Guideline to elicit requirements on industrial product-service systems

Patrick Müller, Felix Schulz, Rainer Stark
Chair of Industrial Information Technology, TU Berlin, Pascalstraße 8-9, Berlin, 10587, Germany
patrick.mueller@mailbox.tu-berlin.de

Abstract

Industrial solutions integrating products and services are common practice in many business-to-business applications and branches. An explicit integrated planning, development, delivery and use of products and services, which is important for so-called industrial product-service systems (IPS^2), is hardly implemented. Thus, a need and potential for enhancement of IPS^2 design is driving our research. In our contribution we will present a new guideline to elicit and analyze requirements on IPS^2 properties and quality. The guideline is based on a comprehensive review of product development, service engineering and IT literature. It contains a new checklist of clustered criteria to retrieve and describe IPS^2 requirements systematically and it supports different design views. The guideline is compatible with known approaches as for instance guidelines presented by Pahl et al., van Husen, Steinbach or CobiT/ITIL. Within the paper we discuss reviewed references and present the guideline and the entire checklist.

Keywords

IPS^2 design, IPS^2 requirements and quality, guideline, checklist

1 INTRODUCTION

Industrial solutions integrating products and services are common practice in many business-to-business applications and branches. An explicit integrated planning, development, delivery and use of products and services, which is important for so-called industrial product-service systems (IPS^2) [1], is hardly implemented. Thus, a need and potential for an enhancement of IPS^2 development is driving our research.

During our research we found out that for many domains generic guidelines for requirements generation exit, but that for IPS^2 development no such guideline was published until now. Many authors just mention customer needs and requirements without offering a methodical support to handle those on a more detailed level of abstraction, cp. elaboration in [2]. Thus, the aim was to set up a systematic support for the generation of requirements during the planning and early development phase of an IPS^2.

The following subsections introduce major aspects of IPS^2, requirements engineering and generation and our research methodology. In section 2 we provide an overview of the bandwidth and the content of a comprehensive literature review; in section 3 we compile a guideline for IPS^2 requirements generation; in section 4 we report on applications and findings before we close in section 5 with a summary and an outlook.

1.1 Characteristics of (industrial) Product-Service Systems

The value provided by the concept of (industrial) product-service systems is a broad, holistic view on technical systems by taking into account technical artefacts, services, business models, and drivers like sustainability and dematerialization. The premise is to provide added value to satisfy customer needs along the whole lifecycle of a product-service system. [3]

The basic idea is not to sell products and services separately but to sell a defined result, a system's availability or just functionality. Customer needs are not simply reduced to the single need for product ownership. According business models [4-6] define the value for the customer and couple customers and providers for longer time. Maintenance, adoption to changing needs and boundary conditions, reconfiguration or upgrading can be part of an IPS^2, e.g. in form of services included in the business model. The integration of products and services finally can maintain or enhance functionality of a product or a service or implement new functions, which are not available without integration. In the area of high-cost machinery IPS^2 are sold instead of standalone products or services to exploit earlier unused economical and technical potentials or to enhance the value for the customer [5, 7]. Responsibility and risk are shared among the contractors.

Brief definitional summary

[Necessary] Product-Service Systems (PSS) are customer, lifecycle, and foremost sustainability oriented systems, solutions, or offers, integrating products and services.

[Sufficient] Business models framed by contracts align incentives of the customer and the provider, aim at assuring functionality throughout system life-time and aim at implementing added value to satisfy customer needs.

[Remarks] Industrial Product-Service Systems (IPS^2) represent PSS business-to-business applications. Explicit PSS and IPS^2 are characterized by an integrated planning, development, delivery and use of products and services. Implicit PSS and IPS^2 are not explicitly planned, developed, delivered and used in "integrated" processes, but already existing in today's markets and (somehow) integrating products and services.

Figure 1 illustrates a simplified architecture of IPS^2 core elements. Next to core products and services, stakeholders and contracts are important. IPS^2 are type of long-term commitments regulated by a contract. The contract provides tight linkages between stakeholders and defines how risk, responsibilities, and costs, concerning the integrated delivery and operation of product and service shares, are distributed among them.

Simplified, the stakeholders are one provider, multiple suppliers, and one or more customer(s). They are typically organized in a locally distributed network with partly

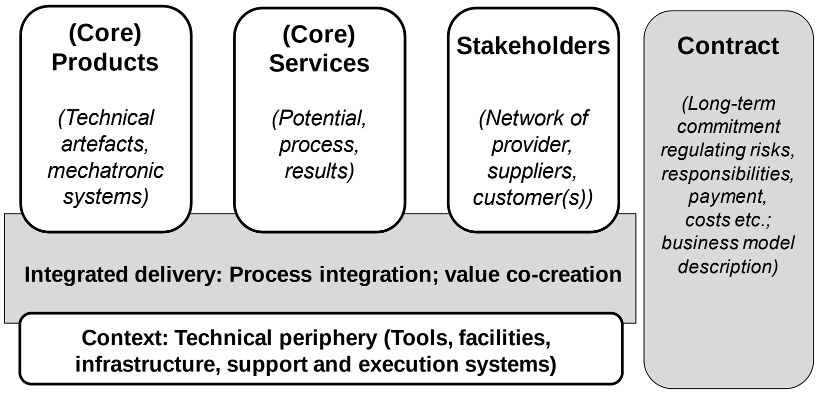

Figure 1: Simplified PSS / IPS^2 architecture.

integrated business processes. An important aim is a value co-creation among the stakeholders during the integrated delivery. Supplemental systems and tools have to be taken into account to enable the delivery of products and services and the exchange of information.

1.2 Requirements Engineering

Definition of a requirement

According to Kruse [8] „a requirement is a defined behaviour, characteristic or property, to be assumed for an object, a person or an activity which has to assure a certain result in a value creation process". (Original text in German: "Eine Anforderung ist ein definiertes Verhalten oder [eine] bestimmte Eigenschaft, anzunehmen von einem Objekt, einer Person oder Aktivität zur Sicherstellung einer Leistung in einem Wertschöpfungsprozess.") This definition has been chosen for our contribution, because it considers objects, actors / stakeholders, activities and values, which all are relevant in the IPS^2 context.

The process of requirements engineering

According to [9] the process of requirements engineering includes the following activities:

- Requirements elicitation (input: user needs, domain information, existing system information, regulations, standards, etc.)
- Requirements analysis and negotiation
- Requirements documentation
- Requirements validation (based on a requirements document which is further used to set up the system specification)

Requirements engineering is a process which is accompanying the planning and development of a system. Starting with an elicitation of customer needs, system level requirements have to be derived and broken down into function and component level requirements. Especially in IPS^2 planning and development a clear distinction of customer needs and requirements is vital in order to exploit the full solution space offered by product-service integration, cp. [2]. Reuse of requirements from former system developments is typical.

The elicitation or generation of requirements, which includes the retrieval, an awareness of relevant issues and the formulization in order to document and model requirements, is not addressed properly in IPS^2 planning and development research so far.

1.3 Research methodology

Perspective, focus and limitations of this paper

The study has been undertaken from an engineering design perspective. The background of the authors is mechanical engineering, engineering design methodology, industrial information technology and virtual product creation technologies. Impulses manly came from experiences in the branches automotive, software engineering, production technologies and renewable energy systems. The aim was to set up a simple guideline which supports the generation of requirements for an IPS^2 to be developed. The aim was not to work out a new theory of requirements engineering. All elaborations in this paper are in principal equally suitable for PSS and IPS^2. A conceptual publication preparing some of the results mentioned here has already been made in late 2008 [10].

Research approach

The investigation was executed in the following steps:

1. Clarification of need for systematic IPS^2 requirements generation.
2. Analysis of the state of the art in research and industry.
3. Compilation of results and solution approach.
4. Application and evaluation.
5. IT tool support implementation.

In step 1 and 2 we made a comparison of implicit IPS^2 from automotive industry, material flow solution providers and PLM vendors [11]. Some findings came from industrial workshops in the area of micro-financed electrification solutions for emerging countries (a brief report will be released in late spring 2010). Others came from interviews in German industry. A comprehensive literature study formed the basis for our new approach. The compilation will be described in the following sections. The development of a micro-manufacturing IPS^2 (show case scenario) with partners in the research project Transregio 29 "Engineering of Industrial Product-Service Systems" is used as an application case. An explorative examination of IT tools for requirements engineering helped to plan how to implement an IT tool support based on existing software tools.

Research questions (RQ)

The following research questions were set and, as far as possible, addressed in our study:

RQ1: Which domains should be considered for IPS^2 requirements generation? (Bandwidth)

RQ2: Which guidelines do exist in such domains?

RQ3: Which characteristics do the existing guidelines have in terms of content and outward appearance?

RQ4: Do we need a new guideline for IPS^2 and PSS requirements generation?

RQ5: Which type of support should be given by a new guideline for IPS^2 requirements generation?

RQ6: How can the existing guidelines be adopted or incorporated in a new guideline? (Compatibility)

2 LITERATURE REVIEW

2.1 Bandwidth of the study and new guideline (addressing RQ1)

As (industrial) product-service systems integrate products and services, the consideration of product development and service engineering is clear. As many functions of modern products and services base on information and communication technologies, IT approaches are obviously important. The rising amount of embedded systems requires an analysis of systems and requirements engineering approaches as well. As software, systems and requirements engineering are methodical and technical close domains we summarized those in one group. Table 1 summarizes references from such domains, which have been analyzed.

2.2 Methods to generate requirements, a small sample (addressing RQ2)

A small sample of guidelines is shown in this subsection to give the reader an idea of where our work is positioned. Not all guidelines and approaches are discussed in detail.

Table 1: Domains investigated in study.

Domain	Comment on focus	References investigated
Product Development	“Classic” references, foremost mechanical engineering	[12-21]
Service Engineering	Foremost European and Japanese references	[22-31]
Software, Systems, and Requirements Engineering	Generic references of organs and communities working on IT development, methodology and standards	[32-38]
PSS and IPS^2 Engineering	References, where requirements or quality criteria are addressed	[7, 39-41]

Product Development

Pahl and Beitz [18] propose a well-known guideline to retrieve requirements in mechanical engineering. This guideline refers to customers, designers, lifecycle phases, cost and time as sources for requirements and quality criteria. It considers many areas, which are addressed by design for X guidelines (e.g. Design-for-Assembly). The mentioned areas are broad and the designer has to search for required system properties on his own. The model of Ehrlenspiel [16] is hierarchical and has two main branches, (i) technical-economical requirements and (ii) organizational requirements. To search for requirements, these are detailed in a tree structure. Below the leaves of the tree (which are pure technical requirements, technical periphery, law, human, society, cost, time, staff, and tools) are “question terms” listed to retrieve requirements from various system lifecycle phases. These question terms are very heterogeneous, considering the abstraction level (examples of questioned issues: manufacturability, transport problems, maintenance duration, or training). Ahrens [12] compares various approaches to retrieve and manage requirements, but she does not compile a new checklist to retrieve requirements in product development.

Service Engineering

Jaschinski [24] elaborates on the process of a quality-oriented redesign of services but he gives no guideline or checklist, which helps to retrieve requirements on service properties in detail. In his thesis, van Husen [23] comes up with checklists to discover stakeholders and influencing factors (strategic, economic, legislative, and social factors, and boundary conditions) to analyse requirements for product-related services. Watanabe et al. [31] elaborate on the process of requirements negotiation embedded in a Japanese Service Engineering Methodology. Here a link is made between requirements and Receiver State Parameters, which represent the desired state of a customer after having received a service.

Software, Systems, and Requirements Engineering

Many sources contain general remarks on the importance of requirements engineering in IT development, cp. for instance [34], issues like IT security, interoperability etc. In some cases generic models like the CObIT cube, cp. Figure 2, define areas where requirements or quality issues come from. Detailed listings have not been found in public domain literature during our studies.

PSS and IPS^2 Engineering

Steinbach [41] delivers a comprehensive list of service characteristics and properties collated from business approaches and sources (examples of service properties mentioned: friendliness, responsiveness, patience, duration of delivery, reliability etc.). Berkovic et al. [39] considered product, service, and software engineering in their study and focused on the process and tools of the requirements engineering, but there was no elaboration on check criteria on requirements and quality.

2.3 Characteristics of existing methods and tools (addressing RQ3 and RQ4)

So far, there is no generic method, guideline or checklist, which addresses the system *IPS^2*, as described in section 1.1, as a whole in order to generate requirements.

Generalized, we found a weak consideration of

- use processes (lifecycle activities, system use-cases), contracts, and customer value in the area of product development literature.
- information and communication technologies as enabler for modern delivery processes in product development and service engineering literature.
- product and hardware specifications in service and software engineering.

This is not surprising as most models are domain specific.

Outer appearance

All models have different outer forms which hampers a straightforward integration of such guidelines for use in IPS^2 development.

Communality

The existing guidelines investigated in our study are foremost listings or simple models, which ether (i) systematize factors and sources influencing requirements and quality issues (cp. Figure 2) or (ii) capture system areas for which requirements have to be considered (cp. Figure 2). In most cases, the influencing factors are grouped or clustered (cp. Figure 2). These methods in general are not system (product, service, application), process or software tool specific. They have the characteristic of “awareness tools”, they are not real checklists.

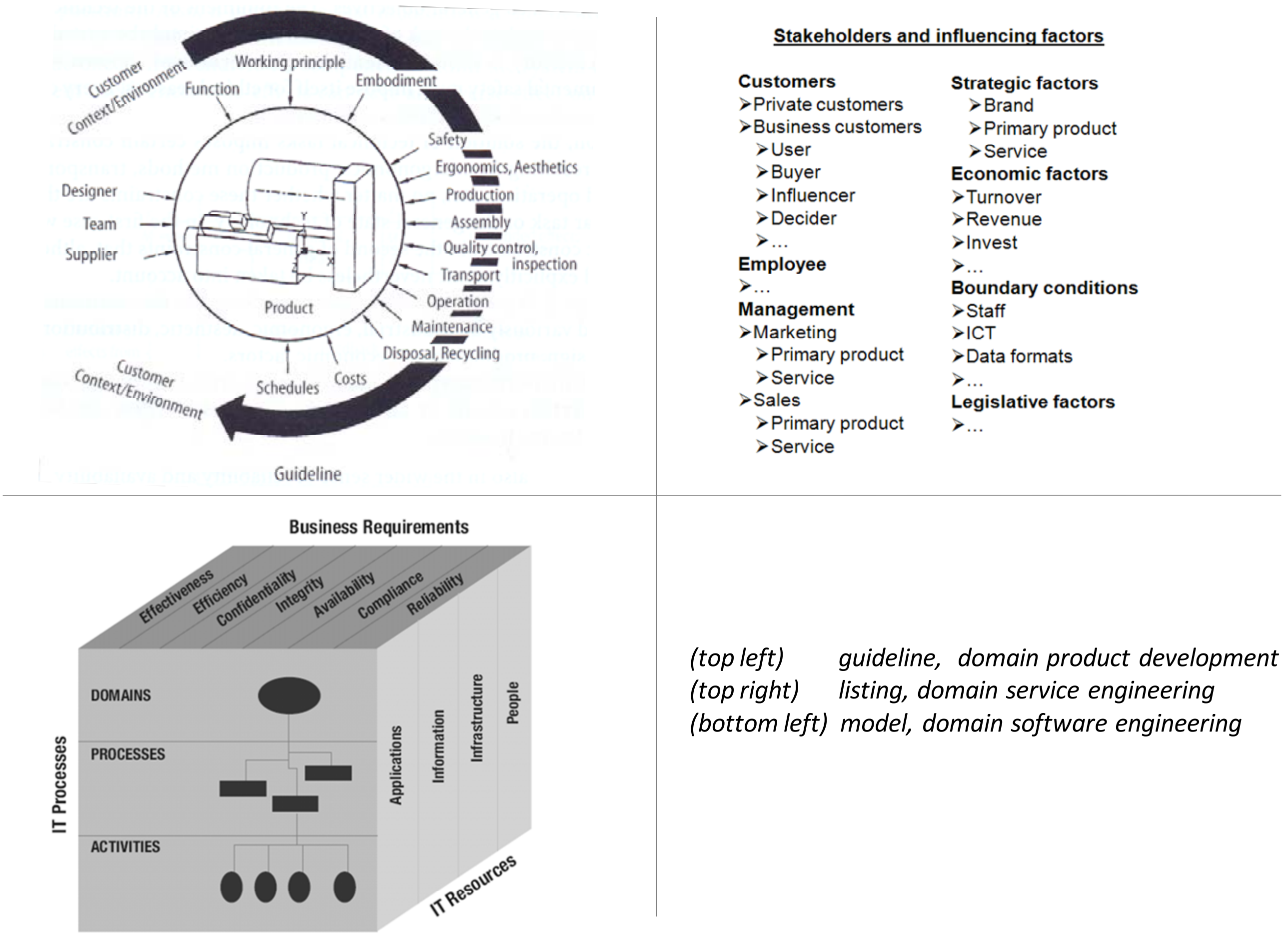

Figure 2: Guidelines and listings from product, service and software engineering after [18, 23, w5].

Software support

Software tools like Doors [w1], TeamCenter Engineering Requirements Management [w2], or in-Step [w3] support text based requirements management and link requirements to system elements or, for instance, UML diagrams for the systems specification. In general this is not depending on the system which has to be developed. The Japanese Service Explorer Tool [w4] implicitly captures requirements by aforementioned Receiver State Parameters. Thus, such tools do not depend on domain or application specific guidelines. The implementation of new domain specific guidelines or checklists in such software tools seems possible.

3 GUIDELINE – REQUIREMENT CHECKLIST FOR (INDUSTRIAL) PRODUCT-SERVICE SYSTEMS

3.1 Approach (addressing RQ5 and RQ6)

As none of the methods investigated offered the bandwidth needed for IPS^2 development and because none is directly adaptable for IPS^2 development we started to set up a new “checklist”. The checklist is predominantly based on clustered text listings which borrow many elements from the approaches investigated. An earlier conceptual version has been introduced in autumn 2008, but without such a deep dive into its background [10].

3.2 Characteristics of the checklist

The entire list has more than 100 entries, i.e. criteria to retrieve requirements. It is not the idea to “tick” every criterion for every potential system function or element when defining the requirements of an IPS^2. The aim is to make the designer aware of relevant influencing factors.

- Clusters help to keep the checklist in mind.
- The checklist is generic and open for customization to a specific branch, user group or type of IPS^2.

3.3 Clusters of the checklist

The checklist provides object and process oriented clusters (most in suitable pairs). The clusters *System* and *Behaviour* support a typical systems view on a very high level of abstraction. The clusters *Technical artefact* and *Service* address the core elements of an IPS^2. The clusters *Information* and *Communication* take into account that delivery processes contain a lot of information which is communicated by IT systems and actors. *Actors* and *Lifecycle activities* are core areas of interest in IPS^2 research and thus the next pair. Actors are performing activities in product use, service consumption and delivery. In lifecycle activities deliverables (products, information etc.) are generated to implement value for the customer. Creation of added *Value* is one of the major drivers for IPS^2 and considered in another cluster. *Business and operation models* and *Contracts* are rarely mentioned in most other methods, but they frame an IPS^2 and thus they should be considered equally.

Criteria collected from products, software, service and IPS^2 engineering have been allocated to the clusters. For some we used the original terms and for some we used synonyms, which were more suitable for IPS^2 development. Additional contributions we have made in all clusters based on experiences and theory.

Figure 3 illustrates the structure of the checklist. Table 2 and Table 3 summarize all collected and added criteria in

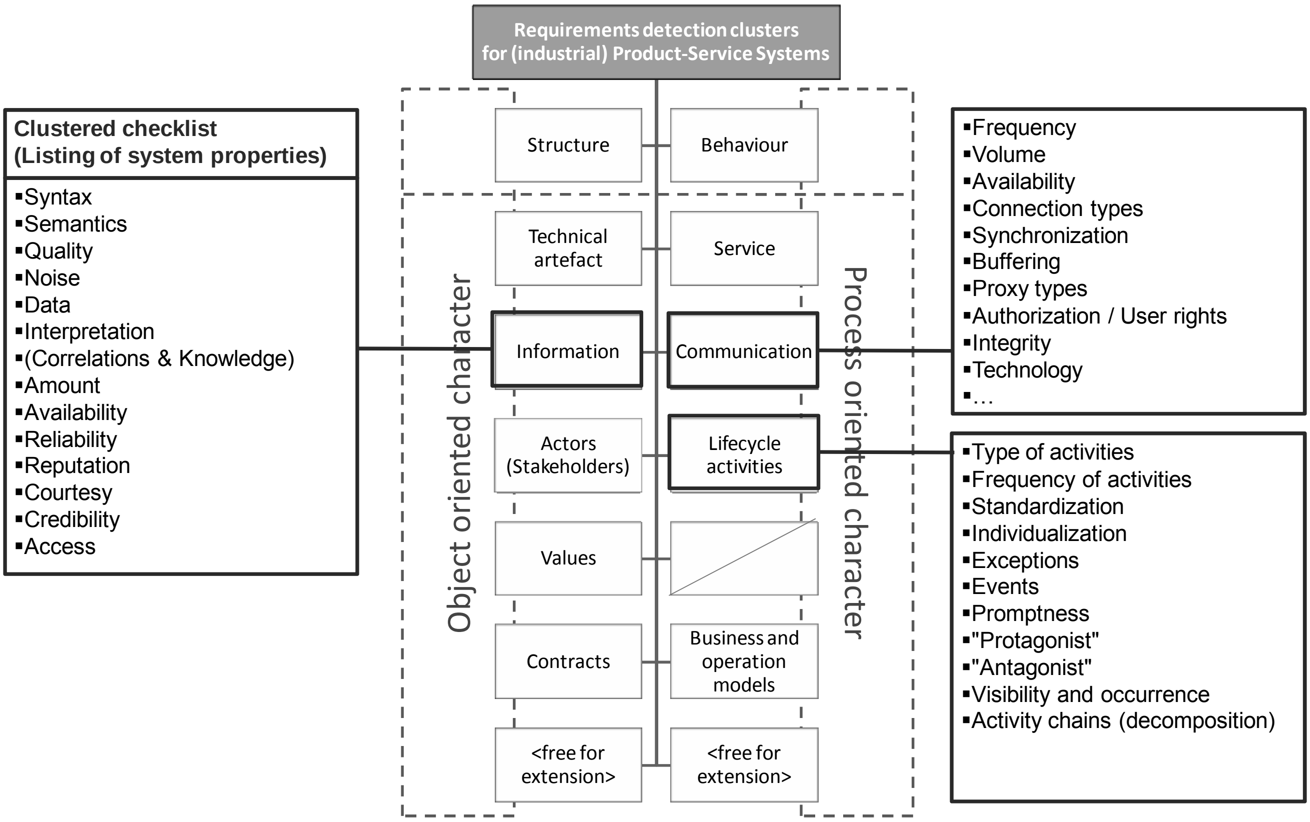

Figure 3: Clustered checklist for requirements generation.

compressed form. However, such a generic list cannot be complete. Some criteria can be allocated to more than one cluster if necessary for the user.

Example

A provider plans to offer a spindle for a milling process in a manufacturing system. The business model is availability-oriented. Now the checklist can be used to retrieve and formalize requirements. The following bullet list shows some requirements on a component level. The items in square brackets might be checklist criteria, which helped to find the particular requirement:

- Requirement 1: The activity maintenance of the deliverable spindle has to be carried out twice a year [LIFECYCLE ACTIVITIES → FREQUENCY] or the spindle functionality has to be monitored permanently to ensure an availability of 98%.
- Requirement 2: The activity maintenance has to fit the

Table 3: Checklist of criteria to generate requirements on (industrial) product-service systems (continuation).

Lifecycle activities	**Values**	**Contracts**	**Business and operation models**
• Type of activity • Frequency of activity • Standardization • Individualization • Exceptions • Events • Promptness • Visibility and occurrence • Activity chains (decomposition) • Min., max., mean duration • Earliest start, latest end • Facultative, optional or supplemental execution • Schedules • Process durations • Process type (management, core or support process) • Process trigger • Push or pull execution • Milestones	• Economic benefits (saved money, raised revenues) • Ecologic benefits (less resource consumption, less waste, ease of recycling and reuse) • Social benefits (access to education etc.) • Technologic values (technological advantage, technology substitution, higher efficiency) • Healthiness • Information and knowledge advantage • Time (saved time) • Training of skills • Flexibility, etc. • Functionality	• Incentives • Partners • Allocation of responsibilities • Duration • Conditions (if, else) • Options • Dependencies • Exceptions • Parameters coupled to market values or operation efficiency • Cost • Payment regulations • Fines • Allowances • Restrictions • Continuation conditions • Context • Risk allocation • Events • Exceptions • Policies	• Operation model (build-transfer-operate; build-operate-transfer, build-rent-transfer, build-operate-own) • Business Model (function oriented, availability oriented, result oriented) • Contract types (reimbursable, cost plus fee, fixed price engineering, engineering procurement construction, turn-key solution)

Table 2: Checklist of criteria to generate requirements on (industrial) product-service systems.

Structure	**Behaviour**	**Technical artefact (core products, periphery, infrastructure)**	**Service**
• Architecture and modularization • Elements and relations • Supplemental infrastructure (e. g. support and execution systems) • Supplemental services • System border set correctly set? • Stakeholders identified? • Input, throughput, output clear? • Lifecycle elicitated? • As-is processes and systems of customers know? • Needs and preferences of customers sufficiently know or anticipated? • Robustness	• Actions, reactions • Velocity of reactions • Delay of reaction • Stimuli • Accuracy • Repeatability • Flexibility, agility • Safety • Main functions • Overall behaviour (background working like operating system, watch-dog like permanent service, chronological execution, execution when requested like IT application)	• Main function (aim) • Related products/artefacts • Interfaces • Related activities • Related service offers • Availability • Robustness • Flexibility • Safety • Input, throughput, output • Required quantity • Design for X requirements • Ownership and "Usership" • Qualification level of user • Cost • Location of product operation	• Required resources • Related activities • Estimated results • Required information • Facultative services • Additional services • Supplemental services • Location of service application
Information	**Communication**	**Actors (Stakeholders, Personas, Players)**	
• Syntax • Semantics • Quality • Noise • Data • Interpretation • (Correlations & Knowledge) • Amount • Availability • Reliability • Reputation • Courtesy • Credibility • Access • Consistency • Traceability	• Frequency • Volume • Availability • Connection types • Synchronization (asynchronous, synchronous); Buffering, Proxy types • Authorization / User rights • Integrity • Technology • Interfaces • Infrastructure (support and execution systems) • Responsiveness • Security • Patience	• Receiver(s) • Supplier(s) • Provider • Individual needs ("personas") • Cultural needs • Individuals and roles • Units within networks • Virtual agents / software agents (e. g. broker) • Target groups • Qualification • Quantity • Authorization • Knowledge • Responsibilities • Empathy • Organisation	• Network type (e.g. "Virtual Enter.") • Fluctuation in network • Start of collaboration (e.g. ad-hoc) • End of collaboration (e.g. ad-hoc) • Duration of collaboration • Internal/external services or products to be delivered • In-sourcing / Outsourcing • Decision making in the network • Risks • Competencies • Authorization • Core processes • Partner, competitor, supporter, ...

customer's operation plan [LIFECYCLE ACTIVITIES → ACTIVITY CHAINS] in order to avoid interruptions in his manufacturing process.

- Requirement 3: In case of permanent monitoring by the provider the customer has to allow the provider access to his IT infrastructure by the contract [CONTRACTS].

The search criteria of the checklist can be used to tag requirements for computer based processing or to link or search requirements in a requirements list.

4 APPLICATION AND FINDINGS

The checklist has been applied during two industrial workshops, in a student project, and in the research project Transregio 29 [w6]. We learned that it is applicable by a moderator in an IPS^2 planning process, but that such a holistic approach contains too much information for users, which are not familiar with this list. The checklist helps to discover requirements quickly, precisely, and well structured within the planning and concept design for PSS. Furthermore, the checklist is relatively easy to implement in software tools due to its tree structure.

5 SUMMARY AND OUTLOOK

A comprehensive literature review has been undertaken and many criteria to generate requirements in product, service and software engineering have been identified. To make those applicable in IPS^2 development we composed a new checklist which includes information of the investigated approaches and many self added criteria. The checklist will be implemented in a project and requirements management software soon. It will be applied with a second method called *PSS Layer Method* [3, 10] for a model driven requirements generation.

6 ACKNOWLEDGMENTS

We gratefully acknowledge the German Research Foundation (DFG) for giving funds to enable our research. Without such funds our research would not be possible.

7 REFERENCES

[1] H. Meier, E. Uhlmann, D. Kortmann, 2005, Hybride Leistungsbündel, Nutzenorientiertes Produkt-verständnis durch interferierende Sach- und Dienst-leistungen, wt Werkstattstechnik online, 95: H. 7/8.

[2] Ericson, Å., Müller, P., Larsson, T., Stark, R., 2009, Product-service systems – From customer needs to requirements in early development phases, in: Roy, R., Shehab, E. (eds.), Proceedings of the 1st CIRP IPS2 Conference, Cranfield University, Cranfield, p. 62 – 67, ISBN: 978-0-9557436-5-8.

[3] Müller, P., Kebir, N., Stark, R., Blessing, L., 2009, PSS Layer Method – Application to microenergy systems, in: Sakao, T., Lindahl, M. (eds.), Introduction to Product/Service-System Design, Elsevier.

[4] A. Tukker, 2004, Eight Types of Product-Service System: Eight Ways to Sustainability? Experiences from Suspronet, Business Strategy and the Environment, 13: 246 – 260.

[5] Meier, H., Uhlmann, E., Kortmann, D., 2005, Hybride Leistungsbündel – Nutzenorientiertes Produktverständnis durch interferierende Sach- und Dienstleistungen, wt Werstattstechnik online, Jahrgang 95, H. 7/8.

[6] Meier, H. (ed.), 2004, Dienstleistungsorientierte Geschäftsmodelle im Maschinen- und Anlagenbau, Springer-Verlag, Berlin, Heidelberg.

[7] Uhlmann, E., Stelzer, C., Geisert, C. Bochnig, H. Meier, H., Sadek, K., 2008, Design of PSS based on Customer Requirements, in: Proceedings of the International Seminar on PSS, TR29, Bochum.

[8] Kruse, P. J., 1995, Anforderungen in der Systementwicklung, VDI Fortschrittberichte Reihe 20, Nr. 191, 1996, VDI Verlag GmbH, Düsseldorf.

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

[10] Müller, P., Stark, R., 2008, Detecting and structuring requirements for the development of product-service systems, in: Meerkamm (ed.), Konferenzband zum 19. Symposium „Design for X", Neukirchen, S. 1–10, ISBN: 978-3-9808539-6-5.

[11] Stark, R., Müller, P., 2009, Product-Service System Methodologies in Research and Industry, in: Proceedings of the International Seminar on IPS2, TR29, Berlin.

[12] Ahrens, G., 2000, Das Erfassen und Handhaben von Produktanforderungen - Methodische Voraussetzungen und Anwendungen in der Praxis, Maschinenbau und Produktionstechnik, TU Berlin, Berlin.

[13] Deubzer, F., Lindemann, U., 2009, Networked Product Modeling - Use and Interaction of Product Models and Methods during Analysis and Synthesis, International Conference on Engineering Design, ICED'09, Stanford University, Stanfort, CA, USA, 24-27 August .

[14] DIN e.V.,1997, DIN EN ISO 9241: Ergonomie der Mensch-Maschine-Interaktion, Beuth Verlag GmbH, Berlin.

[15] DIN e.V., 2007, DIN EN ISO 10993: Biologische Beurteilung von Medizinprodukten, Beuth Verlag GmbH, Berlin.

[16] Ehrlenspiel, K., 1995,Integrierte Produktentwicklung – Methoden für Prozessorganisation, Produkterstellung und Konstruktion, Carl Hanser Verlag, München, Wien.

[17] Hubka, V., 1984, Theorie Technischer Systeme – Grundlagen einer wissenschaftlichen Konstruktionslehre, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo.

[18] Pahl, G., Beitz, W., Feldhusen, J., Grote, K. H., 2007, Engineering Design - A Systematic Approach, 3rd ed., Springer-Verlag, London.

[19] Verein Deutscher Ingenieure, 1982, VDI 2222 Blatt 2: Erstellung und Anwendung von Konstruktionskatalogen, VDI-Verlag GmbH, Düsseldorf.

[20] Verein Deutscher Ingenieure, 2004, VDI 2223: Methodisches Entwerfen technischer Produkte, Beuth-Verlag GmbH, Berlin.

[21] Fraunhofer, 2008, Projektmanagement im Anlagenbau, Arbeitsbericht 8. Industriearbeitskreis "Kooperation im Anlagenbau", Fraunhofer IRB Verlag.

[22] Benkenstein, M., Güthoff, J., 1996, Methoden zur Erfassung der Qualität komplexer Dienstleistungen. Marketing und Marktforschung, pp. 514 - 536.

[23] van Husen, C., 2007, Anforderungsanalyse für produktbegleitende Dienstleistungen, Jost-Jetter-Verlag.

[24] Jaschinski, C., 1998, Qualitätsorientiertes Redesign von Dienstleistungen, FIR, RWTH Aachen.

[25] Kaner, M., Karni, R., 2007, Design of service systems using a knowledge-based approach, Knowledge and Process Management, 14/4: 260-274.

[26] Parasuraman, A., Zeithaml, V., Berry, L., 1985, A conceptual Model of Service Quality and its Implications for future Research, Journal of Marketing, 49: 41-50.

[27] Ramaswamy, R., 1996, Design and Management of Service Processes, Addison Wesley Publishing Company Inc.

[28] Scharitzer, D., 1994, Dienstleistungsqualität - Kundenzufriedenheit (Broschüre), Service Fachverlag, Wien.

[29] Schwarz, W., 1997, Methodisches Konstruieren als Mittel zur Systematischen Gestaltung von Dienstleistungen, Druckhaus Berlin-Mitte GmbH, Berlin.

[30] Webster, F. E., Wind, Y.,1972, A General Model for Understanding Organizational Buying Behavior, Journal of Marketing, 36: 12-19.

[31] Watanabe, K., Kimita, K. Tateyama, T., Shimomura, Y., 2009, Requirement negotiation for feasible service development, ICED'09, Stanford.

[32] Bundesministerium für Sicherheit in der Informationstechnik: E-Government Handbuch (Loseblattsammlung). Bundesanzeiger Verlag, Köln. ISBN: 3-89817-180-9. online: https://www.bsi.bund.de/cln_134/ContentBSI/Themen/Egovernment/EgovernmentHandbuch/Onlineversion/onlineversion.html (14.09.09, 14:00).

[33] The Institute of Electrical and Electronics Engineers, Inc., 1998, IEEE Recommended Practice for

Software requierements Specifications, ISBN: 0-7381-0332-2.

[34] Kruchten, P., 2006, The rational unified process: an introduction, 3rd ed., Addison-Wesley.

[35] Rupp, Ch., 2004, Requierements-Engineering und -Management, 3rd. Ed., Carl-Hanser-Verlag, München Wien.

[36] Sommerville, I., Sawyer, P., 1997, Viewpoints: principles, problems and a practical approach to requieremts engineering, Annals of Software Engineering, 3: 101-130.

[37] VDI-Handbuch Energietechnik, 2007, VDI 4602: Energiemanagement, Beuth-Verlag GmbH, Berlin.

[38] Wolf, S., 2006, IT-Governance mit ITIL, CObIT und der Balanced Scorecard (Diplomarbeit), Hochschule Niederrhein.

[39] Berkovic, M., Esch, S., Leimeister, J. M., Krcmar, H., Requirements Engineering for Hybrid Products as Bundles of Hardware, Software and Service Elements – A Literature Review.

[40] Meier, H., Kortmann, D., Golembiewski, M., 2006, Hybride Leistungsbündel in kooperativen Anbieter-Netzwerken, Anforderungen hybrider Leistungsbündel an die unternehmensinterne und kooperative Organisation von Anbieter-Netzwerken, Industrie Management 22: 25-28, GITO-Verlag.

[41] Steinbach, M., Bley, H. (edt.), Weber, C. (edt.), 2005, Systematische Gestaltung von Product-Service Systems – Integrierte Entwicklung von Product-Service Systems auf Basis der Lehre von Merkmalen und Eigenschaften, Lehrstuhl für Konstruktionstechnik/CAD, Saarbrücken.

[42] Müller, P.; Sakao, T., 2010, Towards Consolidation on Product-Service Systems Design. CIPR IPS2 Conference, Linköping.

WEB LINKS

[w1] Rational DOORS, IBM: http://www-01.ibm.com/software/awdtools/doors/

[w2] Teamcenter Engineering, SIEMENS UGS: http://www.plm.automation.siemens.com/en_us/products/teamcenter/

[w3] in-Step, microTOOL: http://www.microtool.de/instep/

[w4] Service Explorer, Service Engineering Forum, Japan: http://www.service-eng.org/index-e.html

[w5] CObIT cube figure: http://iea.wikidot.com/local--files/cobit/COBIT_Cube.png

[w6] Transregio 29: www.tr29.de (Last request of all web links on March 12th, 2010.