Chapter 2
Engineering and Business Requirements Definition, Analysis and Validation

Iris Karvonen, Tapani Ryynänen, Heidi Korhonen, Matteo Cocco and Donatella Corti

Abstract The objective of Manutelligence platform is to manage manufacturing intelligence; all data, information and knowledge related to the Product Service (PS) and its lifecycle. The platform is based on two existing platforms and some analysis tools (for example LCA and LCC). It was developed according to the needs of four use cases in different industrial fields (automotive, ship, smart house, 3D-printing). The chapter describes the four-phase methodology to define the common aggregated requirements for the platform. The phases include requirement elicitation, structuration and organization, analysis and refinement and validation. In the elicitation phase the requirements were identified from the use cases, in the structuration and refinement phases they were further consolidated, categorized and processed towards aggregated requirements and in the validation phase the resulting aggregated requirements were compared to the original use case requirements. The chapter also shows the main results of each phase.

I. Karvonen (✉) · T. Ryynänen · H. Korhonen
VTT Technical Research Centre of Finland Ltd., P.O. Box 1000, 02044 Espoo, Finland
e-mail: Iris.Karvonen@vtt.fi

T. Ryynänen
e-mail: Tapani.Ryynanen@vtt.fi

H. Korhonen
e-mail: Heidi.Korhonen@vtt.fi

M. Cocco
Dassault Systèmes, Milano, Italy
e-mail: Matteo.Cocco@3ds.com

D. Corti
SUPSI—University of Applied Sciences and Arts of Southern Switzerland, Via Cantonale, 2C, 6928 Manno, Switzerland
e-mail: Donatella.Corti@supsi.ch

L. Cattaneo and S. Terzi (eds.), *Models, Methods and Tools for Product Service Design*, PoliMI SpringerBriefs, https://doi.org/10.1007/978-3-319-95849-1_2

2.1 Introduction

This chapter describes the methodology and main results of the definition and processing of engineering and business requirements for Manutelligence platform. The chapter is focused on the phase before platform implementation; thus also the validation here is about validation of final requirements against the use case scenarios and requirements. The platform validation against the requirements is not discussed here.

As a starting point for development, Manutelligence had two existing platforms and some existing analysis tools. These have been consolidated, complemented and adapted to become the Manutelligence platform. Thus the Manutelligence approach was different from the basic software requirements definition, which often starts from the scratch (new application or module) or has the description or original user requirements of the existing platform available.

The development has been guided by the needs of participating industrial pilots. The Manutelligence project included four industrial pilots from different industrial fields (automotive, ship, smart house, 3D-printing). These cases were the sources for industrial requirements in the project. 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 collected requirements from the use cases do not compose a complete set of requirements for a generic PS engineering platform.

The requirement engineering process was carried out using a common methodology through the following phases: **requirement elicitation, structuring and organization, refinement and validation**. The intermediate requirement processing and consolidation phases were needed because the different pilot scenarios were focused on different processes and industries, with various stakeholders and user needs, which resulted in a heterogeneous set of elicited requirements, difficult to use as such for the platform development. In the process attention was given to keep the traceability.

In the requirement elicitation phase, the idea was to identify requirements with a wide scope, not restricting in what could be implemented in the current project. On one hand the wider scope gave more input for the platform development, on the other hand the pilots were in the elicitation phase not yet able to make the decision about what will be implemented in Manutelligence. Thus it was clear from the beginning that not all wishes in original requirements are implemented in this project with the restricted project resources. Thus the requirements should not be considered as static and final but more as an iterative and evolutionary set of needs.

2.2 Challenges

The main objective in requirement elicitation was to receive requirements that arise from the real needs of end users and the focus was not on the formal quality. The end users were not specialists in requirement engineering, but in PS design and engineering methods and tools. Requirement identification is often challenging, as the end users are not able to express their needs directly. Instead they need to be dragged out using different methods, taking into account the end-user business objectives. Thus user friendly methods were needed. The approach generated a set of heterogeneous requirements, which required further analysis and processing.

The sources of heterogeneity came already from different concepts and terminologies used in different sectors, but also from different groups of stakeholders, PS systems and different engineering processes and practices. Additionally, the pilot companies represent different company sizes and have differences in their preparedness for the utilization of information technology. The different groups also produced requirements with different levels of detail.

Given the above, the four datasets received were challenging to structure and consolidate. Therefore the structuration and analysis of the requirements required manual and iterative processing of data. As the structuration and analysis phases were mainly performed by researchers using different methods, the end users were again in the main role in the validation phase to check that the consolidated requirements were sufficient compared to the original pilot scenarios.

2.3 Methodology

2.3.1 Four Phase Approach

The Manutelligence approach was to integrate and adapt existing technologies to fulfill the development needs of the four pilot cases in selected PS engineering process parts. The approach affected the Manutelligence methodology for requirements engineering.

The selected approach was to apply a four-phase methodology with the phases: Elicitation, Structuration, Analysis and Refinement and Validation. The objectives of the four Requirements Engineering phases were:

Elicitation. During this process heterogeneous needs and opportunities coming from different stakeholders involved in the PS development were identified from the pilots.
Structuration. The main objective of the structuration was to unify and integrate the information collected in the previous step from disparate sources and organize them into a common structure that can be used for analysis.
Refinement and Analysis. The target of this activity was to refine and verify the previously elicited requirements. The refinement consists of the assessment of the completeness, coherence and feasibility of the stakeholders' requirements and their prioritization according to different criteria.
Requirements validation. The purpose of the validation was to ensure that the structured and consolidated requirements were sufficient for the end users (pilots) and could fulfill the defined scenarios. Thus this phase was the validation of the

consolidated requirements against the pilot needs (scenarios and stories), not the validation of the implementation. Later in the project a validation of the platform against the consolidated requirements was performed. This platform validation is out of the scope of this chapter.

2.3.2 Requirements Elicitation Techniques in Manutelligence

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 [3]. The elicitation can be performed using different methodologies such as interview, questionnaire, observation, brainstorming, prototyping, mind-map and checklist. In this phase, human activity is fundamental and it is necessary to identify users involved in the process and establish a relation between them and the developers [2].

The elicitation was started with a **pre-elicitation** phase to identify the context in which the Manutelligence project will be developed. In the pre-elicitation, information was collected using a short questionnaire about the understanding of the holistic PS and what are the stakeholder expectations from the project. It was a kind of "close interview" technique in which the stakeholders answered to a predefined set of open-ended questions (3 questions).

After this preparation phase the actual **elicitation** was carried out. The following elicitation techniques were used: questionnaire, process mapping, pilot stories and pilot scenarios.

In Manutelligence a comprehensive questionnaire with about 30 questions was used to investigate the industrial practices in product and service (PS) development and data management in the four pilots. The questionnaire included the following parts:

Part 1. Design process at glance.
Part 2. Managing knowledge in a design and development context.
Part 3. Managing the development of the PS.
Part 4. Evaluating the lifecycle of the PS.

In the process mapping activity, the PS lifecycle of each pilot was modeled to understand the main life cycle phases and their interaction and the focus of process developments needed. Because of the different levels of complexity in the pilot cases the resulting models varied in the level of detail.

The pilot story is a customer and user centric methodology, useful to understand the whole domain of the project. A pilot story basically is a storytelling with a description of how the user would interact with the Manutelligence platform rather than how it works internally or how it is designed. Telling the story, the end user is able to present the desired future operations. Going through the story, it was possible to identify requirements enabling the story to come true.

In parallel with the requirement identification, pilot scenarios for the Manutelligence project were described. The scenarios described the candidate as-is and to-be use cases to be implemented in the project pilots, offering information in a more structured format: purpose and objectives, actors involved, systems etc. This information was also used in the elicitation of the requirements.

2.3.3 Structuration Methods

The objective of the structuration phase was to organize the requirements coming from different sources into a common structure and to consolidate them to a moderate number of requirements. Thus firstly the structure had to be defined, then all the requirements were allocated to the structure and finally they were aggregated. In the beginning, each requirement was given a unique identifier that also connects it to the original pilot. This identifier followed the requirement throughout the process so that the original requirement could always be traced back.

In the structuration two approaches were integrated: top-down and bottom-up. In the **top-down** approach, the concepts and structures given by the project were identified. These could be found for example in the interviews or questionnaires. The structures were compared to find similarities, which did not have to be exactly the same but on the same dimension, like for example different process phases of product-service lifecycles.

In the **bottom-up** approach, the structures emerging from the data were identified. The task utilized an adaptation of the Thematic analysis method [1]. An understanding of the data (original pilot requirements) was required in the task, often leading to necessity to familiarize oneself with the pilot stories and scenarios.

The bottom-up approach thus meant analyzing the unstructured requirements to identify similarities, categories and structures. The goal was to form a generic structure or hierarchy of categories that suits for all the use cases and supports the development of the Manutelligence platform.

In the next phase the information available from both the given structures (top-down) and from the list of unstructured requirements (bottom-up) was analyzed and relations, similarities and differences were identified. The final structure was formed, based on understanding the knowledge from both approaches and the complete data.

The pilot requirements were organized to the defined structure. The organization also tested if the structure was sufficient, if it was possible to put each requirement somewhere in the structure.

Finally the original pilot requirements belonging to the same subcategory were **aggregated**. The aggregated requirements are not as detailed as the original ones but they aim to integrate similar needs from different pilots. The links to original requirements were maintained.

2.3.4 Analysis and Prioritization Method

The objective of the third phase was to further refine and prioritize the structured requirements coming from the previous phase. The aim of the prioritization was not to remove any requirements but to create an overall view of their high level importance. The final decision of the requirements to be implemented during the project was taken along the pilot development.

The requirements were first reviewed in order to make the level of detail more homogeneous and to eliminate potential duplications. Next a trade-off analysis was performed to identify on one hand mutually supportive and on the other hand conflicting requirements. In the trade-off analysis each couple of requirements was considered and the corresponding relationship was qualitatively evaluated. The correlation was analyzed considering the mutual impact of requirements during the development of the platform. A positive correlation means that the parallel fulfillment of the two requirements is mutually supportive and vice-versa. Values ranging from −2 to +2 were used.

For the prioritization two types of criteria were defined: (1) Manutelligence-related criteria and (2) Pilot-related criteria. Manutelligence-related criteria come from understanding the general objectives of the project. Aggregated requirements were used in this phase. Pilot-related criteria are based on the needs of the pilot cases; thus the original unstructured requirements were used here. These requirements were considered on how much they can positively impact on the design process of the PS in the pilot.

Findings coming from the two prioritization analyses were finally merged to form the final rank. A bonus system was used that favors more those requirements that are addressed as important by both the Manutelligence-related and the pilot-related criteria. This final rank achieved provided evidence about what are the most relevant requirements to be fulfilled within the Manutelligence project since it summarized all the previous analyses based on different points of view.

2.3.5 Requirements Validation Method

Validation has different roles over the application development process. In Manutelligence the first validation took place in the requirement definition phase and it was about **validation of requirements**, not software. Thus the objective of the requirements validation here was not to check that the Manutelligence platform and related tools fulfill the requirements, but that the aggregated requirements fulfill the end user needs. Also the prioritization defined in the previous task was checked. This was needed, as the composition, structuring, aggregation and analysis (including prioritization) of requirements from different use cases were performed by the supporting partners, not the use case owners themselves.

Different methods for the validation were applied. First an individual review using a walk-through approach was used to check the sufficiency of aggregated requirements (not the priorities). The review was performed by a group of researchers representing the partners supporting the end users in Manutelligence. In the review each of the use cases was handled separately. Two pieces of source material for each case were used: (1) the pilot stories and (2) the end user scenario descriptions (to-be). The approach in the review was first to walk through the pilot story step by step and to identify the main functionality needed for each step. Thereafter the list of needed functionalities was compared to the list of aggregated requirements to see if there is a requirement available, which enables taking the step. After that, the same was done for the end user scenarios (to-be). As the aggregated requirements are on a higher level, telling more about "what" than "how", the idea was to find a high level requirement, which could cover the lower level functionality.

It is clear that not all aggregated requirements were needed for each use case, but the other way around; at least one requirement was needed for each step. Otherwise a shortage was recorded.

To include the end users (pilots) in the validation, a specific validation workshop was organized. The workshop contained the following three main sessions:

- Presentation of the aggregated requirements,
- Industrial partners crosschecking the Use Case requirements,
- Industrial partners checking the prioritization of the requirements.

The main task was the crosschecking of the aggregated requirements by the industrial partners. The participants were divided into sub-groups, one for each pilot and one for software developers, five groups in all. The methodology used was a form of Requirements Walk-Through and Reading Technique. The participants were asked to review the partner specific Pilot Stories and Use Case scenario descriptions (to-be) to check the sufficiency of the aggregated requirements. The groups were equipped with printouts, in A3 size, of Pilot Stories, Use Case scenarios and the list of aggregated requirements. Figure 2.1 depicts the methodology.

The participants read through their Pilot Story and Use Case scenarios, section by section. For each encountered step or function in the text, the participant checked that a corresponding requirement could be found in the list of aggregated requirements. These were marked with a circled 1, 2 and 3 etc. as seen in Fig. 2.2. If an aggregated requirement covering the issue could not be found, then a note was made. The number of how many times an aggregated requirement was referenced to was counted for each Industrial partner.

The third and final step in the workshop for each Industrial partner was to point out the most important aggregated requirements. Each industrial partner was asked to mark the five top important aggregated requirements for its specific use cases. The given rankings were summarized into an overall ranking.

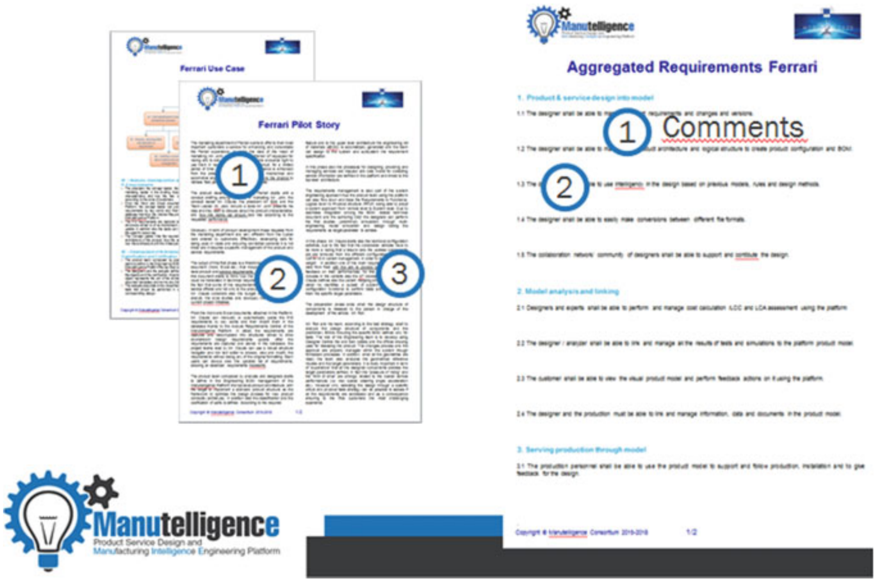

Ferrari Use Case
Ferrari Pilot Story
1
2
3
Aggregated Requirements Ferrari
1. Product & service design into model
1
Comments
2
2. Model analysis and linking
3. Serving production through model
Manutelligence
Product Service Design and Manufacturing Intelligence Engineering Platform

Fig. 2.1 Reading technique used in the workshop

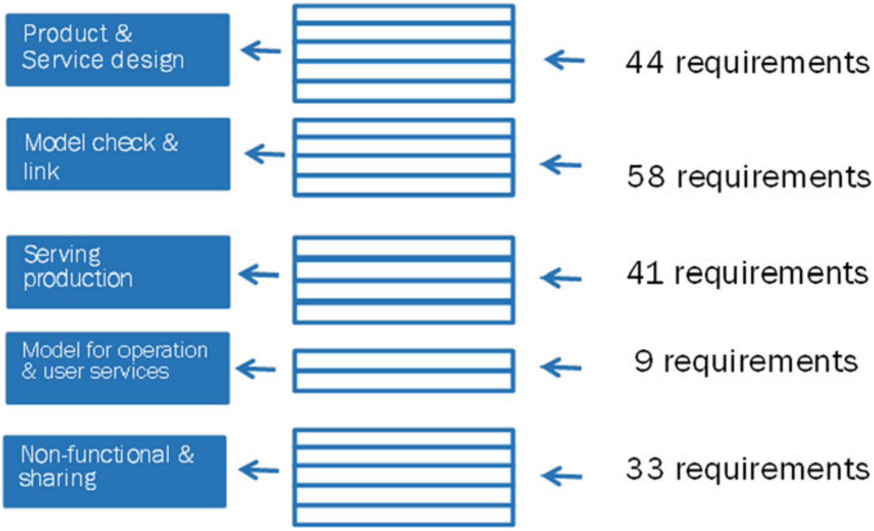


Fig. 2.2 Requirement categories and number of unstructured requirements in each category

2.4 Results from the Definition of Business and Engineering Requirements

2.4.1 Results from Requirements Elicitation

The requirement elicitation generated more than 200 requirements coming from the four industrial pilots (automotive 23, ship 129, smart house 25, 3D-printing 18) and from LCA/LCC technical workpackages (9). The number of requirements coming from one use case (ship) was much higher than from other use cases. This was due to using a requirement hierarchy and more detailed low level requirements. As expected, the requirements were quite heterogeneous and focusing on different process parts in the PS engineering.

2.4.2 Results from Structuration and Organization

As described earlier, the structuring and categorization of requirements were performed by reconciling the results of top-down and bottom-up approaches.

For the top-down approach, concepts coming from the project were studied. Manutelligence project is focused on Product-Service design using manufacturing intelligence and through the development of a platform to support the whole Product Service lifecycle. Thus, from Manutelligence context the following main concepts could be identified:

- Product service (PS) (answering to question WHAT).
- PS Lifecycle (WHEN).
- PS actors/stakeholders (BY WHOM).
- PS related knowledge/information/data.
- Platform (HOW; this is for what the requirements are).

Based on these, from top-down there were 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, it seemed that the most suitable candidate for the top-down structure, which was significant for all the use cases, is based on the lifecycle phases.

In the bottom-up approach the requirements coming from different sources were analyzed to identify a structure, which could suit for all the use cases and assist in the aggregation of their requirements. Additionally it should be understandable.

Table 2.1 The top 12 words and terms in the unstructured requirements

Word	Occurrence	Theme	Occurrence
Design	91	3D-viewing	106
Model	70	Feedback management	104
Feedback	53	Production planning and control	66
Data	50	Platform content	55
Product	47	Product Configuration, BOM	49
Customer	44	Link to drawings	36
Production	32	Customer interaction	33
View	30	Validation and Verification, inspection	23
3D	30	Design	23
Access	26	Training, Guidance	21
Management	22	LCA/LCC, Environmental issues	14
Link	19	Change management	11

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

- Calculation of specific relevant 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 analyzing their occurrence. 17 terms were searched.

Table 2.1 shows the top 12 occurred words and themes. The number of occurrences for each word is affected by the heterogeneity of the requirements. This is mainly because the different use cases have given their requirements on different levels of detail, using more or less words. Thus the requirements including longer and more detailed expressions have more impact on this analysis.

The analysis identified as frequent some words that could be expected to be present in many requirements, like design, model, data product/production and view and access. However, there were also words with high frequency that were not as expected, like feedback, customer, and link. On the other hand, some terms like service and lifecycle did not belong to the top group.

On the right side of Table 2.1 again the top 12 occurrences of themes (not exact words) are presented. The themes seem to be in line with the identified words.

Structuring based on Product Service life cycle seemed to be most suitable in the top-down approach. 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 were clearly related to one specific lifecycle phase and some were 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 information/model for intelligent actions 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 design/model) and those using the product model for linking additional information/documents, to perform analysis and checking and to prepare for life cycle services.

The defined five high level requirement categories are described below. The origin of the included requirements is shown with the codes: A-Automotive, S-Ship, C-Construction/Smart House, F-3D printing, LCA.

Product and 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 (F), product configuration (F+C), design based on construction method (S), creation and conversion of design for 3D printers (F), design changes and version management (A, C).

Model checking and linking
This includes checking and analyzing the product using the 3D model and linking information/feedback to it, for example: tests, analysis, simulation, data into the platform (A), cost calculation (C), LCA/LCC-analysis (LCA), sustainability analysis (F), customer feedback through 3D and gaming experience (S).

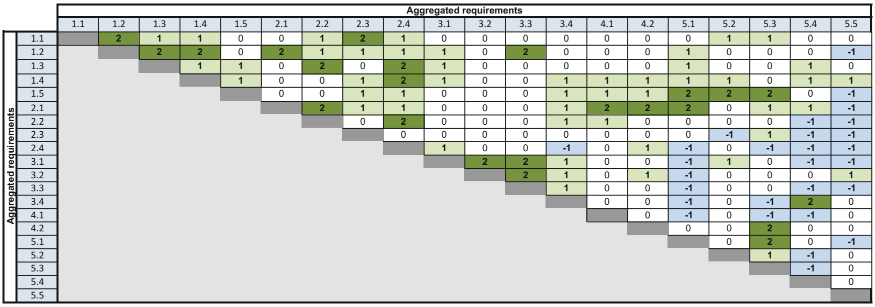
Serving production through model
This includes using the 3D-model to support manufacturing/construction/installation, for example: installation support (S), inspection support (S), production feedback (S), project planning and management (C), developing the production cycle (F).

Model for operation and user services
This includes all the requirements for the operation and usage phase, like measurement with sensors and IoT (C), operation feedback (S), monitoring (S) which use the product model and related information.

Sharing and 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 (S), all information embedded and managed in the platform (A), sharing Fablab-models (F) and security aspects.

Categories 1–3 belong to the Beginning of Life and 4 to Middle of Life. We identified no requirements for the End of Life.

After defining the high level structure, it was validated by organizing all the single requirements to this common structure. Mainly it was easy to place the requirements to the structure but in some cases a requirement was set in two different groups. There were mainly two reasons for this: 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.



		Aggregated requirements																			
		1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4.1	4.2	5.1	5.2	5.3	5.4	5.5
Aggregated requirements	1.1		2	1	1	0	0	1	2	1	0	0	0	0	0	0	0	1	1	0	0
	1.2			2	2	0	2	1	1	1	1	0	2	0	0	0	1	0	0	0	-1
	1.3				1	1	0	2	0	2	1	0	0	0	0	0	1	0	0	1	0
	1.4					1	0	0	1	2	1	0	0	1	1	1	1	1	0	1	1
	1.5						0	0	1	1	0	0	0	1	1	1	2	2	2	0	-1
	2.1							2	1	1	0	0	0	1	2	2	2	0	1	1	-1
	2.2								0	2	0	0	0	1	1	0	0	0	0	-1	-1
	2.3									0	0	0	0	0	0	0	0	-1	1	-1	-1
	2.4										1	0	0	-1	0	1	-1	0	-1	-1	-1
	3.1											2	2	1	0	0	-1	1	0	-1	-1
	3.2												2	1	0	1	-1	0	0	0	1
	3.3													1	0	0	-1	0	0	-1	-1
	3.4														0	0	-1	0	-1	2	0
	4.1															0	-1	0	-1	-1	0
	4.2																0	0	2	0	0
	5.1																	0	2	0	-1
	5.2																		1	-1	0
	5.3																			-1	0
	5.4																				0
	5.5																				

Fig. 2.3 Trade-off among macro-requirements. The scale used is the following: 2: strong positive correlation; 1: positive correlation; 0: no correlation; −1: negative correlation

As a result of the aggregation all the ~200 requirements were aggregated into 5 + 4 + 4 + 2 + 5 = 20 requirements (Fig. 2.2).

2.4.3 Results from Prioritization

The trade-off analysis (correlation between the requirements) was performed by the research partners. The results are shown in Fig. 2.3. Each aggregated requirement has been given a number (for example 1.3), where the first number represents the requirement category.

The prioritization criteria were defined as follows: (1) Manutelligence-related prioritization criteria including Implementation time, Implementation cost, Technical gap, Usability in other sectors, Scalability, PSS fitting, and Enabling collaboration. Weights for the criteria were also defined. (2) Pilot related criteria, for example, Design time, Change management agility, Improved communication with customers and Improved communication among designers. Each pilot defined its own set of weights for the criteria, since it was expected that the relative importance of the performance associated to the criteria vary depending on the specific context.

The Manutelligence-related criteria were applied for the 20 aggregated requirements while the pilot-related for the original pilot requirements separately for each pilot. The individual scores where then summarized to the aggregated requirements.

For the final ranking additionally a bonus system was defined. The basic idea was that the score of an aggregated requirement obtained from the Manutelligence-related criteria consideration was increased by a bonus if the same requirement was considered to be important also by one or more pilots.

It should be noted that the final list and the relative rank was not frozen at this point: the overall requirements engineering process followed a spiral approach and during the development new interests also came up. Also, no requirements were eliminated from the list in this phase, even though some of them were assigned a lower score. The highest scores were received by the following requirements:

4.2 The service provider shall be able to manage the services and their data on the platform.

2.1 Designers and experts shall be able to perform and manage real time cost calculation/LCC and LCA assessment along the design using the platform and the data from design and previous projects.

1.1 The designer shall be able to systematically manage product requirements and trace design changes and versions within the platform.

1.5 The collaboration network/community of designers shall be able to support and contribute to the design on the platform.

2.5 Validation Results

In the individual walk through the pilot stories and to-be scenarios, 1–7 steps were identified for which there was no clear corresponding requirement. Mostly these needs were quite detailed or very specific. In addition there were some steps for which link to an existing requirement could be identified, but the requirement should somehow be extended to cover a specific aspect. There were two main types of comments:

- The requirement should include "service" in addition to product. This is very important as Manutelligence aims to support Product-Service design.
- In addition to other stakeholders, also customer or end user should have access to the PS information. This is relevant as the customer interaction and participation is even more important when providing PS than when providing products.

Based on these comments, the aggregated requirements were reformulated to cover also services and customers/users.

All the four use cases participated in the validation workshop. First the 20 aggregated requirements were discussed and clarified with their source: which use cases and original requirements have affected to each requirement. Most of the requirements were considered understandable but also some comments regarding them were received from the use cases. These mainly included terminology.

The validation of requirements was performed in five groups: one for each use case + one group for platform providers. The use case representatives were supported by the research partners. The methodology described before was followed: the use cases went through the pilot story and identified the aggregated requirements, which supported the pilot story activity. The links between the pilot story steps and the requirements were marked with the same number. The idea was also to identify missing requirements for supporting pilot steps.

The number of identified links for each aggregated requirement from the use cases was identified. The following was observed:

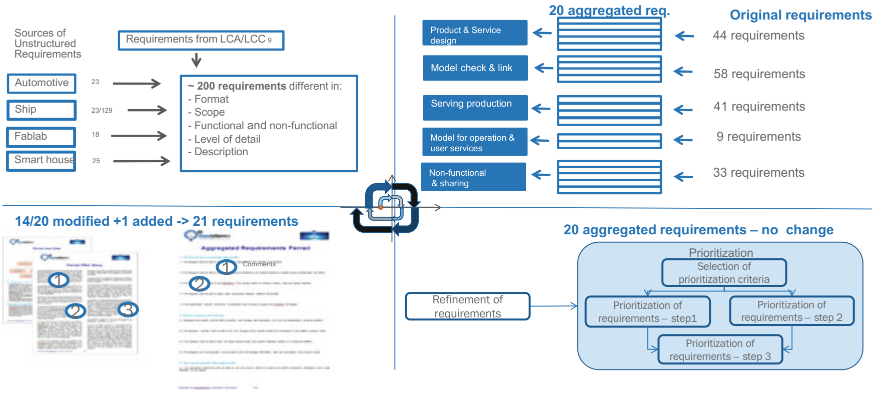

Sources of Unstructured Requirements
Requirements from LCA/LCC 9
Automotive
Ship
Fablab
Smart house
23
23/129
18
25
~ 200 requirements different in:
- Format
- Scope
- Functional and non-functional
- Level of detail
- Description
20 aggregated req.
Original requirements
Product & Service design
Model check & link
Serving production
Model for operation & user services
Non-functional & sharing
44 requirements
58 requirements
41 requirements
9 requirements
33 requirements
14/20 modified +1 added -> 21 requirements
Comments
1
2
3
20 aggregated requirements – no change
Refinement of requirements
Prioritization
Selection of prioritization criteria
Prioritization of requirements – step1
Prioritization of requirements – step 2
Prioritization of requirements – step 3

Fig. 2.4 Requirement identification, processing and validation process

- The total number of occurrences for a single requirement was between 1 and 14.
- No requirement was found unnecessary.
- 4 requirements were relevant only for one use case.
- There was only one requirement that was needed for all use cases.
- 6 requirements had a link to all but one (3) use case.

The validation revealed some missing requirements and comments for the requirement text.

Finally, the results (comments and missing requirements) of the different validation steps were collected together and the requirements were reviewed to study how they should be changed based on the comments. As a whole, 14 requirements were reformulated and 1 new requirement was added.

Furthermore, to validate the prioritization, the use cases were asked to select the 5 most important requirements from their viewpoint. The selections of use cases were quite scattered. There was no requirement that belonged to the 5 most important of all the use cases. 6 requirements were selected by 2 use cases and 6 were selected by no use case. Thus in each pilot case the decision was made what to implement in Manutelligence project, and what to leave for later.

2.6 Conclusion

A summary of the four phases of the business and engineering requirements is presented in Fig. 2.4 (starting from up-left). The process went through requirement elicitation, structuration and organization, analysis and prioritization and finally validation. In the process a large number of diverse and heterogeneous requirements were identified, organized and aggregated into a manageable set of requirements, well-structured and prioritized PS requirements.

The final list of 21 aggregated and validated requirements is the following:

1. **Product service design into model**

   1.1 The user shall be able to systematically manage product and service requirements and trace design changes and versions within the platform.
   1.2 The user shall be able to manage the product and service structure to create product configuration and BOM on the platform.
   1.3 The designer shall be able to use knowledge in the design based on previous models and the platform shall provide automatically the rules and design methods.
   1.4 The platform shall be able to make easily data conversions (CAD files, product models, visualization models, manufacturing models).
   1.5 The collaboration network/community of designers shall be able to support and contribute the design on the platform.

2. **Model analysis and linking**

   2.1 Designers, customers and other users shall be able to perform, manage and view real time LCC and LCA. Here the data from design and previous projects should be available.
   2.2 The platform shall use the product model to link, manage and allow access to all the results of (quality) tests and simulations.
   2.3 The customer shall be able to view the visual product model (including virtual walk/driving) and give feedback on it using the platform.
   2.4 The designer and the production shall be able to link and manage information, data and documents in the product model supported by platform specifications and rules.

3. **Serving production through model**

   3.1 The production personnel and the customer shall be able to use the product model on the platform to support and monitor production, installation and to give feedback for the design.
   3.2 Project manager and customer shall be able to use the model on the platform to plan, monitor and manage the contract and the production.
   3.3 The manufacturer/production coordinator/user shall be able to manage the production resources and suppliers through the platform.
   3.4 The production/quality management shall be able to manage and follow the quality and failure data on the platform.

4. **Model for operation and user services**

   4.1 The user/ service provider shall be able to monitor the behaviour of the product using the product model and linked sensors with access to the platform.
   4.2 The service provider and the customer shall be able to manage the services and their data on the platform.

5. **Non-functional requirements (including sharing information)**

    5.1 All information shall be managed and embedded in a common platform, which is applicable for different industrial sectors.
    5.2 The platform shall provide user management and allow access to stakeholders according their rights and needs.
    5.3 The platform shall support sharing and communication, also remotely/online/off-line.
    5.4 The platform shall manage data security and quality (including metadata).
    5.5 User interface of the platform services shall be easy to use and include user support, like tutorials.
    5.6 The platform should allow to display PSS-related advertisements for the customer and support the selling process for additional products and services in dependency of a PSS's life cycle phase and its actual condition (for example by visualizing them in the product context).

These aggregated requirements were used in the development phase to support the platform and pilot development. In the final phase of the project the implemented solution (pilots and the platform) was validated against the aggregated requirements.

References

1. Aronson J (1995) A pragmatic view of thematic analysis. Qual Rep 2(1):1–3. Retrieved from http://nsuworks.nova.edu/tqr/vol2/iss1/3. Accessed 13 Dec 2017
2. Sawyer P, Kotonya G (2001) Software requirements. SWEBOK, 9
3. Sommerville I, Kotonya G (1998) Requirements engineering: processes and techniques. Wiley, New York