ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

MANAGING A METHODOLOGY-DEVELOPMENT REQUIREMENT-PROCESS – BEST PRACTICES AND LESSONS LEARNED

Phillip SCHRIEVERHOFF, Udo LINDEMANN
Technische Universtät München, Germany

ABSTRACT

The goal of the EU-Project AMISA – Architecting Manufacturing Systems and Industries for Adaptability – is to develop a methodology which allows optimizing system's architecture towards maximum lifecycle value and furthermore implement it as a software tool. This presents a complex undertaking in itself and is enhanced by the complexity originating from the multi-national consortium with broad industry and academic backgrounds.

This paper focuses on the requirement-management-process towards the Design for Adaptability methodology and software tool consisting of the main steps Requirement Acquisition and Categorization, Requirement Consolidation and Grading as well as Status Control and Management of Critical Requirements. The conducted steps are outlined in terms of the procedure, results and lessons learned and best-parcitces described.

Keywords: design methodology, requirements, design to X, requirment-management-process, methodology development

Contact:
Phillip Schrieverhoff
Technische Universtiät München
Institute of Product Development
Garching
D-84748
Germany
schrieverhoff@pe.mw.tum.de

1 INTRODUCTION

The goal of the EU-Project AMISA – *Architecting Manufacturing Systems and Industries for Adaptability* – is to develop a methodology which allows optimizing system's architecture towards maximum lifecycle value. All systems and products are designed to fulfil the needs of their stakeholders. The more accurately they are able to meet those needs, the higher is their value to the stakeholder. This is not a one-point-in-time problem, but applies to the whole product life cycle. Design for Adaptability (DfA) aims at minimizing the gap between stakeholder needs and the capability of a system or product to fulfil them. (Schrieverhoff et. al., 2011)

Correspondingly, Hashemian (2005) describes adaptations as the response of a system to new service or operational requirements. Adaptations involve modifications to the internal structure of the system. There is still a lack of understanding on the concept of adaptability, how to systematically design adaptability into systems and how to quantify the degree of adaptability of a system (Kissel, Schrieverhoff and Lindemann 2012; Fletcher, Brennan and Gu, 2009). This is the reason for the implementation of AMISA. The project started in April 2011 and is conducted by a consortium of two academic and six industrial partners from five countries with the goal of developing a DfA methodology as well as software tool.

As elementary basis for methodology development the requirement-management-process received special attention within the course of action. The collection of individual goals, consolidation and alignment thereof and especially the compilation of a common "big picture" on both the conceptual as well as practical level are perceived as vital factors for project success and represent the output strived for in the process.

The requirements for the methodology and software tool were determined and concretized in several steps. Due to the size and broad scope of the consortium, transparency of the process and consensus-decisions supported by all involved stakeholders were seen as essential aspects during requirement acquisition. Challenges mainly originated from the resolution of conflicts between partners' special interests and technical feasibility, differences in understanding as well as different interests and capabilities.

This paper focuses on the process of requirement acquisition and management towards the DfA methodology and software tool and outlines the conducted steps and procedure, best practices and lessons learned. In a first step all partners documented the essential requirements from their individual perspective. Those were categorized and consolidated into one document that was made available for the whole consortium as a basis of further work. During two requirement workshops with representatives of each partner being present, the requirements were reviewed and elaborated further. Redundancies were removed; requirements were challenged, discussed and sharpened. In a final step, a consensus decision for each requirement that could not be agreed on before had to be achieved in order to take it into further account or delete it. Furthermore the compiled requirement document was used for status control of project progress and to recapitulate the achievement of project success.

2 PROJECT ENVIRONMENT AND REQUIREMENT-PROCESS-OVERVIEW

AMISA is a project funded by the European Commission in the course of the 7th Framework Program. Companies and researchers from five countries, Italy, Spain, Israel, Romania and Germany, work together to develop a methodology to design manufacturing industries and systems more adaptable to future needs. (European Commission, 2011)

The project expects to deliver a step-change in the performance of European industry characterized by a higher reactivity to customer needs and more economical production lines, product systems and customer services. More specifically, AMISA objectives are:

Objective-1: Develop a generic (widely applicable) and tailorable quantitative and usable method for architecting systems for optimal adaptability to unforeseen future changes in stakeholder needs and technology development. Such systems will exhibit better cost-efficiency, longer lifetime as well as reduced cycle time, thus, provide more value to stakeholders.

Objective-2: Validate and prove the methodology by means of real-life pilot projects in order to provide concrete evidence that it is 1) Generic and tailorable, 2) Scalable, 3) Usable and 4) Cost effective.

Objective-3: Show by the end of the project, that reconfiguring manufacturing systems or products/services designed for adaptability, yield savings either in cost or cycle time or a combination thereof.
Objective-4: Show by the end of the project, that the lifespan of manufacturing systems or products/services designed for adaptability increases.
Objective-5: Show that systems yielding more service for a longer duration will exhibit the following qualitative benefits: (1) during the manufacturing process, the overall usage of natural resources and energy consumption as well as the overall pollution and byproduct waste will be reduced (2) adaptable systems will be more amenable to sustained evolving regulatory framework (i.e. environmental, health, safety, etc.)
Those objectives are reflected from the consortium members' individual backgrounds in systems engineering, mechanical engineering, packaging solutions in food industry, manufacturing of machine tools, truck and bus, aerospace, optoelectronics and communications, leading to resulting requirements may therefore be very individual as well and have to be actively managed towards a common goal.
Within the project, the requirement-management-process can be structured into three main elements.
Requirement acquisition and categorization was followed by **requirement consolidation and grading** in the first months of the project. Thereafter **status control and management of critical requirements** took place continuously at relevant consortium meetings.

Figure 1. Requirement-Management-Process

Currently more than half of the project duration has passed and allows an initial reflective judgment of the process.

2.1 Step 1: Requirement Acquisition and Categorization

The aim of the first phase was to generate an overview of the individual partners' needs and wants towards the methodology and software under development. Each participant was asked to list the requirements essential for fulfilling the project objectives as well as assure individual project success. Subsequently, the inputs were categorized thematically by one of the academic partners. Furthermore similar requirements were clustered, which on the one hand prepared the consolidation process and on the other hand gave an indication about the weight of certain clusters according to the number of inputs.

Procedure

Step one in the procedure was the collection of input from the consortium members, for which they were given one month time after the project kick-off. A template was provided in order to assure uniform and consistent inputs. Each requirement had to be described and the rationale for its inclusion given. For reasons of documentation and traceability the source of the requirement (code identifying each partner) had to be stated and an ID assigned, which was done by ascending numbering. Furthermore the possibility was given to assign each requirement to methodology and/or software development. Figure 2 depicts the template.

Req ID			Requirement			
Source of Requirement	Requirement ID (###)	Version ID (##)	Method	Software	Requirement description	Requirement rationale

Figure 2. Scheme of Requirement Description

In total 164 Requirements were collected by the eight partners. Of those 107 referred to methodology development and 92 to software development, implying and intersection of 35 requirements. Since that

intersection was significant and many requirements related both to method and tool development, it was decided to keep the requirements to both integrated in one document and to introduce a thematic categorization according to the requirements' background.
In order to improve the overview and the workability of the list of requirements, eight categories were established into which the requirements were sorted in relation to their subject and scope. Those categories, their description as well as the number of requirements assigned to each category are shown in the following table:

Table 1. Thematic categorization of requirements

Category	Definition	Total	Method	Tool
System architecture/ -modeling	Definition of structure, behavior, and views of the system the methodology is based on	39	38	17
Input data	Data that the methodology/tool requires or has to be able to process as input	24	22	2
Output data	Data that the methodology/SW delivers as output	26	19	12
Cost calculation	Processing of monetary data	18	17	11
Computational Specifications	Software and Hardware requirements for use of the tool	24	1	24
Usability	Ease of use and learnability	22	3	22
Uncertainties/Risk	Consideration of uncertainties, risks and unknown factors within the calculation	6	4	3
Transparency of calculation	Traceability, comprehensibility and transparency of internal calculation	5	3	2

Most requirements referred to the category *System architecture/-modeling*, followed by similar numbers in *Input data*, *Output data*, *Computational Specifications* and *Usability*. *Cost calculation* follows closely after and in the areas of *Uncertainties/Risk* as well as *Transparency of calculation* significantly less requirements were named.
Table 2 gives an overview of exemplary requirements in each category. The categories were determined after the input by the partners was given and thus reflect the main areas of relevance out of the consortium point of view.

Table 2. Exemplary requirements within categories

Category	Example
System architecture/-modeling	The methodology shall be cascaded from the product design to the production chain
Input data	Method must cope with different levels of detail of input data
Output data	The methodology shall provide cost/benefit indicators for each architectural scenario under study
Cost calculation	Methodology takes different controlling instruments into consideration: target costing; total cost of ownership;…
Computational Specification	The software tool should be supported for different operating systems: Linux, MAC OS, Windows
Usability	The DFA tool shall be capable of accepting partial data allowing users to add further data as they acquire it over time
Uncertainties/Risk	The method should include a risk scale about the dependency of each component with external supplier(s)
Transparency of calculation	The reliability of the output has to be traceable

As stated before, requirements within categories exhibiting a similar objective and related content were clustered. Since each partner provided input, redundancies and overlaps occurred frequently. Nevertheless, nuances in the requirement definition and implication were often different and furthermore the number of requirements addressing a similar aspect underlined its importance.

Table 3. Exemplary cluster of related requirements

ID			Requirement							
Source of Requirement	Requirement ID (###)	Version ID (##)	Requirement Method	Requirement Software	Requirement description	Requirement rationale	Initial prioritisation (Low/Medium/ High)	Category	Degree of abstraction
TAU	020	01		x	The DFA tool shall be created so it will be simple, intuitive and friendly to operate	Although this requirement is not testable in a formal manner, it is still not impossible to verify it on an intuitive basis	High	Usability	Appropriate
MAN	034	01		x	The GUI should be clearly arranged, user friendly, intuitive, structured	For a company-wide rollout a system and GUI of a high professionality and quality is required	Medium	Usability	Appropriate
TTI	009	01		x	The software should have a friendly interface with the user.	The software should permit to see a graph/table that represents the components/subsystems/modules of the product in a interactive way.	High	Usability	Abstract

The academic partner managing the requirement acquisition process initially assigned two quality criteria to each requirement as a basis for the subsequent consolidation process. On the one hand the degree of abstraction was assessed, indicating a tendency of the requirements be to detailed/abstract or appropriate for the conceptual stage (Table 4). On the other hand the prioritization was graded (low/medium/high) indicating the importance of the requirement. As stated, the grading was initial and open to be changed by the consortium during the consolidation process.

Table 4. Exemplarily requirements in different degrees of abstraction

Requirement	Degree of abstraction
The tool shall be modeled in C++	Detailed
The method should allow to increase adaptability	Abstract
The DFA tool shall be configurable to accept different modeling equations	Appropriate

Most requirements were judged *appropriate* for the conceptual phase of development, approximately one third was judged to be too *detailed* or *abstract*.

Results and Lessons Learned

In total 164 requirements were collected by the consortium, categorized and initially graded by one of the academic partners and prepared for the consolidation and grading process in plenum.

The procedure is judged to have been exhaustive, because until mid-term of the project no significant gaps in the entity of requirements evolved.

The central processing and categorization of the input, as well as the uniform input-scheme and assured traceability of the individual contributions, led to the avoidance or quick resolve of unclarities and major iterations in the further course of the project. Nevertheless the initial judgment of the requirements out of the point of view of the responsible academic partner may have imposed a bias on the later judgment of the industry partners. Even so it was highlighted during the subsequent meetings that all categories and gradings were open to be changed, it cannot be fully excluded to have had a disproportionate influence.

Out of the point of view of the academic partner conducting the requirement consolidation process, a key factor for consolidating the requirements was to be provided not only the requirement formulation but also the according rationale. In many cases this additional information helped to resolve ambiguities and to understand the given requirement in detail.

The fact that approximately one third of the requirements was judged to be too *detailed* or *abstract* shows that the understanding of requirement formulation was not homogeneous for all of the partners.

This may have been improved by the provision of clear formulation criteria and exemplary requirements.

2.2 Step 2: Requirement consolidation and grading

The consolidation and grading of the collected requirements was conducted in two workshops during consortium meetings approximately two months apart. In the first workshop with duration of five hours the requirements were reviewed and consolidated in groups. In the second workshop requirements that were left “open”, meaning that could not be agreed on referring to prioritization and degree of abstraction, were discussed and judged in plenum.

Requirement Workshop I

The Workshop took place in course of the first Technical Meeting (TM) conducted in the project consortium two month into the project. The goal was to remove redundancies from the requirement list, sharpen the requirements and to judge the degree of abstraction as well as importance. It was communicated that the development was to start based on the requirements defined but that nevertheless the requirement-list has to be seen as a “living” document that allows for changes and further detailing during the development process. Special attention was paid to transparency, relevance for the conceptual stage and consensus within the consortium. The exact procedure as well as results and lessons learned are described in this section.

Procedure

As described the workshop was conducted during the first TM of the consortium. The agenda contained an introduction, an overview of the gathered input, two consolidation and grading sessions with an intermediate status check as well as a wrap-up session comprising a final discussion and definition of the next steps. In the following, special focus will be laid on the consolidation and grading sessions as integral part of the workshop.
The requirements had been divided into eight categories and a station setup of four stations was chosen due to group size and available facilities, thus two categories were placed at each station. This furthermore allowed for the homogeneous amount of approximately 40 requirements to be worked through at each station since large and small categories could be balanced out against each other.
The consortium was split into four groups of three to four people each. The groups were assorted in a way that members of the same partner were in different groups to provide maximally heterogeneous points of view in each one. The groups visited each station sequentially, whereas 75 minutes were given for the initial consolidation at the first station and subsequently 30 minutes for further consolidation and review.

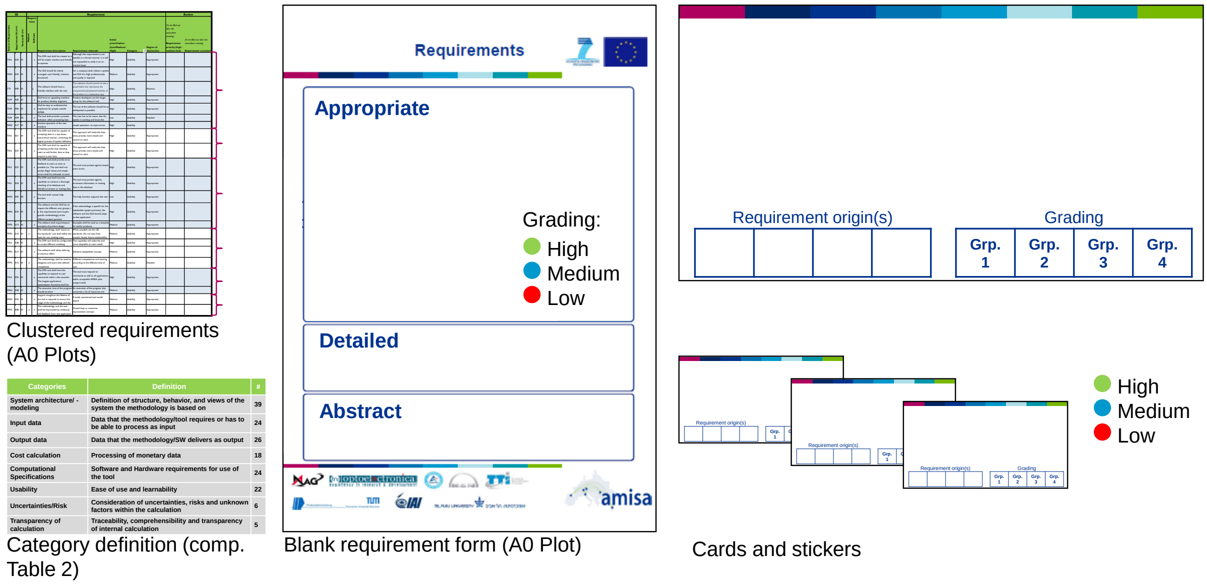

Figure 3. Station setup

The station setup included a plot of the list of clustered requirements, a category definition, a blank requirement form as well as cards and stickers to transfer requirements from the plotted list to the blank form and grade them (compare Figure 3). On the cards the possibility was given to document the requirements origin, stating which original requirement(s) the new requirement emanated from as well as the group conducting the consolidation for traceability reasons. The blank requirement form contained three boxes for the cards to be put into, representing *appropriate*, *detailed* and *abstract* degree of abstraction.

The groups were given the task to move the requirements from the plotted requirement list to the blank requirement list, concentrating the content of clustered requirements as much as possible by either removing redundancies directly or by rephrasing. Latter also allowed for an adjustment of the degree of abstraction "on the fly". Furthermore the grading of the importance into high, medium or low was asked to be performed by putting the corresponding sticker onto the card. Requirements that had been worked off were to be crossed out on the plotted list in order for the following groups to know which ones were left over for further consolidation.

Results and Lessons Learned

Within the workshop, the requirements were consolidated from 168 to 112 in total, representing a subtraction of 56 requirements and a reduction of approximately one third. 31 of the 112 requirements were still left "open", though, meaning that no final decision on the phrasing, the degree of abstraction and the prioritization was conducted. Those non-approved requirements were documented by the academic partner and given out to the consortium for further comments and grading propositions as preparation for Workshop II.

The timeframe for the group-work sessions must be described as rather too small. The groups did not have sufficient time to discuss all the requirements in detail, which was part of the reason why a considerable number of requirements was left open.

The approved requirements were consolidated in one excel-file whereas all information regarding the requirements origin was kept. This procedure assured maximum traceability for all partners and in the latter course of the project no issues due to non-considered requirements occurred.

The workload for the procedure can be summed up to approximately 200 man-hours for the entire consortium which is a considerable amount of time. Nevertheless the procedure resulted in a high degree of mutual understanding in terms of individual and overall perspectives. The partners reported a substantial gain in goal orientation and concretization.

Requirement Workshop II

The second requirement Workshop was conducted at the following TM two month after the initial consolidation and grading. The focus of the workshop lay on the clarification of the "open", non-approved requirements. Each of those 31 requirements was discussed in plenum and a consensus decision was aimed for. As described in the last section each partner had been asked to provide comments on the requirements, which was provided to the consortium in advance of the TM as a basis for discussion.

Procedure

The procedure was comparatively simple, for each open requirement the rationale, pro and con arguments as well as proposals for changes in the formulation had been asked of the consortium. The comments were bundled on one slide together with the requirement (Figure 4) and provided as basis for discussion in plenum. For the discussion of 31 requirements a 3 hour time period was foreseen, which allowed approximately 6 minutes each. In the discussion a consensus decision was aimed for, in case of prolonged discussion the possibility of postponement was left open to assure the compliance with the timeframe.

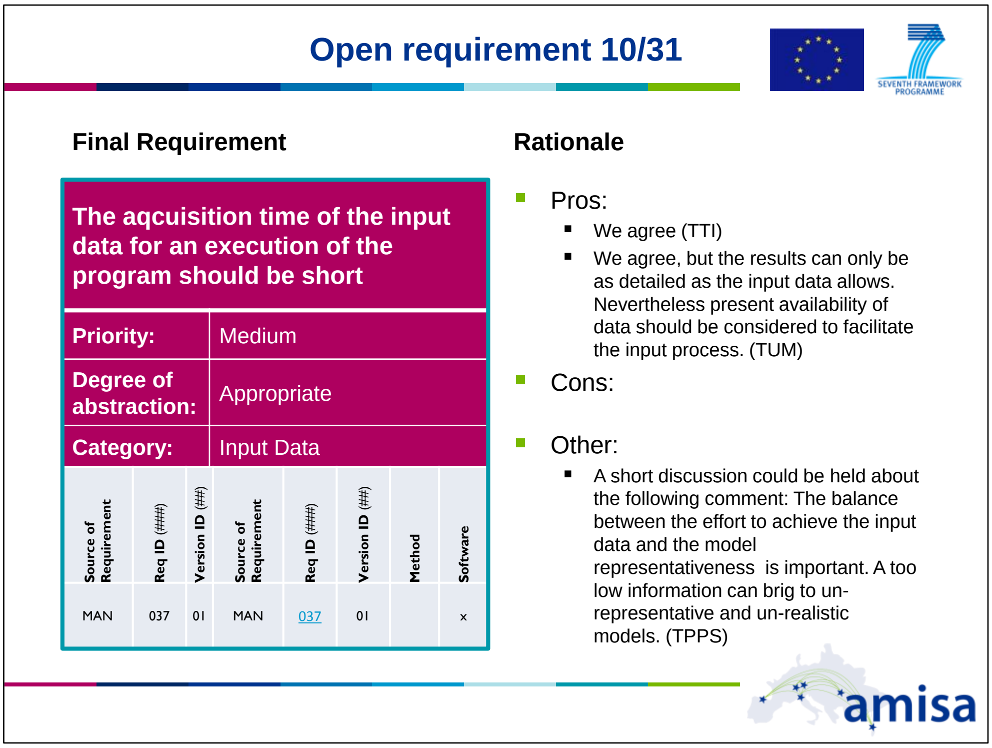

Figure 4. Slide layout open requirements

Figure 5 shows the transfer of an open requirement to an agreed requirement. In the specific case the degree of abstraction was decided to be changed from abstract to appropriate.

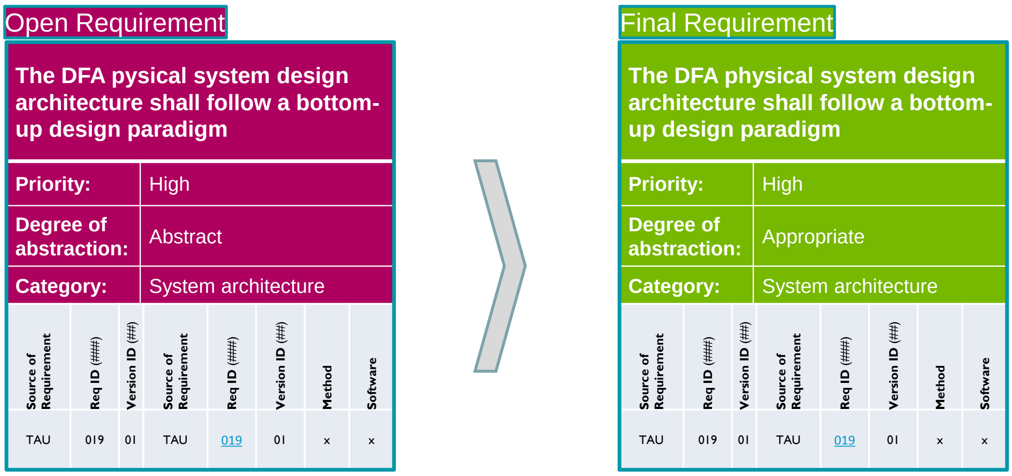

Figure 5. Agreement on open requirements

Results and lessons learned

Of the total number of 31 open requirements 27 were decided to be further considered but modified as to the formulation and degree of abstraction. Four were decided to be crossed out and not considered further, whereas the time frame could be kept. The changes were documented in the central excel list which at that stage contained 108 requirements. The file represented a deliverable to the EU-Commission and was reported 4 month into the project.

In the process of the requirement management it became evident that decisions on critical topics required time for clarification and therefore the multi-step process was conducted. Furthermore the opportunity to gather arguments and think through of implications caused by requirement changes was seen as essential by the partners to enable consensus decisions in the end, so it proofed advantageous to conduct sessions at different Technical Meetings.

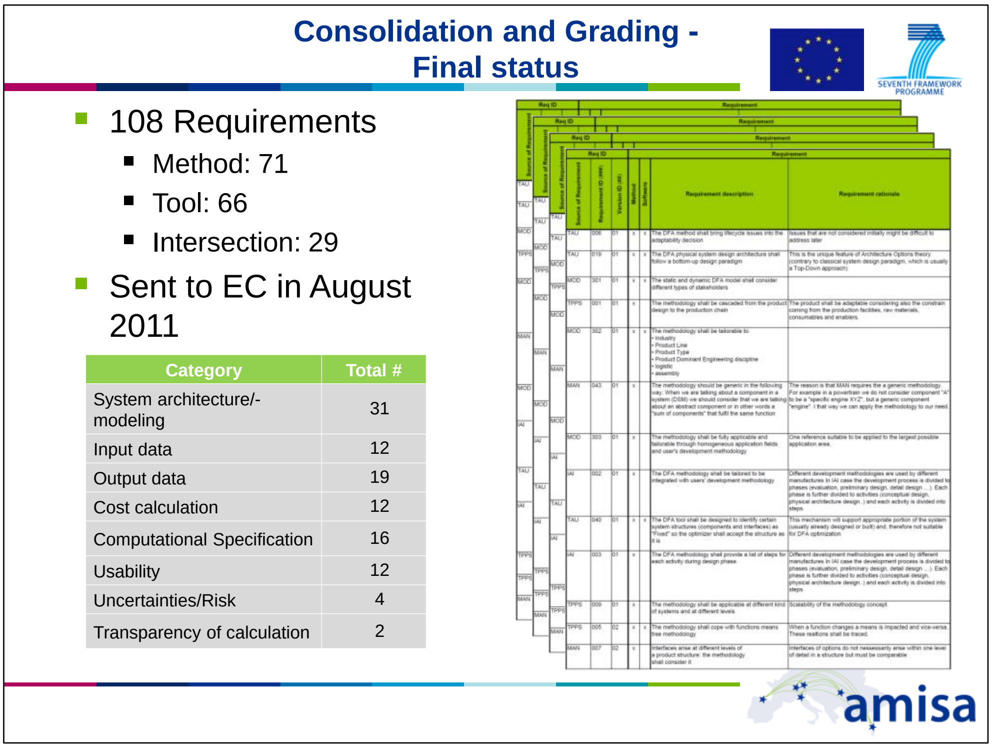

Figure 6. Final requirement status

2.3 Step 3: Status control and Management of critical requirements

Naturally, the requirement management process had the explicit goal to define the direction of methodology and software development and concretize goals as well as success criteria. In addition, the collected requirements were used as means of status control to monitor the project progress. That way the overall advancement as well as critical aspects could be made transparent for the consortium on a much more granular level than official project deliverables.

Procedure

The first major status control of the requirements' fulfillment was conducted one year into the project, eight month after the official requirement list had been handed to the EU Commission. Again transparency and consortium involvement were high goals. Therefore all partners were asked to individually judge the requirements fulfillment and rate it on a five-level scale from implemented to critical. Furthermore the importance of the requirements to the project could be stated in case changes were perceived.

The overview of requirements status could then be visualized for each category and the project as a whole. It was used as indirect check of work package progress, concept evaluation and, more than anything, identification of central points for further research. The input provided by the partners was in most cases not on the entity of all requirements but only on those specifically relevant to them individually. Therefore the feedback was not entirely uniform and the scheme depicted in figure 7 was introduced for the prioritization of the most critical requirements to be raised to the consortium attention. As soon as the judgment of one partner was located in the red area, the requirement was discussed in plenum.

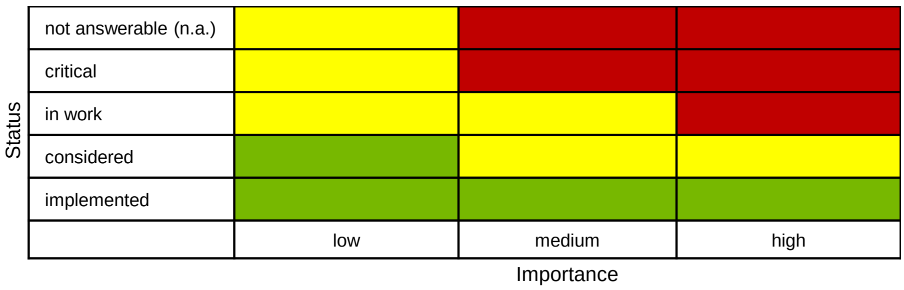

Figure 7. Requirements status check

Results and Lessons learned

The global project requirement list as well as a more detailed focus on two exemplary categories is shown in figure 8. Even though the depiction is evidently too small for the requirements to be read, an impression of the status can be derived for each of the lists.

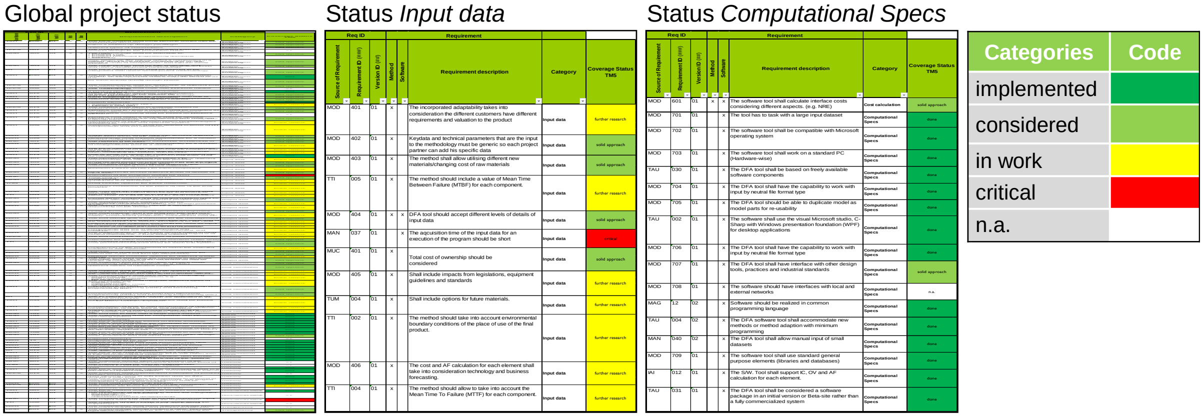

Figure 8. Requirements based status check

Overall, the project status derived from the requirements status check was in-line with the project plan, but a total of 16 requirements were rated to require direct attention. Those represented critical points that had not been explicit before and therefore provided significant value-add for the project.

Discussion of all requirements took place in plenum, whereas comments of each partner were provided and explained during the meeting. 10 of the 16 aspects could be resolved, but 6 were left for detailed discussion in a scientific meeting. Of those six, four had already been on the list of non-agreed requirements in the second consolidation and grading workshop four month into to project. This indicates a postponement of problem solving concerning critical requirements.

3 LESSONS LEARNED

There are three main aspects that can be highlighted in this section. First of all the uniformity of the input, achieved by providing templates at all stages, and the transparency regarding the processing of data, achieved by the complete avoidance of data deletion and provision of cross-links for user-friendly traceability have to be named. Consistent input significantly simplified the processing of data and lead to quick resolves of unclarities in meetings, which avoided ineffective discussions and iterations.

Second of all a multi-step requirements-management-process with central processing of the requirements and organization of the workshops proofed to be highly appreciated within the consortium. Between the steps of acquisition, consolidation and grading it became evident that decisions on critical topics required time for clarification and the opportunity to gather arguments and think through the implications caused by them. This was highlighted by the partners and contributed to a high degree of consensus decisions.

Thirdly, it showed to be beneficial to use the requirements' status to check overall project progress and highlight critical aspects on a more granular work-level than deliverables considered in project planning. The identification of critical points led to the opportunity to react to those in time with a scientific meeting, before project deliverables were affected. A critical aspect which could be witnessed is the postponement of the decision on the fate of critical requirements from one meeting to another. Within the project it was experienced that those requirements were strong indication for goal conflicts of different partners. Goals were considered disparate and only enhanced attention on project level and discussions in focus groups led to progress.

4 CONCLUSION AND DISCUSSION

The paper outlines the requirement-management-process for methodology development on the example of a Design for Adaptability methodology and software tool. The three main steps *requirements acquisition and categorization*, *requirement consolidation and grading* as well as *status control and management of critical requirements* are described in detail and related to their background, success and lessons learned.

Overall, the conducted procedure is judged to be reasonable in the course of the project, which at the stage of this paper has passed mid-term. The requirements towards the methodology could be acquired exhaustively in the early phase of the project and provided the basis for target oriented methodology development. Overall 108 requirements in eight categories were documented, whereas several workshops were needed to remove redundancies and achieve an adequate level of abstraction for all formulations. Nevertheless a total of 16 requirements remained critical half-way through the project of which 6 could not be resolved during technical project meetings and have been decided to be addressed during scientific meetings of the university partners.

Since methodology and tool development has not been finished, yet, the overall success of the requirement-management process remains to be assessed concludingly. The paper provides a guideline of how the task can be approached and highlights aspects of importance on base of the experiences made in the context of the project.

ACKNOWLEDGMENTS

Partial funding for this research work was received from the European Commission; project AMISA (Call identifier: FP7-NMP-2010-SMALL-4, Grant agreement: 262907). More information about the AMISA project is available at website: http://www.amisa.eu/

REFERENCES

European Commission (2011) Architecting Manufacturing Industries and systems for Adaptability, AMISA, <*http://cordis.europa.eu/fetch?CALLER=FP7_PROJ_EN&ACTION=D &DOC=4&CAT=PROJ&QUERY=0131132bc7b4:27e1:5c42c2c6&RCN=98517*> 2011-08-01.

Fletcher, D., Brennan, R. W., and Gu, P. H. (2009) A Method for Quantifying Adaptability in Engineering Design, *Concurrent Engineering-Research and Applications*, Vol 17.

Hashemian, M. (2005) Design for Adaptability, *Doctor's Thesis*, University of Saskatchewan, Saskatoon.

Kissel, M., Schrieverhoff, P., and Lindemann, U. (2012) Design for Adaptability – Identifying Potential for Improvement on an Architecture Basis. In *Proceedings of the NordDesign Conference – NordDesign 2012*, Aalborg, Aalborg University.

Schrieverhoff, P., Bender, T., Kissel, M., Lindemann, U. (2012) Evaluation of Architecture Options in Systems Engineering. In Marjanovic D., Storga M., Pavkovic N. and Bojcetic N. (eds) *International Design Conference - DESIGN 2012*, Dubrovnik, University of Zagreb/The Design Society.