Please quote as: Berkovich, M.; Mauro, C.; Leimeister, J. M.; Weyde, F. & Krcmar, H. (2011): Towards Cycle-Oriented Requirements Engineering. In: Wirtschaftsinformatik Konferenz (WI) 2011, Zürich, Schweiz.

Association for Information Systems
AIS Electronic Library (AISeL)

Wirtschaftinformatik Proceedings 2011 | Wirtschaftinformatik

1-1-2011

Towards Cycle-Oriented Requirements Engineering

Marina Berkovich
Technische Universität München, berkovic@in.tum.de

Christian Mauro
Technische Universitat Munchen, mauro@in.tum.de

Jan Marco Leimeister
Universitat Kassel, leimeister@uni-kassel.de

Felix Weyde
Technische Universitat Munchen, felix.weyde@in.tum.de

Helmut Krcmar
Technical Universität Munchen, krcmar@in.tum.de

Recommended Citation
Berkovich, Marina; Mauro, Christian; Leimeister, Jan Marco; Weyde, Felix; and Krcmar, Helmut, "Towards Cycle-Oriented Requirements Engineering" (2011). *Wirtschaftinformatik Proceedings 2011*. Paper 19.
http://aisel.aisnet.org/wi2011/19

This material is brought to you by the Wirtschaftinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftinformatik Proceedings 2011 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Towards Cycle-Oriented Requirements Engineering

Marina Berkovich
Lehrstuhl für Wirtschaftsinformatik
Technische Universität München
Boltzmannstr.3
85748 Garching
berkovic@in.tum.de

Christian Mauro
Lehrstuhl für Wirtschaftsinformatik
Technische Universität München
Boltzmannstr.3
85748 Garching
mauro@in.tum.de

Jan Marco Leimeister
Fachgebiet Wirtschaftsinformatik
Universität Kassel
Nora-Platiel-Str. 4
34127 Kassel
leimeister@uni-kassel.de

Felix Weyde
Lehrstuhl für Wirtschaftsinformatik
Technische Universität München
Boltzmannstr.3
85748 Garching
felix.weyde@in.tum.de

Helmut Krcmar
Lehrstuhl für Wirtschaftsinformatik
Technische Universität München
Boltzmannstr.3
85748 Garching
krcmar@in.tum.de

ABSTRACT

Product Service Systems (PSS) – a combination of software, hardware and service elements – are an emerging trend on the market. Their development encounters different difficulties, amongst them the existence of dynamic cyclic interdependencies. These cycles make the challenging requirements engineering (RE) for PSS even more complicated. Up to now there is no integrated RE approach for PSS. An important step in developing such an approach is to understand the effects of cycles on RE. In this paper the cycles and their causes are analyzed by an industrial case study, a literature review, and by expert interviews. The contribution of this paper is twofold. First, a detailed explanation of cycles' causes emerging in the development and RE of PSS is given and the effects of the resulting cycles on RE are elaborated. Second, a list of requirements to RE for PSS is derived, which describes how the cycles can be managed adequately.

Keywords

requirements engineering, cyclic interdependencies, cycle, complex solutions, product service systems.

1. INTRODUCTION

Differentiation from competitors is one of the most important aspects for the success of a company. Customers want a solution that solves their problems and provides advantages for them [2]. Thus, differentiation by physical technological products (tangible products) or services alone is not sufficient to assure the fulfillment of customers' and market's needs, nor is it adequate for long-term success [13, 37]. Therefore, companies offer "a combination of goods and services that are integrated and customized to meet the idiosyncratic requirements of a customer" [39]. That means companies offer complex solutions, referred to as PSS – product service systems – or hybrid products, consisting of integrated bundles of products and services [23, 38]. The product can be either hardware or software, or a combination of both. Especially software plays a major role in the realization of PSS, because it is essential for innovative multi-functional systems [25]. Accordingly, software has often the task to realize the integration of hardware and services. Integration of different domains as product, software and service engineering in the development process of PSS, different lifecycles of PSS' components, a high degree of technological integration, and a high level of customer integration characterize product service systems and influence their development, making it rather complex [10].

In the development of PSS requirements engineering *(RE)* plays a vital role. RE is one of the most important and also critical activities within a development process and is defined as "gathering, documenting and managing requirements" [3]. A successful RE is a precondition for the success of the development process [11]. Products need to be adapted to the customer requirements, while they are becoming more complex and innovative [28]. Changes of requirements in late phases of the development are up to 100 times more costly than during the initial development steps, during the RE phases [9]. Because of the challenges of PSS described above the RE for PSS is especially intricate and complex [5, 7]. Thus, the overall research question of our research is *how RE for PSS should look like*. Since the research alongside this research question is very extensive, this paper will focus on one special topic. The topic is

10th International Conference on Wirtschaftsinformatik,
16th - 18th February 2011, Zurich, Switzerland

the influence of cyclic interdependencies on the requirements engineering of PSS.

An important aspect in the development of products and services is cyclic interdependencies (further called *cycles*). They lead to iterations, bad coordination between the areas involved in the development process, permanent adaption and changes of PSS' components and of the activities in the development process [20, 21]. Thus, the cycles cause increasing development time and costs. The costs are increasing because the additional iterations in the development cause additional effort and resource spending. For example, it is even possible that the product, which has been already built, has to be modified several times if the customers' requirements have changed. These additional iterations cause time delays as well, which may have a negative impact on the customer satisfaction. Furthermore, supplemental consultation with the customer may be necessary and can cause additional efforts on customer-side.

Due to the complexity and interdisciplinarity of PSS, the cycles are especially challenging for PSS. One of the challenges is different lifecycles of the PSS' components. For example, the hardware may be replaced by a new one within a timeframe in which the provided services keep the same. Notwithstanding, the modification of the hardware can lead to changes of the services which have to be considered. Thus, such differences in lifecycles cause the need for coordination and iterative adjustments of the PSS between different domains.

Many cycles arising during the development of PSS have their roots in requirements engineering. Thus, the first step is to understand the cycles during the RE of PSS profoundly, which is the focus of this paper. For this purpose, the cycles' causes are identified and then examined. For each cycle's cause it is elaborated what cycle it is inducing in the RE. The cycles are inevitable in the RE, but they should be minimized and controlled as far as possible. Based on the insights into cycles' causes and their effects on the RE, we derived a list of recommendations how the cycles in the RE for PSS can be managed adequately.

The research question for this paper is *how RE for PSS should look like in order to be able to deal with cycles.* The contribution of this paper is twofold: (1) a list of cycles and their causes emerging in the RE of PSS (2) a list of requirements for the RE for PSS in order to make it cycle-oriented.

To analyze the cycles, an industrial case study is presented. In this case study, the development process of a PSS in industry is analyzed. Furthermore, the analysis relies on an extensive literature review with the goal to find cycles that have already been described in research literature. We also conducted qualitative expert-interviews to capture current practices in RE in companies producing PSS, with special attention on emerging cycles and their management.

2. RESEARCH METHODOLOGY

Our research was conducted in two phases. The first phase can be seen as a pre-study for the second phase. Hence, the first phase is handled in brief in this paper, and more room is given to the second phase. In figure 1 an overview of the phases and the single steps within the phases is given.

2.1 Phase 1

In phase 1 the development process of PSS must be understood and the role of RE must be clarified. In **step 1a**, a literature review about PSS was conducted in order to understand their characteristics and their development process. A special focus was laid on the RE of PSS and its integration into the development process. The selection of the literature relied on high-quality journals and conferences of information systems ranked as A-Journals and A-Conferences by the WI-list [1]. All in all 88 publications were included in the review.

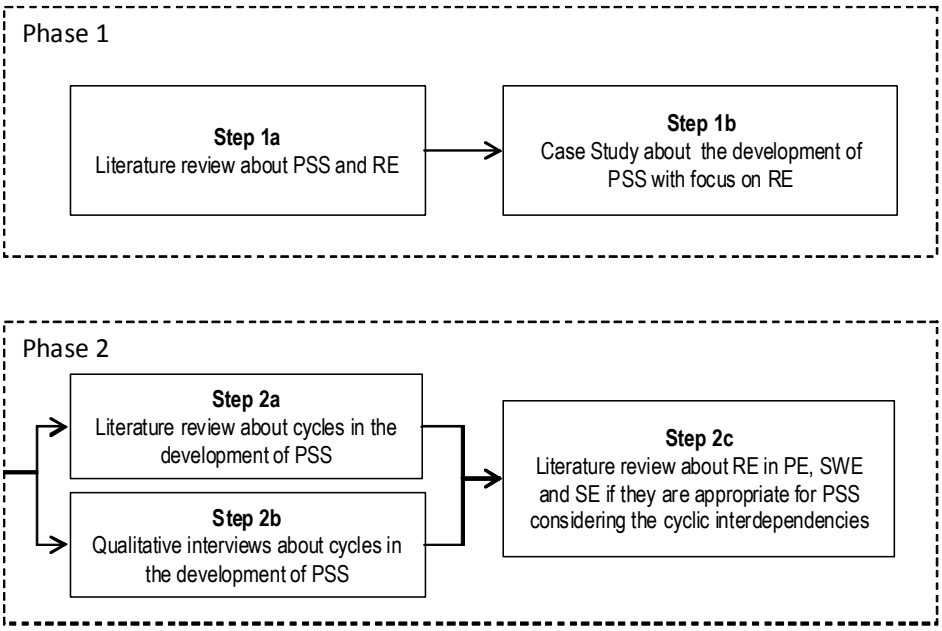

Figure 1: Research Design

Then, in **step 1b**, a case study about the development of PSS was conducted in order to understand their development in practice. The design of this phase is aligned according to Yin [42]. In this case study a large company of the chemical industry producing PSS was analyzed between July and November 2006 by interviewing employees of three departments. The result of phase 1 is a list of characteristics of PSS and of their development process. The insight that cyclical interdependencies are a challenge in the development and RE was made here.

2.2 Phase 2

Phase 2 relies on literature reviews and a series of expert interviews. They had a broader scope and were analyzed in multiple steps. The results of the first analysis did not take the cycles into account and have already been published. These precedent publications are summarized here in short:

- Within an empirical exploration of requirements engineering for hybrid products 15 expert interviews in industry were conducted and analyzed regarding the handling of RE for PSS in practice [7].
- A literature review had the goal to examine the current practices of RE in the three domains involved in the development of PSS. It summarized these practices and evaluated them based on defined criteria [5].
- A similar literature review as described in [5] was also conducted in the product development and published in Berkovich et al. [8]. The criteria for the evaluation of the practices were further elaborated and the techniques are described in more detail.

The second phase has the goal to identify the cycles' causes, to further analyze the role of cycles in the development process of the RE and to analyze especially the role of the RE in handling the cycles.

In **step 2a** a further literature review on cycles arising during the development of PSS (including also the RE) was conducted. The goal of the literature review was to identify cyclic interdependencies and their causes in the development, to understand how they were handled and which implications they had on the development and especially on RE. This literature review included (in addition to the literature on PSS) the top-10 textbooks on product, service and software engineering and design according to the sales ranking of amazon.com (accessed on 07.02.09), as well as publications according to the WI-list [1].

The results of **step 2b** present cycles' causes based on the qualitative expert interviews. These interviews were also used to categorize the cycles using expert opinion. A series of semi-standardized expert interviews was conducted, with the goal to capture current practices in RE in companies producing PSS, with a special attention on the handling of cycles during the development. As described earlier, an analysis of a part of these interviews was already published in Berkovich et al. [7]. The study design is summarized here in brief. In the course of the study 15 experts in companies involved in the development of PSS were interviewed. Some companies were selected because there were contacts to them; others were selected according to the list of Luenendonk [26]. The interviews were carried out between June and October 2008 and were guided according to the process steps of RE: requirements elicitation, analyses and negotiation, documentation and validation, change management and traceability [30, 41].

The result of **steps 2a and 2b** is a list of cycles' causes arising during the development process of PSS and having an impact on RE. During the interviews the cycles of the literature review were presented to the experts. Together with the newly found cycles the experts categorized the cycles according to their causes. The criterion for choosing the categories was that all cycles of one category should have the same impact on RE. The categories developed in each interview, were consolidated by the researchers.

In the **step 2c** we analyzed the existing approaches for RE in the product engineering, software engineering and service engineering about their handling of cycles with the goal to understand if they were appropriate for PSS and to state the requirements on a cycle-oriented RE for PSS. For our literature review we included the domains that are involved in the development of PSS.

As described earlier, the literature review presented in this paper relies on the same data as the reviews published in Berkovich et al. [5] and Berkovich et al. [8]. The research design is summarized here in short. The selection of textbooks relied partially on sales ranks on amazon.com; partly specialized books on RE known to us were also included. In product and service engineering RE is not seen as a special discipline. Thus, for these domains general textbooks on development methods were chosen. Additionally, conference papers and journal publications were selected according to the WI list [1] and a list of high-quality journals available at our institute. The result of this step is an analysis whether the cycles identified before are handled by the RE approaches adequately, if the methods for the handling of cycles used by these approaches are appropriate for PSS and which requirements must be fulfilled by a new RE approach for PSS to handle the cycles.

3. STEP 1A – LITERATURE REVIEW ABOUT PSS AND THE REQUIREMENTS ENGINEERING

3.1 Characteristics of PSS

In contrast to classical PSS consisting of a physical product and a service part, we also regard a combination of software and service or of hardware, software and service as a PSS [4, 10, 24]. The increasing importance of PSS has been shown by several studies. For example Sturm and Bading [35] asked 1641 medium-sized companies about the importance of integrated bundles of products and services and concluded that 59.8% assessed them as very important for the success on the market. A main driver in the development towards PSS is the awareness that a customer does not have an interest in a product or service per se, but he wants his problems to be solved [23].

PSS consist of both material and immaterial components developed by different domains, leading to interdisciplinary work and to the need for technological integration [7, 23]. Interdisciplinary work means that different domains have to

cooperate in the development process in order to achieve a comprehensive understanding of the problem to be solved. The technological integration means that the different parts of PSS have to be integrated technically and organizationally, both between each other and into the value-creation process of the customer [10]. PSS are regarded highly individualized [29], and this individualization is especially important in fulfilling customer wishes.

3.2 Requirements Engineering in the context of PSS

RE is generally understood as a cooperative, incremental and iterative process with the goal of identifying all requirements that are relevant for the system under development, creating a common understanding of the problem to be solved between all stakeholders, and then documenting these requirements in a form that can be used in later development steps [31].

Both in practice and in literature the RE approaches for PSS manage the requirements to a PSS as a whole and in integrated way just insufficiently. Also the domains involved in the development of PSS meet a challenge to gain a comprehensive understanding of the solution to be developed [5, 7]. However, it has been realized that an integrated RE approach would be especially important for PSS, because of their complexity and interdisciplinarity [33]. Berkovich et al. [6] identified the most important activities of RE for PSS (see Figure 2) based on the role of RE in the lifecycle of PSS and on the definition of RE in product engineering, software engineering and service engineering (e.g. [3, 12, 17, 40]): requirements elicitation, analysis and documentation, negotiation, validation, change management and tracing.

The requirements elicitation has the task of gathering all goals, wishes and requirements from the stakeholders. During the requirements analysis the initially elicited requirements are refined and concretized. In this step the initial requirements are "translated" into the language of the developers so that they can be used by the single domains. The requirements analysis includes also the search for conflicting requirements both between initial and concretized requirements and between concretized requirements belonging to the different domains. The purpose of requirements negotiation is to resolve conflicts regarding requirements between stakeholders. The requirements documentation has the task of documenting all requirements according to documentation standards [41]. The requirements

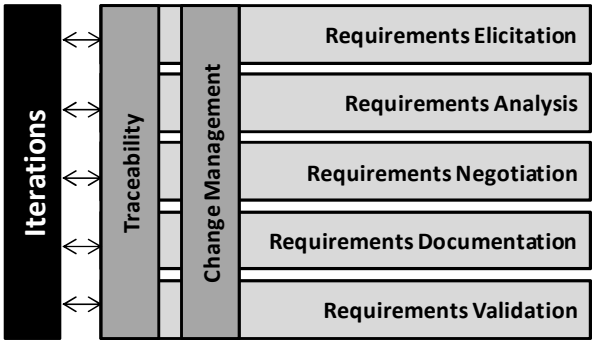

Figure 2: RE Activities and Iterations for PSS according to Berkovich et al. [6]

validation has the task of assuring that the requirements actually represent the vision of the stakeholders. The last two phases – change management and tracing – are done during the development process. The traceability refers to the ability to follow the life-cycle of a requirement and to assess the impact of changes of requirements.

4. STEP 1B – CYCLES FOUND IN AN INDUSTRIAL CASE STUDY

In order to understand the development of PSS in practice we analyzed a company of the chemical industry. Further, we describe the parts of the case study that stay in relation to the cycles. The PSS of this company consisted of mainly two parts, according to which the company was organized: (1) Engineering division: construction of gas processing plants (2) Gas division: supplying gas to customers.

In greater detail, the first division develops both customer-individual facilities and standardized facilities based on serial production. The reason for offering standardized facilities is the cost-advantage. These facilities are cheaper, but may not fully meet customer requirements. The second division offers customers a supply of gas. Depending on the circumstances and requirements of the customer, a solution is provided that may consist of pipelines, transport of gas by trucks, or construction of plants at the customer's side. This second division orders the required engineering work from the first one. Additionally, services for already existing facilities are offered, like maintenance, installation, training and initial operation. The customers of the company simply acquire the service "gas supply" and are unconcerned about how the gas is brought to them. The contractor then constructs the necessary facilities and operates them.

In the development of the gas facilities, a concurrent development of hardware and software takes place. During this development process cycles can be observed. The engineering division is exclusively concerned with chemical processes and their implementation by hardware. All software development for control software is outsourced. The first step when starting a software development is to define the concretized requirements which can be realized by the domains. During the contracting phase a solution is iteratively developed. Due to the cost-pressure, the processes and also the software are adjusted during construction and operation of the facility (see Figure 3).

In this way, experiences can be transferred to the development. However, during the development some development steps have

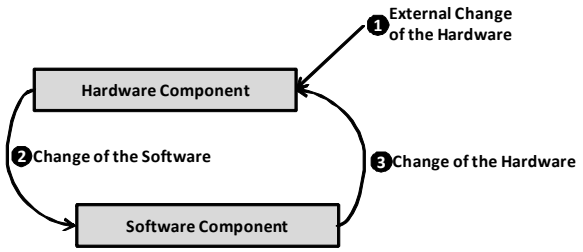

Figure 3: Cycle in the in the development of gas facilities caused by a concurrent development of hardware and software

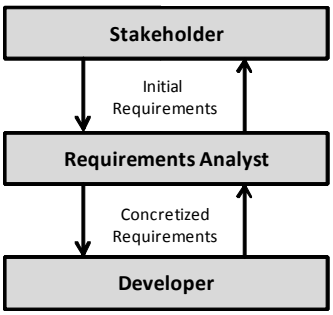

Figure 4: Cycle in the development of gas facilities caused by misunderstandings between the RE and the development

to be repeated because of changing requirements. Hence, the RE plays a major role in handling the cycles. For a most effective handling of cycles, the RE should be able to predict cycles and give the possibility to avoid them.

During the development also another type of cycle was observed. When the development is starting, the initial requirements are captured in a specification document. Then these requirements are concretized and a more precise specification is created. This detailed specification is given to the developers as the basis of their work. During the requirements validation of the concretized requirements as well as during the realization of requirements by the developers, variations and inconsistencies were found. They lead to a rework of the requirements specification, which created multiple cycles between initial stakeholders, requirements analysts, and developers (see Figure 4).

5. STEP 2A AND 2B – CYCLES' CAUSES INFLUENCING REQUIREMENTS ENGINEERING

In this section the results of the expert interviews and literature reviews are described. In our description of the cycles, we describe causes for cycles and group them in categories of similar causes (see Table 1). For each group, it is described which RE activities are iteratively repeated and form a cycle consequently, if the cycle's cause emerges.

Cycles in Literature –The cyclic dependencies within and outside of the development process are a challenge for companies to overcome. According to Langer et al. [21], a cycle is characterized by "a repeated succession of similar occurrences and of results initiated by them, such as sub-processes, artifacts, developments, etc., and the succession of different occurrences within one sequence, e.g., the innovation process". The development of PSS marked as very complex is affected by dynamic cyclic behaviors [21].

Comprehensive List of Cycles Causes – This section categorizes all causes of cycles based on the literature review and interviews into seven major categories. Each category describes the causes for the cycle, gives an example, and explains the implications for the requirements engineering.

Category 1*: Availability and maturity of technologies:* This category contains the causes for cycles that have to do with the availability of new technologies. They were identified in the literature review, for example, Dodgson et al. [14]. If a new technology emerges on the market, it is possible to use it to realize components such as hardware, software or service of the PSS.

An example of such a cycle is the introduction of VoIP, which offers the possibility of restructuring the communication infrastructure (hardware and software) of solutions under development (http://www.msnbc.msn.com/id/6354872/ accessed on 27.04.09). If a component or a part of it is realized by new technology, the concretized requirements can change. These changes can have an impact on the customer requirements of the overall product and on other concretized requirements, and thus have to be analyzed. It must also be checked whether the new technology realizes all requirements that were realized by the old one. Further, new conflicts due to changed concretized requirements may emerge and need to be considered.

If, for example, a new technology is available and should be

Table 1. Categories of Cycles' Causes

Nr.	Category of Cycles' Cause	Causes of Cycles
1	Availability and maturity of technologies	• new software technology / product technology / service technology emerges on the markets
2	Competitive trends	• product is no longer required by the customer • rivalry product is more attractive for the customer • service is no longer required • new market trend is observed
3	Different lifecycles of hardware / software	• hardware / software is out-dated
4	Change of customer-requirements	• change of customer-requirements
5	Financial cycles on the capital market	• product has to be released immediately to the customer • price of the product has to be reduced
6	Development cycles	• hardware / software / service cannot be realized • integration of domains' results failed due to derivations • internal stakeholders change the requirements
7	Legislative cycles	• laws / standards change

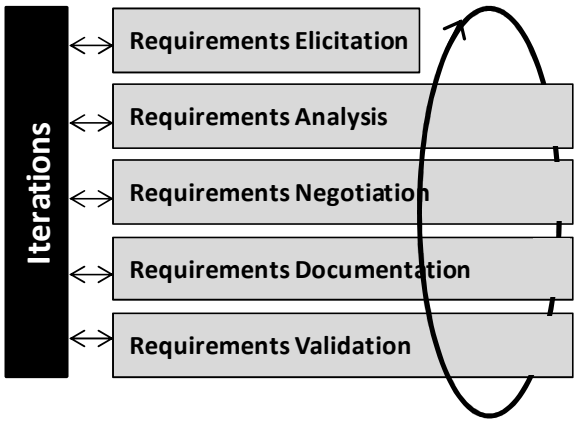

Figure 5: Cycle in the RE caused by "*Availability and maturity of technologies*"

used in the PSS, then the following steps are executed within the RE. Therefore, it is necessary, to check if the requirements emerging from the new technology are conform to the old requirements. First, the requirements emerging from the new technology are collected. Then, they are analyzed for conflicts with the existing requirements. If conflicts are present, they have to be solved by discussion between all stakeholders. Furthermore, the usual activities of RE are executed, such as documentation, and validation. Figure 5 shows this cycle in the RE exemplarily.

Category 2*: Competitive trends:* The causes for this category of cycles were found in the interviews and the literature review, for example, Fuchs and Apfelthaler [15], Spath et al. [34] and Hauschildt and Salomo [16]. These cycles have to do with product management and market trends.

If a PSS or a service is no longer needed by the customer, it means that the customer-requirements have changed.

Given that the customer is interested in the overall product, the new customer-requirements have to be elicited and the RE process has to be repeated, but still reusing the old requirements as much as possible.

If new market trends are observed, the customer needs to be consulted as to whether he wants to integrate them into the PSS. Integrating them would mean adding new customer requirements and refining them to corresponding concretized requirements.

If a rivalry product seems more attractive for the customer since it has better characteristics than the considered PSS, the rivalry product has to be analyzed (for example Benchmarking techniques [40]) and new requirements need to be derived. Particularly quality requirements and attractiveness requirements can have an influence of the competitive advantage [27].

Category 3*: Different lifecycles of hardware and software:* These causes of cycles were found in the literature review, for example, Pahl et al. [27], Lavagno and Passerone [22]. The components of PSS have different life-cycles [5], and thus if one component of a PSS (hardware, software and service) is out-of-date, it has to be replaced. Thereby, new detailed requirements

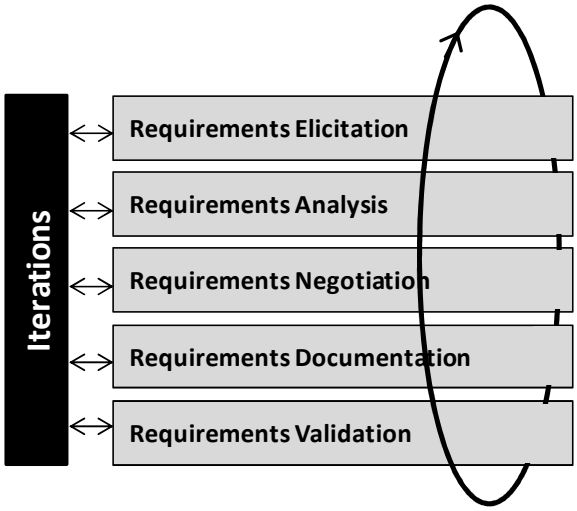

Figure 6: Cycle in the RE caused by "*Change of customer-requirements*"

can emerge within the development, which have an effect on customer requirements.

In general, if it is discovered that a hardware or software component is out-dated, all requirements affected by those components have to be determined. This can be done by using classical traceability techniques. Then, it has to be analyzed whether these requirements change when the old technology is replaced, and the potential effects on other requirements have to be analyzed. All this information is collected and given to the developers in order to give them the possibility to decide which next steps have to be taken.

An example for such cycles is ATMs, where the card-reading hardware had to be replaced due to new standards for credit cards (http://www.stern.de/wirtschaft/geld/87195.html accessed on 20.04.09). Also, the software had to be modified, which was not easily possible because the operating system was no longer supported by the supplier.

Category 4*: Change of customer-requirements:* The customer-requirements can change during development and use of a product [19, 31]. In such cases it is important to clarify the impacts on concretized requirements, on the already developed system, and on the environment. The change of one customer-requirement can also have an impact on other customer-requirements.

This cycle can occur during development time as well as during the use of the PSS. If the customer requirements change, the elicitation of requirements has to be done, in order to elicit all changes requirements. Then, they are analyzed and checked for contradictions to existing requirements. In the requirements negotiation, the contradictions have to be resolved. Then, the new requirements are documented and validated. As we can see, all phases of the requirements engineering are repeated in this cycle. It is shown exemplarily in Figure 6.

Category 5*: Financial cycles on the capital market:* These cycles were identified in the literature review, for example, Fuchs and Apfelthaler [15]. The cycles in this category can have the consequence that the product must be immediately released

to the customer or that the price of the product has to be reduced. In both cases it is necessary to decide which functionality of the product can be abandoned to achieve the necessary savings or the necessary delivery-time, for example the imitations of the iPod. After the release of the iPod, several competitors forced their products under development to move to an immediate market release.

In the case that the price of a product has to be reduced, the RE has to reprioritize all requirements. In order to do so, the requirements analysis has to be redone, because when prioritizing the requirements, the interdependencies between them have to be regarded. The reprioritization of the requirements and the omitting of low prioritized ones necessitate the repetition of the requirements negotiation and validation.

***Category 6**: Development cycles:* These cycles were identified by the interviews. These cycles occur when it is impossible to realize a component (hardware, software, service) of the PSS.

If a component cannot be realized, it has to be decided whether the affected customer-requirement can be changed or whether the functionality can be realized by another component.

If the integration of the partial solutions experiences major problems, the implications for the customer-requirements have to be determined, and the requirements possibly have to be changed.

The change of requirements by internal stakeholders can also cause cycles. It has to be analyzed whether the customer-requirements are affected by these changes and whether coordination with the customer is necessary.

***Category 7**: Legislative cycles:* These cycles emerge when a law or standard that is relevant for the PSS changes. These changes are typically known in advance and it can be planned for them. If such a change emerges, first all new requirements have to be elicited. Then, the new requirements are incorporated into the existing specification. That makes it necessary to newly execute the requirements analysis; in order to check which requirements are affected and to find and resolve newly originated conflicts. Then, the requirements negotiation and validation have to be repeated. As a last step, the effects of the changes of the requirements on the already build parts of the PSS have to be analyzed.

An example of such a cycle is the change of the laws regarding the system for managing the unemployment payments in Germany. After a major amendment of the law, the system had to be changed by 2005-01-01 (http://www.mid.de/fileadmin/documents/pdf/Anwenderberichte/Sonderdruck_eGov_0307_MID.pdf accessed on 18.04.09).

6. STEP 2C – CYCLE-ORIENTED REQUIREMENTS ENGINEERING

The cycles described in the previous section have an impact on the RE of PSS. In order to effectively handle these cycles, the RE has to support certain activities and exhibit certain characteristics. The objective of this section is to analyze which characteristics and activities are needed and to assess whether existent RE approaches are capable of implementing them.

Results of the Literature Review – RE is widely accepted in product engineering and is integrated in the engineering processes. The handling of requirements is, however, often limited to the first phases of the development [7]. The requirements are mostly documented in lists, and changes are then documented within those lists. In software engineering, RE is applied throughout the development process [31]. In contrast to product engineering, the management of requirements during the development is more advanced: changes are documented more carefully and methods for carrying out impact analyses of changes are available. Because service engineering is still a relatively young discipline, there is a general lack of systematic approaches [18]. Some process models of service engineering mention RE but without offering techniques.

Requirements for a Requirements Engineering Approach

As described in the research design, requirements to the RE of PSS – without cycles perspective – can be found in [7]. Based on the same data basis as in [7], we developed additional requirements to the RE of PSS, in order to make it cycle-oriented.

The main requirements to RE for PSS without cycle-orientation are summarized here: The first aspect is that a coordinated RE process for all components of PSS has to be established. The requirements for the different domains cannot be elicited, analyzed, etc., separately. Second, because of involving different domains in the development process, the interdisciplinary work needs to be improved. Third, the customer-integration is essential for successful RE and needs to be improved.

The requirements to the RE process that were derived from the needs of handling the cycles are described in detail here. Table 2 gives an overview of the requirements and the cycles that they were derived from. Each requirement is described here in detail:

Requirement 1: Examination of the influence of a cycle on the RE: If cycles arise, it is necessary to provide a method for determining whether the cycle has an influence on the RE. Scenario techniques [32], which are used to model interactions, are suited for this purpose. By modeling the interaction of the cycle and the development process, the influence can be detected.

Requirement 2: Iterative RE: All identified categories of cycles, except category 5, cause the requirements' changes. A proven concept of handling changes of requirements is an iterative approach. If a requirement is changed, the entire RE process or some activities of it can be repeated [34].

Requirement 3: Tracing from customer- to concretized requirements: The cycles of categories 1, 3, 5, 6, and 7 cause at first the change of concretized requirements, while cycles 2, 4 and 7 cause at first the change of customer-requirements and then of concretized requirements. It is necessary to analyze the impact on concretized requirements if customer requirements are changed, and vice versa. A possibility for tracking such interdependencies is the implementation of traceability. In software engineering, Sommerville and Sawyer [32] introduce traceability to enable tracing of requirements. Traceability matrices are a further means for implementing traceability [19]. For PSS, it must be considered that the concretized requirements concern different domains [7]. The concretized requirements of

Table 2. Requirements for RE to be cycle-oriented

Nr.	Requirements for RE	Cause for this requirement	Cycles involved
1	Examination of the influence of a cycle on the RE	Cycles exist in the development process	all
2	Iterative RE	Requirements can change during the development process due to reasons	1, 2, 3, 4, 6, 7
3	Tracing from customer- to concretized requirements and vice versa	Customer requirements can change Concretized requirements can change	1, 3, 5, 6, 7 2, 4, 7
4	Interdependencies between concretized requirements	On changes, interdependent requirements can also change.	1, 2, 3, 4, 6, 7
5	Identifying requirement conflicts	On changes, new conflicts may emerge.	1, 2, 3, 4, 6, 7
6	Reuse of requirements	After changes, already documented requirement can be reused.	1, 2, 3, 4, 6, 7
7	Defining a validity period of requirements	Previously known changes must be regarded	7
8	Prioritization of requirements	Selection of important requirements.	2, 5

each domain are documented using different notations and are not easily understandable by different people.

Requirement 4: Interdependencies between concretized requirements: If a concretized requirement is changed, further changes of concretized requirements can be triggered. It is important to redo the requirements analysis for the affected requirements, which can be determined by means of traceability. In the case of PSS, these traceability and analysis methods have to support the distribution of the concretized requirements over the domains. Because the requirements in these domains are documented differently, special methods have to be developed. For example, there is no formal way of defining requirements to services [7, 18]. Thus, a method has to be developed for establishing the traces between concretized requirements of different domains. If customer requirements are changed, an analog process and corresponding methods have to be provided.

Requirement 5: Identifying requirement conflicts: When concretized requirements are changed, new conflicts between changed and unchanged requirements can emerge, which are very important to analyze and resolve. Requirements 4 and 5 support the identification of conflicts by providing traceability between requirements; however, the main problem is the identification of conflicts. A common method is to formalize requirements by modeling them, e.g., modeling with use cases [19, 36]. Such mechanisms for identifying conflicts have to be adapted for the three involved domains, and the identification of conflicts between the two different domains must be possible. The identification and resolution of conflicts between customer-requirements can be done analog. Conflicts between customer requirements have to be resolved by established methods, e.g., workshops [3].

Requirement 6: Reuse of requirements: When the customer-requirements and the concretized requirements change (see requirement 2) it is necessary to reuse already defined requirements. By reusing requirements, time and costs can be saved [32]. The reuse can be supported by documenting interdependencies between requirements and by traceability.

Requirement 7: Defining a validity period of requirements: The legislative cycles (category 7) lead to previously known changes in the system on a previously known date. Hence, it is possible to take into consideration these changes during the development by specifying not only how each requirement has to be defined, but also in which period of time it is valid.

Requirement 8: *Prioritization of requirements:* Cycles of categories 2 and 5 lead to a reduction of the offered solution. Therefore, it is important to define priorities for requirements which specify how important a certain requirement is for the customer [28]. Based on these priorities, a decision on the reduction of the functionality can be taken. In the development of PSS it is important to find a prioritization which takes into account the different domains. Thus, the initial prioritization of the customer-requirements has to be transferred to the concretized requirements.

7. CONCLUSIONS AND THE OUTLOOK FOR FUTURE RESEARCH

In this paper a variety of research steps have been presented that were performed in order to understand the role of cycles in the RE of PSS and to derive requirements for the RE. A cycle in the development process covers a sequence of events and the results like sub processes, artifacts, etc. Such cycles complicate the planning of development processes and are problematic in conjunction with high time- and cost-pressure.

A main contribution of this paper is a comprehensive list of causes of cycles derived by literature reviews and expert interviews in industry. This way, the causes for the cycles and the implications of them on the RE were identified. The list of cycles comprises seven categories of different types of cycles and explains the causes of the cycles. It is interesting to see that the cycles' causes appear on different stages of the development and have both company-internal and external causes.

Although cycles appear also in classic requirements engineering processes of the single domains, in the context of PSS, they have even graver implications. Therefore, a special handling of the cycles in the RE of PSS is necessary. It was especially

surprising for us that most of the identified cycles have a direct effect on the requirements for the PSS. Thus, for the effective management of these cycles, the prerequisites need to be laid in RE.

The second part of the study addresses what a requirements engineering needs to provide in order to enable an effective handling of cycles. Eight requirements to a requirements engineering for PSS were derived and are described in detail. Because the cycles emerge during the development, a great challenge is to deal with changing requirements during the development process. This finding is reflected in the requirements to the RE which are largely concerned with impact analyses of changing requirements and traceability issues. The list of requirements to the RE does only comprise specific requirements staying in relation with cycles and PSS. To achieve this, all initially identified issues were compared to known RE approaches in the three domains relevant to PSS. By this means only unsolved issues have been recorded in the list of requirements. Surprisingly, these issues – such as traceability and change management – are already known to requirements engineering research, but not addresses adequately. To achieve a cycle-oriented RE it is therefore necessary to improve these topics of RE.

In our future research the results of this paper will be used to develop a cycle-oriented requirements engineering for PSS. First, the requirements to the RE approach will be incorporated into a process model for RE that describes the activities of RE. The process model has to regard the requirements presented in this paper in order to be cycle-oriented. By the execution of the activities of RE, a specification is created or an existing specification is modified and adapted. Because of the cycles, it is challenging to maintain a specification consistent and avoid that its complexity is growing over time. Therefore, a structure for the specification will be developed that addresses these problems.

8. ACKNOWLEDGEMENT

We thank the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) for funding this project as part of the collaborative research center „Sonderforschungsbereich 768 – Managing cycles in innovation processes – Integrated development of product-service-systems based on technical products.

9. REFERENCES

[1] 2008. WI-Orientierungslisten *Wirtschaftsinformatik*.

[2] Abramovici, M. and Schulte, S. 2007. Optimising customer satisfaction by integrating the customer's voice into product development. In *Proceedings of the International Conference on Engineering Design, ICED'07* (Paris, France, 2007).

[3] Aurum, A. and Wohlin, C. 2005. *Engineering and Managing Software Requirements.* Springer, Berlin.

[4] Becker, J. and Krcmar, H. 2008. Integration von Produktion und Dienstleistung – Hybride Wertschöpfung. *Wirtschaftsinformatik 50*, 3 (2008).

[5] Berkovich, M., Esch, S., Leimeister, J.M. and Krcmar, H., Requirements engineering for hybrid products as bundles of hardware, software and service elements – a literature review. In *Proceedings of the 9. Internationale Tagung Wirtschaftsinformatik, WI 2009* (Wien, Österreich, 2009).

[6] Berkovich, M., Leimeister, J.M. and Krcmar, H., Ein Bezugsrahmen für Requirements Engineering hybrider Produkte. In *Proceedings of the Multikonferenz Wirtschaftsinformatik, MKWI 2010* (Göttingen, 2010).

[7] Berkovich, M., Leimeister, J.M. and Krcmar, H., An empirical exploration of requirements engineering for hybrid products. In *Proceedings of the XVIIth European Conference on Information Systems, ECIS 2009* (Verona, Italy, 2009).

[8] Berkovich, M., Leimeister, J.M. and Krcmar, H., Suitability of Product Development Methods for Hybrid Products as Bundles of Classic Products, Software and Serivce Elements. In *Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*, (San Diego, 2009).

[9] Boehm, B. and Basili, V.R. 2001. Top 10 list [software development]. *Computer 34*, 1 (2001), 135 - 137.

[10] Böhmann, T. and Krcmar, H. 2007. Hybride Produkte: Merkmale und Herausforderungen. In *Wertschöpfungsprozesse bei Dienstleistungen: Forum Dienstleistungsmanagement*. Gabler, 240-255.

[11] Browne, G.J. and Rogich, M.B. 2001. An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques. *Journal of Management Information Systems 17*, 4 (2001), 223-249.

[12] Byrd, T.A., Cossick, K.L. and Zmud, R.W. 1992. A synthesis of research on requirements analysis and knowledge acquisition techniques. *MIS Quarterly 16*, 1 (1992), 117-138.

[13] Davies, A. 2004. Moving base into high-value integrated solutions: a value stream approach. *Industrial & Corporate Change 13*, 5 (2004), 727-756.

[14] Dodgson, M., Gann, D. and Salter, A. 2008. *The Management of Technological Innovation: Strategy and Practice: The Strategy and Practice*. Oxford University Press.

[15] Fuchs, M. and Apfelthaler, G. 2008. *Management internationaler Geschäftstätigkeit* Springer, Wien, 2008.

[16] Hauschildt, J. and Salomo, S. 2007. *Innovationsmanagement*. Vahlen.

[17] Hefley, B. and Murphy, W. 2008. *Service Science, Management and Engineering: Education for the 21st Century*. Springer, Berlin.

[18] Husen, C.v. 2007. *Anforderungsanalyse für produktbegleitende Dienstleistungen.* Fakultät Maschinenbau, Universität Stuttgart.

[19] Lamsweerde, A.v. 2009. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley & Sons.

[20] Langer, S., Kreimeyer, M., Müller, P., Lindemann, U. and Blessing, L. 2008. Prozessmodellierung für das Zyklenmanagement der Entwicklung hybrider Leistungsbündel. In *Dienstleistungsmodellierung: Methoden, Werkzeuge und Branchenlösungen*, O. Thomas, and M. Nüttgens, Ed. Physica-Verlag, 71-89.

[21] Langer, S. and Lindemann, U. 2009. Managing cycles in development processes - analysis and classification of external context factors. In *Proceedings of the International conference on engineering design, ICED'09* (Stanford CA, USA, 2009).

[22] Lavagno, L. and Passerone, C. 2005. Design of Embedded Systems. In *Embedded Systems Handbook*, R. Zurawski, Ed. CRC Press Inc.

[23] Leimeister, J.M. and Glauner, C. 2008. Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik. *Wirtschaftsinformatik, 50,* 3 (2008).

[24] Leimeister, J.M., Knebel, U. and Krcmar, H. 2009. Hybrid Value Creation in the Sports Industry - the Case of the Mobile Sports Companion. *International Journal of Information Systems in the Service Sector (IJISSS).*

[25] Liggesmeyer, P. and Rombach, D. 2005. *Software-Engineering eingebetteter Systeme: Grundlagen-Methodik-Anwendungen*. Spektrum Akademischer Verlag.

[26] Lünendonk GmbH 2008. Lünendonk®-Listen 2008.

[27] Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. 2006. *Engineering Design: A Systematic Approach*. Springer, Berlin.

[28] Pohl, K. 2007. *Requirements Engineering. Grundlagen, Prinzipien, Techniken.* Dpunkt Verlag.

[29] Sawhney, M. 2006. Going beyond the Product: Defining, Designing and Devlivering Customer Solutions. In *The Service-dominant Logic of Marketing*, R.F. Lusch, and S.L. Vargo, Ed. M. E. Sharpe, New York, 365-380.

[30] Sommerville, I. 2004. *Software Engineering*. Pearson, Boston.

[31] Sommerville, I. and Kotonya, G. 1998. *Requirements Engineering: Processes and Techniques.* Wiley & Sons.

[32] Sommerville, I. and Sawyer, P. 1997. *Requirements Engineering: A Good Practice Guide*. Wiley & Sons.

[33] Spath, D. and Demuß, L. 2003. Entwicklung hybrider Produkte – Gestaltung materieller und immaterieller Leistungsbündel. In *Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen*, Bullinger, H.-J. and Scheer, A.-W. eds., Springer, Berlin, Heidelberg, New York.

[34] Spath, D., Dill, C. and Scharer, M. 2001. *Vom Markt zum Markt*. Log_x.

[35] Sturm, F. and Bading, A. 2008. Investitionsgüterhersteller als Anbieter industrieller Lösungen – Bestandsaufnahme des Wandels anhand einer Umfrage. *Wirtschaftsinformatik 50,* 3 (2008).

[36] Sutcliffe, A. and Maiden, N. 1998. The domain theory for requirements engineering. *Software Engineering, IEEE Transactions on 24*, 3 (1998), 174-196.

[37] Tan, A.R., McAloone, T.C. and Gall, C., Product/Service-System Development - An explorative Case Study in a manufacturing Company. In *Proceedings of the International Conference on Engineering Design, ICED* (Paris, France, 2007).

[38] Tukker, A. 2004. Eight types of product-service system: eight ways to sustainability? Experiences from suspronet. *Business Strategy and the Environment, 13*. 246-260.

[39] Tuli, K.R., Kohli, A.K. and Bharadwaj, S.G. 2007. Rethinking Customer Solutions: From Product Bundles to Relational Processes. *Journal of Marketing 71,* 3 (2007), 1-17.

[40] Ulrich, K. and Eppinger, S. 2003. *Product Design and Development* Mcgraw-Hill Professional.

[41] Watson, H.J. and Frolick, M.N. 1993. Determining information requirements for an EIS. *MIS Quarterly 17*, 3 (1993), 255-269.

[42] Yin, R.K. 2009. *Case Study Research. Design and Methods*. B&T.