A Web-based Intelligent Collaborative System for Engineering Design. In W. D. Li, Chris McMahon, S. K. Ong, Andrew Y. C. Nee (Eds.): Collaborative Product Design and Manufacturing Methodologies and Applications. London: Springer London (Springer Series in Advanced Manufacturing), pp. 37–58.

Maussang, Nicolas; Sakao, Tomohiko; Zwolinski, Peggy; Brissaud, Daniel (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, pp. pp. Available online at https://hal.archives-ouvertes.fr/hal-00182705.

Meier, H.; Roy, R.; Seliger, G. (2010): Industrial Product-Service Systems—IPS2. In *CIRP Annals - Manufacturing Technology* 59 (2), pp. 607–627. DOI: 10.1016/j.cirp.2010.05.004.

Meier, Horst; Uhlmann, Eckart; Kortmann, Daniel (2005): Hybride Leistungsbündel-Nutzenorientiertes Produktverständnis durch interferierende Sach-und Dienstleistungen. In *wt Werkstattstechnik online* 95 (7), p. 8.

Mont, O.K (2002): Clarifying the concept of product–service system. In *Journal of Cleaner Production* 10 (3), pp. 237–245. DOI: 10.1016/S0959-6526(01)00039-7.

Morelli, Nicola (2006): Developing new product service systems (PSS): methodologies and operational tools. In *Journal of Cleaner Production* 14 (17), pp. 1495–1501. DOI: 10.1016/j.jclepro.2006.01.023.

Müller, Patrick; Stark, Rainer Rainer; Fraunhofer, I. P.K. (2014): Integrated engineering of products and services: Layer-based development methodology for product-service systems: Fraunhofer Verlag.

Murthy, D. N. P; Rausand, Marvin; Østerås, Trond (2008): Product reliability. Specification and performance. London: Springer (Springer series in reliability engineering).

Nilsson, Patrik; Fagerström, Björn (2006): Managing stakeholder requirements in a product modelling system. In *Computers in Industry* 57 (2), pp. 167–177. DOI: 10.1016/j.compind.2005.06.003.

Nuseibeh, Bashar; Easterbrook, Steve (2000): Requirements engineering. In Anthony Finkelstein (Ed.): ICSE '00 Proceedings of the Conference on The Future of Software Engineering. Limerick, Ireland, pp. 35–46.

Pahl, Gerhard; Beitz, Wolfgang (2007): Konstruktionslehre. Grundlagen erfolgreicher Produktentwicklung ; Methoden und Anwendung. 7. Aufl. Berlin [u.a.]: Springer (Springer-Lehrbuch).

Penzenstadler, Birgit; Eckhardt, Jonas (2012): A Requirements Engineering content model for Cyber-Physical Systems. In : 2012 IEEE Second Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems (RES4). Chicago, IL, USA, pp. 20–29.

Pohl, Klaus (2008): Requirements Engineering. Grundlagen, Prinzipien, Techniken. 2., korrigierte Aufl. Heidelberg: Dpunkt-Verl.

Project Management Institute (2013): A guide to the project management body of knowledge (PMBOK guide). Fifth edition.

PSYMBIOSYS Consortium (2014): "Product-Service sYMBIOtic SYStems" Grant agreement no: 636804. Annex 1 - Description of Action. FoF-05-2014: Innovative Product-Service design using manufacturing intelligence.

Ramaswamy, Rohit (1996): Design and management of service processes. Reading, Mass.: Addison-Wesley Pub. Co. (Engineering process improvement series).

Rausch, Andreas; Broy, Manfred (2006): Das V-Modell XT. Grundlagen, Erfahrungen und Werkzeuge. 1. Aufl. Heidelberg, Neckar: Dpunkt.

Ribeiro, Claudia; Farinha, Carla; Pereira, João; Mira da Silva, Miguel (2014): Gamifying requirement elicitation: Practical implications and outcomes in improving stakeholders collaboration. In *Entertainment Computing* 5 (4), pp. 335–345. DOI: 10.1016/j.entcom.2014.04.002.

Rouse, William B.; Sage, Andrew P. (2009): Handbook of systems engineering and management. 2nd ed. Hoboken, N.J.: John Wiley & Sons (Wiley series in systems engineering and management).

Royce, Winston W. (1970): Managing the development of large software systems. In : proceedings of IEEE WESCON, vol. 26. Los Angeles, pp. 328–388.

Rupp, Chris (2009): Requirements-Engineering und -Management. Professionelle, iterative Anforderungsanalyse für die Praxis. 5., aktualisierte und erw. Aufl. München, Wien: Hanser.

Schneider, Kristof; Daun, Christine; Behrens, Hermann; Wagner, Daniel (2006): Vorgehensmodelle und Standards zur systematischen Entwicklung von Dienstleistungen. In Hans-Jörg Bullinger, August-Wilhelm Scheer (Eds.): Service Engineering. Berlin/Heidelberg: Springer-Verlag, pp. 113–138.

Schweitzer, Eric; Fiekers, Christoph; Möhrer, Jürgen (2010): Realisierung investiver Produkt-Service Systeme. In Jan C. Aurich, Michael H. Clement (Eds.): Produkt-Service Systeme. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 95–116.

Shen, Hui; Wall, Brian; Zaremba, Michal; Chen, Yuliu; Browne, Jim (2004): Integration of business modelling methods for enterprise information system analysis and user requirements gathering. In *Computers in Industry* 54 (3), pp. 307–323. DOI: 10.1016/j.compind.2003.07.009.

Spath, Dieter; Demuß, Lutz (2006): Entwicklung hybrider Produkte — Gestaltung materieller und immaterieller Leistungsbündel. In Hans-Jörg Bullinger, August-Wilhelm Scheer (Eds.): Service Engineering. Berlin/Heidelberg: Springer-Verlag, pp. 463–502.

Specht, Günter; Beckmann, Christoph; Amelingmeyer, Jenny (2002): F- & -E-Management. Kompetenz im Innovationsmanagement. 2., überarb. und erw. Aufl. Stuttgart: Schäffer-Poeschel (UTB für Wissenschaft).

Thoben, K.-D.; Eschenbächer, J.; Jagdev, H. (2001): Extended Products: Evolving Traditional Product Concepts. In K.-D. Thoben, F. Weber, K. S. Pawar (Eds.): Engineering the knowledge economy through co-operation. ICE 2001, the 7th International Conference on Concurrent

Enterprising ; Bremen, Germany, 27 - 29th June 2001 ; [proceedings]. Nottingham: Centre for Concurrent Enterprising, Univ. of Nottingham, pp. 429–440.

Tingström, Johan; Swanström, Lennart; Karlsson, Reine (2006): Sustainability management in product development projects – the ABB experience. In *Journal of Cleaner Production* 14 (15-16), pp. 1377–1385. DOI: 10.1016/j.jclepro.2005.11.027.

Ulrich, Karl T.; Eppinger, Steven D. (2012): Product design and development. 5th ed., internat. ed. New York, NY: McGraw-Hill.

Unger, Darian; Eppinger, Steven (2011): Improving product development process design: a method for managing information flows, risks, and iterations. In *Journal of Engineering Design* 22 (10), pp. 689–699. DOI: 10.1080/09544828.2010.524886.

van Husen, Christian (2007): Anforderungsanalyse für produktbegleitende Dienstleistungen. Heimsheim: Jost-Jetter (IPA-IAO-Forschung und Praxis, Nr. 458).

van Lamsweerde, A. (2009): Requirements engineering. From system goals to UML models to software specifications. Chichester, England, Hoboken, NJ: John Wiley.

Vasantha, Gokula Vijaykumar Annamalai; Roy, Rajkumar; Lelah, Alan; Brissaud, Daniel (2012): A review of product–service systems design methodologies. In *Journal of Engineering Design* 23 (9), pp. 635–659. DOI: 10.1080/09544828.2011.639712.

Vink, Josina (2015): Storytelling. Conceptualize, Define, Design, Discover, Implement. Edited by Christine Keene. Available online at http://designresearchtechniques.com/casestudies/storytelling/, checked on 12/11/2015.

Volz, Thomas (1997): Management ergänzender Dienstleistungen für Sachgüter. Der schwierige Weg vom Sachgut-Hersteller zum Problemlöser. Göttingen: Vandenhoeck & Ruprecht (Innovative Unternehmensführung, Bd. 27).

Whitehead, Jim (2007): Collaboration in Software Engineering: A Roadmap. In : Future of Software Engineering. Minneapolis, MN, USA, pp. 214–225.

Wiesner, Stefan; Gorldt, Christian; Soeken, Mathias; Thoben, Klaus-Dieter; Drechsler, Rolf (2014): Requirements Engineering for Cyber-Physical Systems. In Bernard Grabot, Bruno Vallespir, Samuel Gomes, Abdelaziz Bouras, Dimitris Kiritsis (Eds.): Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World, vol. 438. Berlin, Heidelberg: Springer Berlin Heidelberg (IFIP Advances in Information and Communication Technology), pp. 281–288.

Wynn, M. T.; Ouyang, C.; ter Hofstede, A.H.M.; Fidge, C. J. (2011): Data and process requirements for product recall coordination. In *Computers in Industry* 62 (7), pp. 776–786. DOI: 10.1016/j.compind.2011.05.003.