REQUIREMENTS ENGINEERING FOR HYBRID PRODUCTS AS BUNDLES OF HARDWARE, SOFTWARE AND SERVICE ELEMENTS – A LITERATURE REVIEW

Marina Berkovich, Sebastian Esch[1], Jan Marco Leimeister[2], Helmut Krcmar[1]

Abstract

In this paper we compare different approaches and show the need for systematic requirements engineering for hybrid products beyond disciplinary boundaries. Hybrid products consist of combinations of hardware, software and service elements. The purpose of this paper is to report on a literature review on requirements engineering for hybrid products. Each academic discipline involved (software engineering, product engineering and service engineering) has a different view on requirements engineering. The goal of the literature review is to discover how the approaches of each discipline are able to cope with requirements engineering for hybrid products.

1. Introduction

Increasing competition and increasing technological development result in high time-, cost- and development pressure. These trends induce the need for differentiation for most companies [32, 34]. Thus companies are compelled to individualize their products and services regarding to customer needs. The companies' goals are to improve the innovation processes continuously and to increase their ability to determine, to track and to cover customer-needs better than competing companies. The improvement of the ability to innovate is an essential factor for the prosperity and growth of companies [10].

Hauschildt [14] defines innovations as qualitative new products or methods, which noticeably differ – however this is measured – from the prior state. The novelty has to be realized. The sole generation of a new idea is not sufficient; sale or utilization distinguishes between innovation and invention.

Therefore companies develop more and more complex solutions. These complex solutions consist of combinations of product-, software and service elements. The distinction between product and service becomes blurred and a so called hybrid product emerges [23]. A hybrid product is a

[1] Technische Universität München, D-85748 Garching b. München, Boltzmannstr. 3

[2] Universität Kassel, D-34127 Kassel, Nora-Platiel-Str. 4

complex solution consisting of several parts, usually hardware, software and service elements, which are not easily, recognized as single parts, but different characteristics of these three parts define the hybrid product [23]. Especially the development of hybrid products is different because of the high number of sub-elements, the high level of technological integration and the degree of customer-integration [23, 37]. But in the literature the development of products and services are elaborated separately.

Hybrid products have some differences compared with other areas. Hybrid products are more complex as products consisting of one part as hardware, software or service element [23]. The trend to customer individuality is also one of the important aspects characterizing these products.

The lifecycle of hybrid products is considered differently as traditional products. The company that builds hybrid products does also offer services in combination with the product, what means in practice that the later operation of the product is also important. The operator model changes from "build and forget" to "build, operate & advance". Ultimately also the business model of the company changes to a "pay per use" philosophy.

The individual components of these hybrid products have different development and manufacturing periods and are developed by different functional areas within a company. These aspects cause various challenges for companies.

2. Requirements Engineering

Requirements engineering (RE) is a crucial aspect for a successful development of products and software [5, 24]. A dominant understanding of RE in software engineering defines requirements engineering as consisting of requirements analysis, prioritization and negotiation of requirements, the activities to control and to administer requirements of a system under development. The change- and realization-management and the validation of requirements are also parts of RE.

Strong evidence that RE is especially important gives [33] which states that "surveys which have been carried out so far suggest that, for large hardware/software systems, about 15% of the total budget is taken up by requirements engineering activities." All changes of requirements cause modifications of parts of the solution and increase development costs. Requirements engineering has to capture and to map the origin and the context of changes. Furthermore the impact of changes should be made assessable. Another important mission of tracking changes is to anticipate future changes of requirements.

The development of hybrid products is influenced by the disciplines product-, software- and service-engineering. Thus, the development of hybrid products needs a special designed process of requirements engineering. In order to develop the requirements engineering for hybrid products we analyze whether the RE in each of the discipline is appropriate for hybrid products.

3. The State of the Art of RE in the Literature

Most development process models in product engineering, software engineering and service engineering consider requirements engineering only during the first phases of the overall development process. The goal of the requirements engineering's activities is to elicit and specify the requirements. The fact that missing or incorrect requirements have negative effects on the success of the product is widely recognized [5, 24].

The development process of hybrid products involves different stakeholders and components. In the following sections we describe and compare different approaches in product engineering, software engineering and service engineering, each of them specialized in developing mainly one component: (physical) products, software or services.

3.1. Analyzed aspects

In software engineering requirements engineering has already been accepted as an independent discipline and is done systematically. Many concepts and methods for handling of requirements have already been elaborated there. The development of process models has been elaborated there, which is especially important for hybrid products [23]. From the requirements engineering in software engineering we derived the aspects based on the phases of the requirements engineering process and proper questions listed below:

- Requirements elicitation: How are the requirements elicited? What methodical support is available?
- Requirements analysis: How are the requirements analyzed? How are the requirements translated to the language of the developer? What methodical support is available?
- Requirements negotiation: How are conflicts between requirements resolved? Are priorities assigned to requirements? What methodical support is available?
- Documentation: How are the requirements documented? Which information is documented?
- Validation: How are the requirements validated for completeness, correctness and consistency?
- Change management: How are changes of requirements managed?

Especially the next aspects are important for the literature review:

- Systematic requirements engineering: whether the requirements engineering is evaluated in all phases of the development process.
- Methods for formal specification of requirements: are there methods for a formal specification of requirements?
- Methods for requirements elicitation are described: are the methods of requirements elicitation described?
- Tool-support: are there tools supporting the process of requirements engineering
- Documentation and management of changes of requirements: are changes of requirements documented and is the change management process defined?
- Whether all phases of the innovation process are covered: are all phases of the innovation process considered in requirements engineering?
- Whether hybrid products are captured: does the process of requirements engineering consider hybrid products?

3.2. Sources of RE-approaches: Product Engineering

Jung [20] defines a requirement as a request that a product fulfills certain properties or functions. A requirement can be posed consciously or unconsciously by any person that has an interest in justifying the requirement. A requirement consists of a describing attribute and a defining value (quantification).

Most development approaches in product engineering take requirements engineering into consideration [1, 3, 7, 9, 11, 16, 21, 24, 27, 35]. Some examples illustrate the common views of these approaches. Cross [7] suggests a goal tree, which is used to vaguely collect the initial requirements. During the development process the requirements are refined as the problem-understanding of the customer and engineers increases. Ehrlenspiel [11] defines an iterative process for eliciting the requirements. He suggests using checklists to support the customer interviews and to overcome misunderstandings between the customer and the engineer. Lindemann [24] has developed a set of methods for requirements elicitation. These methods are not used in a predefined order and can be applied repeatedly. Examples for methods involving the customer are: checklists, mind maps and questionnaires. A sequential process model is suggested by Pahl [27]. He states that the engineer has to extract the requirements from the customer's wishes. He also recognizes that the customer is often not able to express his requirements appropriately, but he does not suggest methods for eliciting the requirements. Ulrich [35] collects the requirements in hierarchical weighted lists. The author states that it is important to reveal implicit customer-needs and that a common product-understanding between customer and engineer is absolutely necessary.

Most authors demand that changes of requirements are documented in form of requirement-lists. The origin and cause of changes is hardly documented. Interdependencies between requirements are captured by creating links between requirements, but an anticipatory handling of requirements is missing.

The analysis of the mentioned literature revealed that RE in product engineering is mostly restricted to the early phases of the development process. During the late phases RE does not seem to play a substantial role. Most of these approaches state that the customer plays a central role during the entire development process, but type and degree of customer integration into the development process varies. The integration of the customer into the process of requirements elicitation is emphasized, but not in later phases [20].

The methods of product engineering deal only with requirements for products. They do not mention that a product can consist of hardware and software or even of a bundle including services. The inclusion of services into the development process is not mentioned.

3.3. Sources of RE-approaches: Software Engineering

The term "requirement" is defined by IEEE-Standard IEEE 610.12-1990 [17] as follows: "A requirement is a (1) condition or capability needed by a user to solve a problem or achieve an objective; (2) condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents; (3) documented representation of a condition or capability as in (1) or (2)".

In contrast to the definition of product engineering, this definition considers the role of the user. A functionality does not exist on its own, but the user needs it, to fulfill his tasks. Additionally, this definition states the origin of the requirements. Because the user will use the finished software, he is the main source of requirements. Therefore the integration of users into the requirements elicitation is a decisive factor for the success of software-projects [22].

Boehm [4] defines requirements engineering as follows: „Software requirements engineering is the discipline for developing a complete, consistent unambiguous specification – which can serve as a basis for common agreement among all parties concerned – describing what the software product will do (but not how it will do it, this is to be done in the design specification)“.

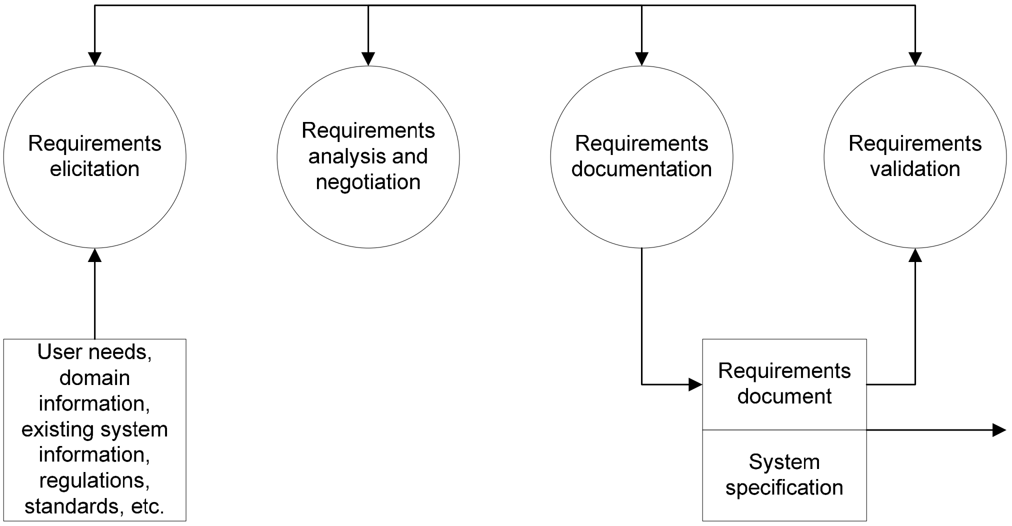

Requirements elicitation
Requirements analysis and negotiation
Requirements documentation
Requirements validation
User needs, domain information, existing system information, regulations, standards, etc.
Requirements document
System specification

Figure 1: RE process (according to [33])

In *Figure 1* the RE process is depicted according to Sommerville and Kotonya [33]. The authors give the following definition of the process of requirements engineering: “A requirements engineering process is a structured set of activities which are followed to derive, validate and maintain a systems requirement document. Process activities include requirements elicitation, requirements analysis and negotiation and requirements validation.”

As the arrows in *Figure 1* indicate requirements engineering is not a linear process but an iterative process. The outcome of the RE process is the requirements document and the system specification respectively. The main processes are described below:

- Requirements elicitation: The task of requirements elicitation is the identification of requirements’ sources and the elicitation of requirements according to the identified stakeholders and other requirements’ sources [28]. The methods applied for elicitation of requirements are interview, workshop, observation and perspective-based reading, also brainstorming, prototyping, Mind Maps and checklists [28]. Also [33] suggest interview and observation as methods for eliciting the requirements. Sources of requirements among others can be users, domain knowledge, existing systems and regulations.
- Requirements analysis and negotiation: The goal of requirements analysis and negotiation is to solve conflicts that exist between requirements and eventually get a consistent set of requirements [28]. Conflicts exist because different stakeholders have different interests in the system and different views on the system. Mostly the win-win-approach is applied to solve the conflicts [28].
- Requirements documentation: Pohl [28] suggests natural language based and model based forms of specifications. Model based documentation is based on modeling languages such as UML. To express the importance of each requirement, priorities are assigned to requirements. There exist many methods for prioritizing requirements.
- Requirements validation: The goal of requirements validation is to check if the requirements specify the system as the customer wants and needs it. The methods for validation presented by Sommerville and Kotonya [33] are reviews, prototyping and testing. Pohl [28] suggests inspection, reviews, walkthroughs, perspective-based reading and prototypes.

Aurum and Wohlin [2] give a comprehensive compilation of practices used in RE in software engineering. In this paragraph we give a short overview of these practices. The methods requirements elicitation, specification and modelling mean "understanding the needs of stakeholders, eliciting requirements, modelling and collecting them in a repository" [2]. Here we see that software engineering is aware that there are several stakeholders with different interests in the product. To resolve these conflicts the activities of negotiation and prioritization are introduced [28]. Prioritizing requirements "assists project managers with resolving conflicts […], plan for staged deliveries, and make necessary trade-off decision" [2].

Relationships between requirements determine factors for choosing the way of software development work [8]. In order to obtain a connection between requirements engineering, design, implementation and test, requirements traceability is introduced. Requirements traceability is defined as "the ability to describe and follow the life of a requirement, in both a forwards and backwards direction" [13]. Basing on the traceability the impact analysis tries to identify the parts of a system to be changed, if a requirement is changed and to identify the consequences on the system if the change is done [19].

We conclude that RE in software engineering is highly elaborated and many methods and process models are known. Software engineering sees RE as a continuous activity which is performed throughout the entire development process, while product engineering considers RE as a phase at the beginning of the development.

3.4. Sources of RE-approaches: Service Engineering

The term service engineering is defined by Gill [12] as "the systematic development of predominantly technical services by deploying engineering methods, practices and by using tools of the engineering design field".

According to Schmitz [30] service engineering in large part is handled by marketing divisions. Schmitz [30] sees requirements engineering as a task carried out by qualitative marketing research. To elicit the requirements the customer should imagine the service and should then express a hypothetical evaluation.

Ramaswamy [29] understands services as transaction processes. He suggests separating the quality of services into quality of service design and quality of service delivery. For the process of developing services, the requirements to the services are crucial. The process model of Ramaswamy consists of various steps. In the step "Defining Design Attributes" the key-customers and their expectations and requirements are elicited and prioritized. So we can see that the elicitation of requirements is intended, but no precise methods are provided. The management of requirements is not mentioned.

The process model of Schneider et al. [31] envisions the elicitation of customer requirements during the concept phase. Methods and tools for RE are not provided.

We conclude that there are process models for service engineering. These models describe the process on a very high level; hence tangible methods are not given. We can also show that most approaches mention the customer as a source of requirements. According to Zahn [36] the integration of the customer is an important factor for the development of services. But precise methods and applicable tools can hardly be identified and therefore the role of RE in Service Engineering is very vague.

4. Conclusion

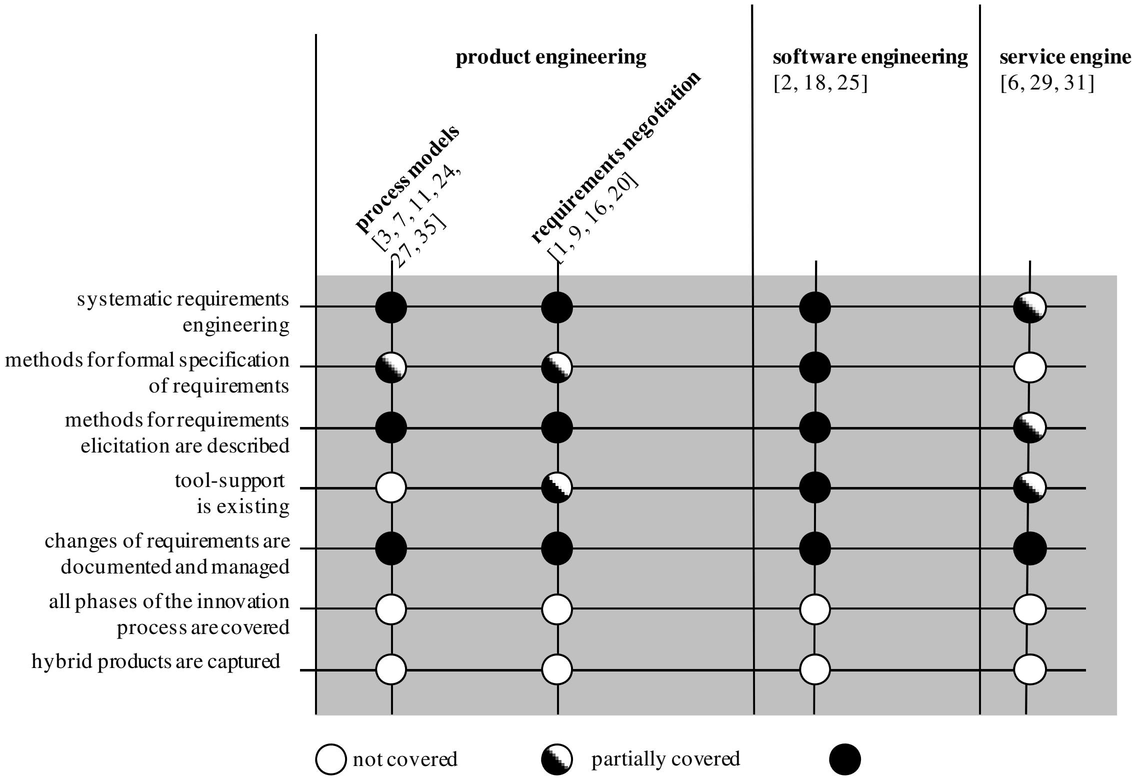
The importance of RE to successfully conduct development processes has been recognized by all three disciplines of product-, software- and service-engineering.

In software engineering RE is applied throughout the development process and there are a large number of methods for eliciting and managing requirements.

In product engineering RE is also widely accepted. The handling of requirements is integrated in the engineering-process. Most approaches consider RE only during the first phases of the innovation process. Origin, context and period of validity of requirements are not captured.

Service engineering is still a young discipline and lacks support for systematic approaches [6, 15]. There exist some general process models, which mention RE, but offer no tangible methods. The abstraction level of these models is very high, so they are not relevant for practical use. Methods for elicitation and handling of requirements are treated only rudimentarily.

In *Figure 2* some aspects of RE are set in connection to the RE process of the three disciplines. The criterions are derived from the aspects covered in the paper. We can see that the RE in software engineering is most elaborated and RE in service engineering has the biggest gaps.



	product engineering		**software engineering** [2, 18, 25]	**service engineering** [6, 29, 31]
	process models [3, 7, 11, 24, 27, 35]	**requirements negotiation** [1, 9, 16, 20]		
systematic requirements engineering	●	●	●	◐
methods for formal specification of requirements	◐	◐	●	○
methods for requirements elicitation are described	●	●	●	◐
tool-support is existing	○	◐	●	◐
changes of requirements are documented and managed	●	●	●	●
all phases of the innovation process are covered	○	○	○	○
hybrid products are captured	○	○	○	○

○ not covered ◐ partially covered ●

Figure 2: aspects of requirements in the three disciplines

In *Figure 2* we can see that especially the aspects that are most relevant for hybrid product are not covered by the existing approaches of RE. Also tool-support and methods for formal specification of requirements should be improved.

Hybrid products need customer orientation [23, 37]; to guarantee that the customer-need is satisfied, the customer has to be integrated into the whole development process, but none of the approaches covers all phases of the innovation process.

Services are a part of the hybrid product [23, 37], but the RE in product development and software engineering does neither mention the integration of services nor the parallel development of them. Additionally the RE approaches of product engineering do not cover software parts of products. In the RE of software engineering hardware is considered as already existing and is taken into account as an unchangeable "system context". Because of this lacks in the approaches all the four circles for "hybrid products are captured" in *Figure 2* are selected as "not covered".

5. Limitations and Outlook for Future Work

This literature review covered only the most dominant literature found in these domains. It could only present the most significant differences and deficiencies of RE for the development of hybrid products – more detailed and thorough analyses are required.

Future work should develop models, modelling methods and supporting tools to cover the RE for hybrid products. It is particularly important to support all phases of the innovation process. Therefore models have to capture changes of requirements and interdependences between requirements and the methods have to deal with them accordingly.

To meet the challenges of RE for hybrid products, models are needed, that are capable of treating the requirements and the product as a whole. The models must also be able to capture service packages and its interplay of hardware-, software- and service components along the entire development process appropriately. By comparing the existing approaches and industrial practices it should be possible to develop a transdisciplinary understanding of requirements and its handling. Future research has to answer - amongst others - the following questions: Which methods and models are used by the RE in practice? Which methods and models presented in the literature are used in practice? Which are success and failure factors of RE in practice? Which activities must be supported by a process model for RE in the field of hybrid products? Is an integrated RE supporting all phases of the innovation process possible, reasonable and applicable?

6. References

[1] AHRENS, G. Das Erfassen und Handhaben von Produktanforderungen, Dissertation der TU Berlin, Berlin, 2000.

[2] AURUM, A. WOHLIN, C. Requirements Engineering: Setting the Context, in: Engineering and Managing Software Requirements, Springer, Heidelberg, 2005.

[3] BREIING, A., FLEMMING, M. Theorie und Methoden des Konstruierens, Springer, Berlin, Heidelberg, 1993.

[4] BOEHM, B.W. Guidelines for verifying and validating software requirements and design specifications, EURO IFIP 79, North Holland 1979, S. 711-719

[5] BROOKS, F. P., JR. "No Silver Bullet: Essence and Accidents of Software Engineering, in: Computer, Vol. 20, No. 4, April 1987, pp. 10-19.

[6] BULLINGER, H.-J., SCHEER, A.-W. Service Engineering – Entwicklung und Gestaltung innovativer Dienstleistungen, 2nd Edition, Springer, Berlin, 2005.

[7] CROSS, N. Engineering Design Methods, Wiley, Chicester, 1996.

[8] DAHLSTEDT, A. G., PERSSON, A. Requirements Interdependencies: State of the Art and Future challenges, in: Engineering and Managing Software Requirements, Hrsg.: Aurum, A., Wohlin, C., 1st Edition, Springer, Berlin, 2005.

[9] DANNER, S. Ganzheitliches Anforderungsmanagement, Shaker, Aachen, 1996; Diss. der TU München, 1996.

[10] EDGETT, S. J., COOPER, R. G. New Product Development, Ancaster, Product Development Institute, Ontario 2005.

[11] EHRLENSPIEL, K. Integrierte Produktentwicklung, Hanser, München, 2003.

[12] GILL, C. Architektur für das Service Engineering zur Entwicklung von technischen Dienstleistungen, 1. Auflage, Shaker, 2004.

[13] GOTEL, O.C.Z., FINKELSTEIN, C.W. An analysis of the requirements traceability problem, in: Proceedings of the First International Conference on Requirements Engineering, Colorado Springs, CO., 1994.

[14] HAUSCHILDT, J., SALOMO, S. Innovationsmanagement, 4td Edition, Franz Vahlen GmbH, München, 2007.

[15] HERRMANN, T., KLEINBECK, U., KRCMAR H. Konzepte für das Service Engineering: Modularisierung, Prozessgestaltung und Produktivitätsmanagement, 1st Edition, Physica, Heidelberg, 2005.

[16] HUMPERT, A. Methodische Anforderungsverarbeitung auf Basis eines objektorientierten Anforderungsmodells, HNI-Verlagsschriftenreihe, Paderborn, 1995.

[17] IEEE Standard Glossary of Software Engineering Terminology, The Institute of Electrical and Electronics Engineers, New York, 1990, IEEE Std. 610.12-1990.

[18] JACOBSON, I., BOOCH, G., RUMBAUGH, J. Unified Software Development Process: The complete guide to the Unified Process from the original designers, Addison-Wesley Longman, Amsterdam 1999.

[19] JÖNNSON, P., LINDVALL, M. Impact Analysis, in: Engineering and Managing Software Requirements, Hrsg.: Aurum, A., Wohlin, C., Springer, Berlin, 2005.

[20] JUNG, C. Anforderungsklärung in interdisziplinärer Entwicklungsumgebung, Dr. Hut, Aufl. 2006.

[21] KRUSE, P.J. Anforderungen in der Systementwicklung, VDI-Verlag, Düsseldorf, 1996.

[22] KUJALA, S. M., KAUPPINEN, L., LEHTOLA, KOJO, T. The role of user involvement in requirements quality and project success, in: 13th IEEE International Conference on Requirements Engineering, Paris, 2005, pp. 75 – 84

[23] LEIMEISTER, J. M.; GLAUNER, C. Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik, in: Wirtschaftsinformatik, Vol. 50, No. 3, 2008 .

[24] LINDEMANN, U. Methodische Entwicklung technischer Produkte, Springer, Berlin, Heidelberg, New York, 2005.

[25] MARSCHALL, F., SCHOENMARKERS, M. Towards model-based Requirements Engineering for web-enabled B2B Applications, in: Proceedings of the 10th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), 2003, S. 312 – 320

[26] MEIREN, T. Service Engineering – Systematische Entwicklung von kunden- und mitarbeiterorientierten Dienstleistungsprodukten, in: Technologiemanagement in der Praxis: Forschen und Anwenden, Hrsg.: Spath, D., Fraunhofer IRB , Stuttgart, 2006.

[27] PAHL, G., BEITZ, W., FELDHUSEN, J., GROTE, K.H. Konstruktionslehre, Springer, Berlin, Heidelberg, New York, 2003.

[28] POHL, K. Requirements Engineering. Grundlagen, Prinzipien, Techniken, 1. Auflage, Dpunkt Verlag, 2007.

[29] RAMASWAMY, R. Design and Management of Service Processes, Addison- Wesley, Massachusetts, 1996.

[30] SCHMITZ, G. Die Ermittlung der Kundenanforderungen an industrielle Dienstleistungen, in: Zeitschrift für Planung, 2000, Nr. 2, pp. 195-215

[31] SCHNEIDER, K., DAUN, C., BEHRENS, H., WAGNER, D. Vorgehensmodelle und Standards zur systematischen Entwicklung von Dienstleistungen, in: Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen, 2. Auflage, H.-J. Bullinger, A.- W. Scheer, Springer, Berlin, 2006, pp. 113-138.

[32] SIMONIS, G. Die Zukunftsfähigkeit von Innovationen: das Z-Paradox., in: D. Sauer; Ch. Lang. Paradoxien der Innovation, Campus, Frankfurt/New York, 1999, pp. 266

[33] SOMERVILLE, I.; KOTONYA, G., Requirements Engineering: Processes and Techniques Wiley & Sons, 1998.

[34] SPATH, D., DILL, C., SCHARER, M. Vom Markt zum Markt, LOG_X, Stuttgart, 2001.

[35] ULRICH, K. T., EPPINGER, S. D. Product Design and Development, McGaw-Hill, New York, 2003.

[36] ZAHN, E., SPATH, D., SCHEER, A.-W. Vom Kunden zur Dienstleistung - Methoden, Instrumente und Strategien zum Customer related Service Engineering, Frauenhofer IRB Verlag, Stuttgart, 2004.

[37] ZELLNER, G. Gestaltung hybrider Wertschöpfung mittels Architekturen – Analyse am Beispiel des Business Engineering. in: Wirtschaftsinformatik, Vol. 50, No. 3, 2008.