INTERNATIONAL DESIGN CONFERENCE - DESIGN 2016
Dubrovnik - Croatia, May 16 - 19, 2016.

ORGANIZING END USER REQUIREMENTS FOR PRODUCT-SERVICE ENGINEERING PLATFORM

I. Karvonen, T. Ryynänen, K. Jansson and H. Korhonen

Keywords: product-service design, requirements, structuring

1. Introduction

Manufacturing companies are increasingly moving towards servitization, i.e. offering product-services or services related to their products to their customers. The aim is to deliver more benefit to the customer and at the same time to obtain additional business for the manufacturer. A Product Service (PS) is defined as ”a mix of tangible products and intangible service designed and combined so that they are jointly capable of integrated, final customer needs” [Tukker and Tischner 2004]. There is also a concept of “extended product” which is used to describe products, where the traditional, physical product is extended with an intangible part, for example information, service or other customer benefits [Thoben et al. 2003]. Thus product services are a special case of extended products and also services may be extended with other intangible components.

The increasing interest for providing product services creates a need for a more efficient product service design and engineering. It should be possible to design the service concurrently when designing the product. The service design and provision requires efficient management of product life cycle information. Product-related services may even be built on utilizing the product information. This causes additional needs for engineering platforms, both from the functional and stakeholder point of view. For example, service production needs closer collaboration with the customer; also customer participation in the service activity. Thus, co-creation and collaboration with customers is needed in the product service design process.

The aim of this paper is to contribute to the development of a Product Service engineering platform, partly based on existing tools and platforms. The work is related to project Manutelligence (“Product Service Design and Manufacturing Intelligence Engineering Platform”, H2020 636951), which aims at providing European manufacturing companies a collaborative Product/Service Design and Engineering Platform. This platform will enable designers and engineers to access data from both the traditional enterprise IT systems and IoT enabled systems for physical products information and knowledge management, utilizing natural 3D experiences. The objective of Manutelligence platform is to manage manufacturing intelligence; all data, information and knowledge related to the PS and its lifecycle. The user-centric perspective is important, both in terms of greater attention to the customer requirements (use and goal oriented approaches) and an easier involvement of customers in the PS co-design. Using the platform, it will be possible to design a product and the related service from the first life-cycle stage onward so it acts as an enabler for multiple services on top. Manufacturing companies will be able also to design and develop new innovative services based on their existing products. The product usage information and customer’s wishes can act as a basis to develop new products and tailored services.

2. Approach: importance of end user requirements

As a starting point for development, Manutelligence has two existing platforms and some existing analysis tools. These will be consolidated, complemented and adapted to become a Manutelligence platform. The development is guided by the needs of participating industrial pilots; thus requirement identification, organization and analysis is an important phase in the project. Additionally, requirements from LCA/LCC viewpoint are taken into account.

The Manutelligence project includes four industrial pilots from different industrial fields (automotive, ship, construction, 3D-printing). These were the sources for industrial requirements in the project. The requirement engineering process was carried out through the following phases: requirement elicitation, structuring & organization, refinement and validation [Manutelligence 2015a]. In elicitation, techniques like questionnaires, user stories, interviews and use case to-be descriptions were used to identify the end user requirements. For each end user pilot, a supporting research organization was nominated to guide in using the common elicitation methodology.

All the pilots already use various engineering tools in the product design. The idea was not to collect all the potential functions that an engineering platform could cover, but to identify new needs with relation to their current tools and practices. Thus the collection of requirements from the use cases does not compose a complete set of requirements for a generic PS engineering platform. Neither was the elicitation restricted by currently available tools. Keeping the pilots as the starting point, the needs that can already be fulfilled by some platform and tool providers were not left out.

Each end user pilot defined a set of use case scenarios describing the desired to-be processes in different PS life cycle phases. These were the basis for requirement identification. As the project resources are restricted, not all wishes can be implemented in the same project. In this phase the use cases were not ready to select what should be implemented and what not. The prioritization and selection was made by the use cases later in the project during the validation. However, using all the identified scenarios for requirement elicitation gave a better view to requirements of the platform.

Even if the requirement elicitation followed the same methodology for all the pilot cases, there was no common guidance on how and on which level the requirements should be expressed. Thus to use the requirements better in further development, their structuring, organization, analysis and finally again validation with the end users was needed.

This paper is focused on the structuring phase and the observations received from it. In structuring the heterogeneous requirements coming from different sources are analysed to organize them into a common structure and to identify the main high level requirements. At the same time the traceability to the original requirements must be maintained. Therefore all requirements were given a unique identifier that follows throughout the process.

Chapter 3 describes the challenges of structuring from different requirement sources. In chapter 4 the structuring methods are studied and chapter 5 gives the main results. Chapter 6 dicusses the conclusions.

3. Challenges

Even if in the elicitation phase, there was a common methodology, there were no guidelines for writing down the requirements. Natural language was used to document the requirements. The actors in each use case (company representatives and supporting research partners) were different and the requirements were defined in parallel. As a result the requirements coming from different use cases were quite heterogeneous. The level of detail was varying a lot, from a high level need to more detailed hierarchy of functions needed for the performance of a specific task. The number of requirements from each use case varied between 18-129 requirements; as a whole about 200 requirements were identified.

All the four cases originate from different contexts, cultures, languages and product-service systems. Four different industries are present: automotive, ship, construction and 3D-printing. Different sectors bring along different terminologies and different types of products and production. The companies represent different company sizes. They have differences in their preparedness for the utilisation of information technology. All cases are from different countries: Italy, Finland/Germany, Sweden and Spain. None of the cases is in English speaking countries, thus forcing researchers and interviewees to translate the requirements and case stories into English from their native language. Using natural language is challenging in itself when combining information from different sources. ESA Guide to the

user requirements definition phase [ESA 1995] mentions that "Natural language is rich and accessible but inconsistency and ambiguity are more likely".
The main objective in requirement elicitation was to receive requirements that arise from the real needs of end users and the quality of the requirements was not in the focus. That's why when assessing the requirement quality according to a checklist adapted from [Firesmith 2003] and [UTM 2015], challenges related to some quality criteria, like clarity, verifiability and atomicity can be identified (Table 1). Thus, to structure the requirements, in addition to the listed requirements, other information, like the use case scenario etc., were needed to properly understand the need.

Table 1. Requirement qualities

Quality criteria	*Quality of Manutelligence requirements*
Correctness: Does each requirement meet all or part of an actual need of its relevant stakeholder(s)? Does the requirement specify a true need, desire, or obligation?	*All the requirements are based on end user needs and scenarios. Thus they can be considered as true needs. In most cases it is possible to understand why the requirement is needed.*
Clarity / Completeness: Is the requirement unambiguous and not confusing? Does everyone agree on the meaning of the requirement? Is each requirement self contained with no missing information?	*The requirements are not always very clear even if they are understandable for the use case actors. For outsiders they are often not easy to understand properly.*
Verifiability: Can you determine whether the system satisfies the requirement? Is it possible to define a clear, unambiguous pass or fail criterion? Is it possible to determine whether the requirement has been met through inspection, analysis, demonstration, or test?	*This question is related to the above question about clarity. If the requirement is not very clear it is hard to assess the verifiability. Some of the requirements seem possible to verify, while for others the situation is not as clear.*
Traceability: Is the requirement uniquely identified so that it can be referenced unambiguously?	*All the requirements have been given an id which is kept during the process to allow tracing back.*
Design independency: Are all requirements that impose constraints on the design, thereby limiting design options, justified? Is the requirement stated in such a way that there is more than one way that it can be satisfied?	*There were typically no constraints addressed for the requirements. Thus the requirements can be satisfied with different solutions.*
Cohesiveness/ Atomicity: Does the requirement statement define exactly one requirement? Is the requirement statement free of conjunctions (and, or, but) that could indicate multiple requirements?	*Most of the requirements are not atomic. They typically include several requirements.*

4. Approach for structuring and organization

4.1 Structuring approaches

The aim of structuring and organization in Manutelligence was to understand better the main types of end user requirements and to identify potential similarities between the different cases. Thus the selected approach was first to classify the requirements, then to allocate them to the defined structures and finally to see if it is possible somehow to unify or aggregate them into common high level requirements.
In classification in principle one or more dimensions can be used. Usually each dimension originates from the needs of the analyst and aims to provide structured information. For example, one could classify Lego bricks by colour or size. Classification by colour provides valuable information for a person responsible of pigments while the classification by size helps planning logistics. Using different dimensions leads to several overlapping classifications. For the simplicity, in this case using only one dimension is preferred.
Software engineering guides propose different dimensions of classifications, like functional or non-functional, or whether the requirement is derived from one or more high-level requirements or an emergent property [SWEBOK Guide 2014]. According to the BABOK Guide [BABOK 2009]

requirements can be classified in the following categories: Business, Stakeholder, Solution, Functional, Non-Functional and Transition Requirements. Basically the data and the objective defines the method used and the level of detail of the classification.

Structuring methods, i.e. the methods to create the structure/ classification can be divided into two different groups: (1) Quantitative methods and (2) Qualitative methods. Quantitative methods basically deal with the numbers and statistics. Thus the approach is limited for requirements which include mainly non-numerical data. In Manutelligence the data is use case specific and context dependent thus giving more or less different meanings to the same terms. This type of data requires qualitative methods and interpretation based on researchers' know-how.

One of the popular qualitative methods is the Cluster Analysis, e.g. [Kaufman and Rousseeuw 1990]. "Clustering is the division of data into groups of similar objects. Clustering can be viewed as a data modelling technique that provides for concise summaries of the data. … The applications of clustering usually deal with large datasets and data with many attributes." [Berkhin 2006] Methods used in automated clustering are e.g. case-based reasoning (CBR) and cluster analysis. For Manutelligence, Cluster Analysis is an overly "quantitative" and statistical method. It is based on the similarity or closeness in data and is not a workable method considering the heterogeneousness of the available data set.

Thematic analysis is one of the most popular qualitative methods. The idea of the Thematic analysis method is to identify and define patterns or themes in the collected data. These patterns and themes are similar to the structures of data. The data can already have clear existing structures like questionnaires' topics and questions that are guiding the respondents line of reasoning. These are thus given and can be a good start for structuring. However, the data can provide a better or different structuring to consider. Deductive and Inductive approaches can be applied here in connection with the Thematic analysis but they need to be adapted due to the characteristics of Manutelligence data and context, see e.g. [Fereday and Muir-Cochrane 2006].

Aronson [1995] presents the following six step process of Thematic analysis:

1. Collect the data.
2. Patterns (of experiences) can be listed.
3. Identify all data that relate to the already classified patterns.
4. Combine and catalogue related patterns into sub-themes.
5. Themes are identified by bringing together components or fragments of ideas or experiences, which often are meaningless when viewed alone.
6. Build a valid argument for choosing the themes.

For Manutelligence Thematic analysis was considered a good starting point. In Manutelligence data and context provide preliminary structures to build on or at least to check if the existing structures can be useful when building the final structure. Manutelligence structuring approach and method presented next is based on several data analysis and classification methods to match the nature and quality of data and Manutelligence requirements engineering process.

4.2 Manutelligence approach

To start the structuring, some preparation activities were needed. First, all the requirements were given a unique code which also tells the source of the requirement. This helps in tracing them back during the process. Secondly, some flattening was performed to remove the most detailed level and thus slightly to decrease the hierarchy.

The phases of structuring are described below and in Figure 1. The main approach to define the categorisation was qualitative; viewing the requirements both top-down and bottom-up and then integrating these approaches. The tasks are described below.

4.2.1 Task 1: Top-down approach, identify given concepts and structures

The aim was to identify the given structures available in the project. These can be found for example from the main objectives and concepts of the project and from the interviews and questionnaires. The structures can be compared to find similarities. These similarities do not have to be exactly the same but on the same dimension like for example different process phases of product-service lifecycles. The usefulness of each one can be evaluated from the requirements engineering point of view. To reduce the number of dimensions, the similar ones can be merged into one or more generic ones that best represent the remaining dimensions.

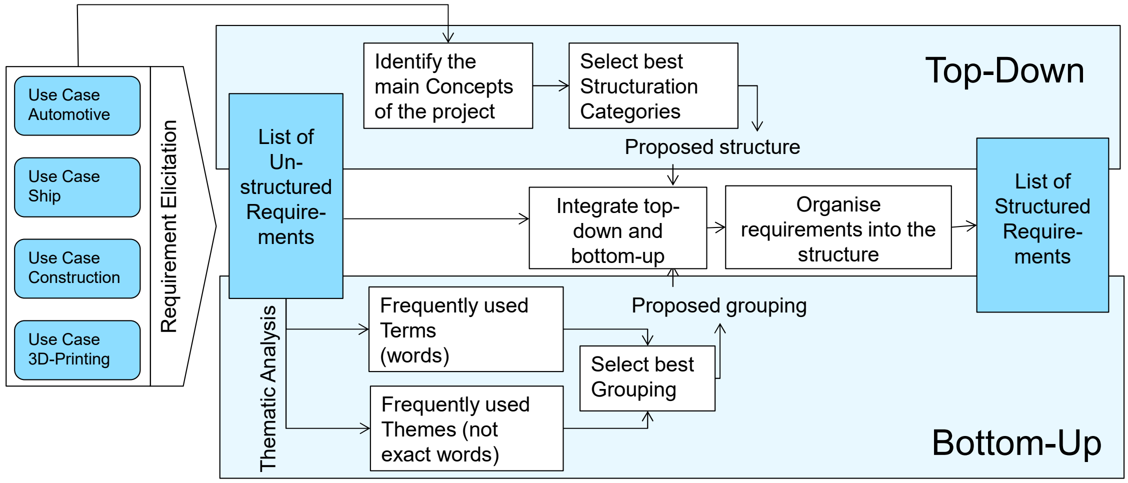

Figure 1. Requirement structuring approach

4.2.2 Task 2: Bottom-up approach, structures emerging from the data

This task is close to the Thematic analysis. Familiarizing with the data is important also in other tasks but here one has to understand not only individual requirements but also their relation to the context and case characteristics.

The bottom-up approach means analysing the unstructured requirements to identify similarities, categories and structures. The goal is to form a generic structure or hierarchy of categories that suits for all use cases and supports the development of the Manutelligence platform. In practice this means that too simple a structure is not useful since too many requirements will remain under each category but also creating too many categories will lose the idea of processing data to simplify and combine information into a better exploitable form.

4.2.3 Task 3: Integrating top-down and bottom-up

After the first two tasks the information available both from the given structures (top-down) and from the list of unstructured requirements (bottom-up) has been processed. By comparing the information now available, relations, similarities and differences are identified. The final structure can be formed, based on understanding the knowledge from the both approaches and the complete data.

4.2.4 Task 4: Organize requirements to structures

In this task the created structure is in a way tested by placing the original unstructured requirements in the structure. However, it is practically impossible to create a structure where all requirements can be placed without uncertainty. It is also possible that a single requirement can be placed in more than one category.

If the placement of requirements is difficult or if several requirements are put into more than one category the structuring should be revised. One possibility is to create new categories or subcategories if needed but they have to be justified. If it becomes clear that the structuring is not working as intended and the dimensions selected do not fit the data then one should consider starting the structuring process from the beginning.

4.2.5 Task 5: Aggregate requirements

To reduce the amount of requirements to be handled and to understand the main needs, an aggregation of similar requirements can be done. Some of the requirements under one category can practically mean the same thing. In such case it is practical for the further analysis to aggregate these into a new requirement that best represents the meaning of the aggregated requirements. It is important to maintain the integrity of data by not removing the links to the original requirements being aggregated.

5. Results

5.1 Definition of structures

The structuring or categorization of the requirements was performed by reconciling the results of top-down and bottom-up approaches. The top-down started from the structures and concepts given by the project and the bottom-up from the identified unstructured requirements.

5.1.1 Top-down approach, given concepts and structures

Manutelligence project is focused on product-service design using manufacturing intelligence and development of a platform to support the whole product service lifecycle. Thus the main concepts of the project include: product, service, lifecycle and the platform. The unstructured requirements are directed towards the Manutelligence platform.

The requirement elicitation process included two questionnaires for the use cases: a pre-elicitation questionnaire and a questionnaire about the industrial practices in product and service development and data management. The pre-elicitation questionnaire asked about the use case view to requirements, about the current product-service provision and the expected benefits. The second questionnaire was more extensive, with the main sections about design process, managing knowledge, product-service offering and evaluating the PS lifecycle.

Thus, from Manutelligence context the following main concepts can be identified: Product service (PS) (answering to question WHAT), PS Lifecycle (WHEN), PS actors / stakeholders (BY WHOM), PS related knowledge/ information / data and Platform (HOW; this is for what the requirements are).

Based on these, from top-down there are several alternatives for requirement categorization, for example based on product service type, information type, stakeholders etc. Product service –type of categorization would not support the integration of requirements of different use cases. Classification according to the stakeholders would be difficult as in most cases the objective is to support the information sharing, communication and collaboration between different stakeholders in all tasks. The division according to information type cannot be strict as many of the requirements consider different kinds of information. Especially there is a need to be able to handle and link them together. Thus performing the categorisation using the information type is not useful when developing support for manufacturing intelligence.

Thus, it seems that the most suitable candidate for the top-down structure which is significant for all the use cases is based on the lifecycle phases.

5.1.2 Bottom-up approach

The bottom-up approach means analysing the requirements coming from different sources to identify a structure, which suits for all use cases, assists in the integration of their requirements to identify the main needs and supports the development of the platform. Additionally it needs to be understandable.

To identify the important topics bottom-up, thematic analysis was applied. Two practical methods were used:

- Calculation of specific relevant terms (words) from the requirement collection to identify subjects that have high interest. About 50 words were searched.
- Definition of a group of terms / themes based on the requirements (more than one word) and analysing their occurrence. 17 themes were searched.

The analysis identified some terms that could be expected to be present in many requirements, like; design, model, data, product/ production, view and access (Table 2). However, there are also terms with high frequency that are not as expected, like feedback, customer, and link. On the other hand, some terms like service and lifecycle were not as high as could be expected.

Three of the four use cases already had an internal grouping of their requirements. The grouping was different for different use cases, but they mainly were focused on a specific life cycle phase. Different life cycle phases seemed to have different weighting; as a whole the engineering/ design / product development phase had most requirements.

Table 2. The 10 most frequent terms in the unstructured requirements

Term	Occurrence	Term	Occurrence
design	91	customer	44
model	70	production	32
feedback	53	view	30
data	50	3D	30
product	47	access	26

5.1.3 Integrating top-down and bottom-up

The selected approach for structuring top-down was based on product service life cycle. The main intrinsic grouping in the use cases also followed the life cycle approach. In the thematic analysis some of the words and themes identified are clearly related to one specific lifecycle phase and some are related to more than one phase. The thematic analysis also revealed different types of functions, especially related to the design phase. Design phase is not only of engineering/ design but also preparing and using the designed product model for intelligent actions and services in different life cycle phases. Thus, when selecting the high level structure for the integrated requirements, it was seen useful to divide the design phase functions to two groups; those related to the real design phase (creation of the product model) and those using the product model for linking additional information and documents, to perform analysis & checking and to prepare for life cycle services.

The defined five high level requirement categories are described below:

1. Product & service design into model: This includes all the requirements related to Product development/ design/ engineering – building the 3D model for the product, for example: product requirement management, product configuration, design based on construction method, creation and conversion of design for 3D printers, design changes and version management.
2. Model checking and linking: This includes checking and analysing the product using the 3D model and linking information / feedback to it, for example: tests, analysis, simulation -> data into the platform, cost calculation, LCA/LCC-analysis, sustainability analysis, customer feedback through 3D and gaming experience.
3. Serving production through model: This includes using the 3D-model to support manufacturing/ construction /installation, for example installation support, inspection support, production feedback, project planning & management, developing the production cycle.
4. Model for operation & user services: This includes all the requirements for the operation & usage phase, like measurement with sensors and IoT, operation feedback, monitoring, which use the product model and related information.
5. Sharing & non-functional requirements: This includes all non-functional requirements, also related to sharing and access to information, like- access to the product model and needed information through it, all information embedded and managed in the platform, sharing 3D-models and security aspects.

5.2 Organization of requirements to the structure

After defining the high level structure, the structure was validated by organizing all the single requirements to this common structure. A MS Excel spreadsheet tool was used to support handling the requirements. Mainly it was easy to place the requirements to the structure but in some cases a requirement was set in two different groups. This had mainly two reasons: 1. Some requirements included in fact more than one requirement in the same sentence. 2. Some requirements were not completely clear and interpretation was needed.

Table 3 shows the number of requirements placed in each category. Design phase seems to have the most interest and especially there are expectations for using the product model as the basis for different design checks and the management of different types of information, also enabling services in subsequent life cycle phases. The product operation phase does not have many requirements; on the other hand there has been some increasing interest for this phase in the use cases after the requirement elicitation.

Table 3. Number of requirements in each category

Category	Number
1. Product & service design into model	44
2. Model checking and linking	58
3. Serving production through model	41
4. Model for operation & user services	9
5. Sharing & non-functional requirements	33

5.3 Aggregation of requirements

Finally, the requirements of each category were classified in substructures, to create unified requirements. The aim was to be able to express the aggregated requirements according to the same requirement structure. The structured format proposed by [Fiordi 2012] was used in a simplified version: "The <stakeholder type> shall be able to <capability>within <performance> of <event> while <operational condition>." Thus in this phase, all the requirements were expressed with the same level of importance ("shall"). The prioritization was performed later in the project.
As a result of the aggregation all the ~200 requirements were aggregated into 5 +4 +4 + 2 + 5 = 20 requirements. The aggregated and unified requirements are listed in [Manutelligence 2015b].
Some observations about the end user needs in each requirement category below (these are not direct requirements):

1. Product service design: Moving towards the system engineering [INCOSE 2007] approach is emphasized: more systematic management of PS requirements and of PS logical structure is required. Support for using existing design experience, rules and design methods could make the design more intelligent. Collaboration between different stakeholders and interoperability between different systems is needed.
2. Model check & link: The engineering platform shall support and keep together all the analysis (lifecycle assessment / lifecycle cost assessment) and test results. The visual product model needs to be available for different stakeholders and it could be used to understand better the design result. It could also act as an interface for all the information, data and documents. Also feedback from customers could be given using the visual product model. This opens up potential for new services for the customer during the design.
3. Serving production: The engineering platform and the product model can be utilized to support the production, installation and checks and to monitor the activities.
4. Model for operation & user services: The product model can support offering services to the customers, monitoring the product e.g. with sensors and IoT and managing the service production.
5. Non-functional and sharing: Applicability for different industrial sectors, management of data quality and security, user management and support for sharing and communication came out here. These are all quite ordinary requirements but the end users seem to have shortages here also.

6. Conclusion

The focus of this paper is to support the development of a Product Service engineering platform in the context of a specific European project. The paper describes the approach and main results used to take the unstructured requirements collected in the project (Manutelligence) as an input and to organize and aggregate them in a common structure with unified manner of presentation. The process and methods

used are described. This transformation caused the about 200 unstructured requirements to be organized in five categories and aggregated as a whole to 20 requirements. Tracing back to original requirements has been preserved and no prioritization has been performed in this phase.
The structuring process showed that even if the companies involved come from different industrial contexts and their products are quite different in size and type (mass, configurable, one-of-a-kind products), they have overlapping or similar requirements for the PS engineering platform. Some main conclusions about the needs:

- The product model visualized as 3D, virtual walk or even gaming experience could be the main body for all the information relating to the product and related services. For the platform this means that all the information, documents and data should be linked easily to this model and usable through it (for example quality test results, cost and sustainability analysis etc.).
- The platform should support collaboration not only in the engineering team, but also towards the customer for early feedback. In some fields also communities of actors should be supported.
- The product information may be used to develop new services to customers based on the information. The information can be used for example for monitoring the product behaviour, maintenance, upgrades and managing the services.

In the next phases the requirements were further analysed and prioritized and finally validated with the end users before using them in the implementation. The validation caused some changes and additions to the requirement list but did not change the requirement structure. Additionally the combined product-service lifecycle management is further studied and methods and tools to support the co-creation of product-services with customers are developed. The aim is to achieve Manufacturing Intelligence through the product-service engineering environment.

Acknowledgement
This work has been partly supported by H2020 project Manutelligence (636951).

References

Aronson, J., "A Pragmatic View of Thematic Analysis", The Qualitative Report, Vol.2, No.1, 1995, pp. 1-3.

BABOK Guide, "A Guide to the Business Analysis Body of Knowledge", Version 2.0, International Institute of Business Analysis, Toronto, Ontario, Canada, 2009.

Berkhin, P., "A Survey of Clustering Data Mining Techniques", Grouping Multidimensional Data, Recent Advances in Clustering, Kogan, J., Nicholas, C., Teboulle, M. (eds.), New York, Springer, 2006, pp. 25–71.

ESA PSS-05-01, "Guide to the user requirements definition phase", Issue 1 Revision 1, ESA Publications Division, ESTEC, Noordwijk, The Netherlands, 1995.

Fereday, J., Muir-Cochrane, E., "Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development", International Journal of Qualitative Methods, 2006, pp. 80-92.

Fiordi, P., "Proposal of a Requirements Process Model for complex IT development projects", Politecnico di Milano, Facoltà di Ingegneria dei Sistemi, Corso di Laurea Specialistica in Ingegneria Gestionale, 2012.

Firesmith, D., "Specifying Good Requirements", Journal of Object Technology, Vol.2, No.4, 2003, pp. 77-87.

INCOSE International Council on Systems Engineering, "INCOSE Systems Engineering Vision 2020", INCOSE-TP-2004-004-02 September, 2007.

Kaufman, L., Rousseeuw, P. J., "Finding Groups in Data: An Introduction to Cluster Analysis", Wiley, New York, 1990.

Manutelligence Deliverable, "D1.1 Elicitation of engineering and business requirements", H2020 project no 636951, <http://www.manutelligence.eu/>, 2015a.

Manutelligence Deliverable, "D1.4 Structuration and organization of requirements", H2020 project no 636951, <http://www.manutelligence.eu/>, 2015b.

SWEBOK V3.0, "Guide to the Software Engineering Body of Knowledge", 3. edition, IEEE Computer Society Press, 2014.

Thoben, K. D., Eschenbächer, J., "Emerging concepts in E-business and Extended Products", E-Business Applications, Gasós, J., Thoben, K. D., Springer-Verlag, 2003.

Tukker, A., Tischner, U. (eds), "New business for old Europe. Product-service development as a means to enhance competitiveness and eco-efficiency", Final report of Suspronet, 2004.

UTM, "Checklist: Qualities of Good Requirements", Technological University of the Mixteca, <http://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/checklists/good_requirements_594ACCBD.html >, 2015, [Accessed 20.10.15].

Iris Karvonen, Principal scientist
VTT, Smart industry and energy systems
P.O.Box 2000, 02044 VTT, Finland
Email: iris.karvonen@vtt.fi