Model-based Method to Utilize a Catalogue of Quality Requirements in Software Development

Lukas Strey
Fraunhofer FOKUS
Berlin, Germany
Email: lukas.strey@fokus.fraunhofer.de

Christian Hein
Fraunhofer FOKUS
Berlin, Germany
Email: christian.hein@fokus.fraunhofer.de

Tom Ritter
Fraunhofer FOKUS
Berlin, Germany
Email: tom.ritter@fokus.fraunhofer.de

***Abstract*—Complex software-based systems must comply with both functional and non-functional requirements (NFRs) to provide usefulness. This paper presents a structured catalogue of quality requirements and a model-based approach to collect NFRs from the catalogue in a given project context. The NFR catalogue is structured according to the quality criteria from the ISO 25000 series of standards and can be further extended. This catalogue can be applied in specific software development or modernization projects and in the preparation of tenders. This application to a specific project context is achieved by using the BPMN-NFR method presented in this paper. In this method, pattern recognition in system models is used to build a soft-goal model that serves as a filter mechanism for selecting relevant quality requirements from the catalogue. Through enrichment with context information concrete system related non-functional requirements are derived, which can be used for the system development. This model-based method was developed and applied in the context of the modernization of the budgetary procedures of Germany's federal government.**

I. Introduction

The development of extensive and distributed IT systems is a complex task, in which the correct implementation of the required functionality is crucial. Besides the functional requirements of a system, quality aspects are likewise important to develop a usable product. If a systems response time behaviour is not acceptable, it might be unusable despite offering all required functionality. Hence not regarding quality requirements in the development cycle can cause software projects to fail [1]. Non-functional requirements (NFRs), which are also referred to as quality requirements, should therefore be considered within the development cycle [2]. Acceptance of the need to prioritise and fulfil NFRs has risen but the elicitation and specification of non-functional requirements is a complex and difficult task [3]. NFRs must be adequately specified and later implemented in consideration of the systems structure, intended behaviours and the context the system shall be used in.

Methodical model-based requirements elicitation provides a structured approach to enable a high degree of coverage. The structure of a system can be analysed on different abstraction levels of logical and technical architecture models. Process models describe the intended behaviour of the system functionality and the interactions with users. For a concrete model-driven system development, the relevance of quality aspects has to be considered with regards to the given models in order to formulate precise NFRs. Existing quality norms and standards, which describe quality criteria in software development, provide an orientation [4].

In the project context of the modernization of the German federal budget procedures, quality assurance in all modernization steps is firmly planned from the beginning. The software system developed within these modernization projects must attest to a high degree of quality to be usable for federal budget processes. A formal methodological approach is being adopted as early as the requirements elicitation stage and high priority is given to the consideration of NFRs. The objective is to achieve a high coverage of NFRs in tenders and a high degree of re-usability for similar projects. At the same time, technical experts in budgetary procedures are to be intensively involved in the modernization process. Furthermore, the modernization process will focus on modelling as extensively as possible in order to achieve a high level of quality, e.g. in terms of consistency of requirements.

Experience gained in the practical application of model-based requirements elicitation allows for the recognition of patterns and shows the positive effects of reusing existing requirements. Based on this, we developed a methodical approach and applied it to a use case in the context of the federal budget procedures modernization. To support this methodological approach, a collection of NFRs is used as a basis to build upon. This allows to provide the context experts with the required knowledge of quality characteristics. Additionally the quality information gathered in a project can be collected and reused in future projects.

In this paper we give an overview of this developed method for the model-based elicitation of non-functional requirements in software development / modernization projects. The method encompasses (1) the creation and maintenance of a catalogue of quality criteria, which provides a set of generic NFRs to be reused in the project context, (2) the analysis of system models regarding the relevance of quality characteristics using pattern recognition, (3) a screening of the NFR catalogue by use of the relevance analysis and (4) the derivation of concrete NFRs chosen for the specific system models, which describe the non-functional requirements of this system.

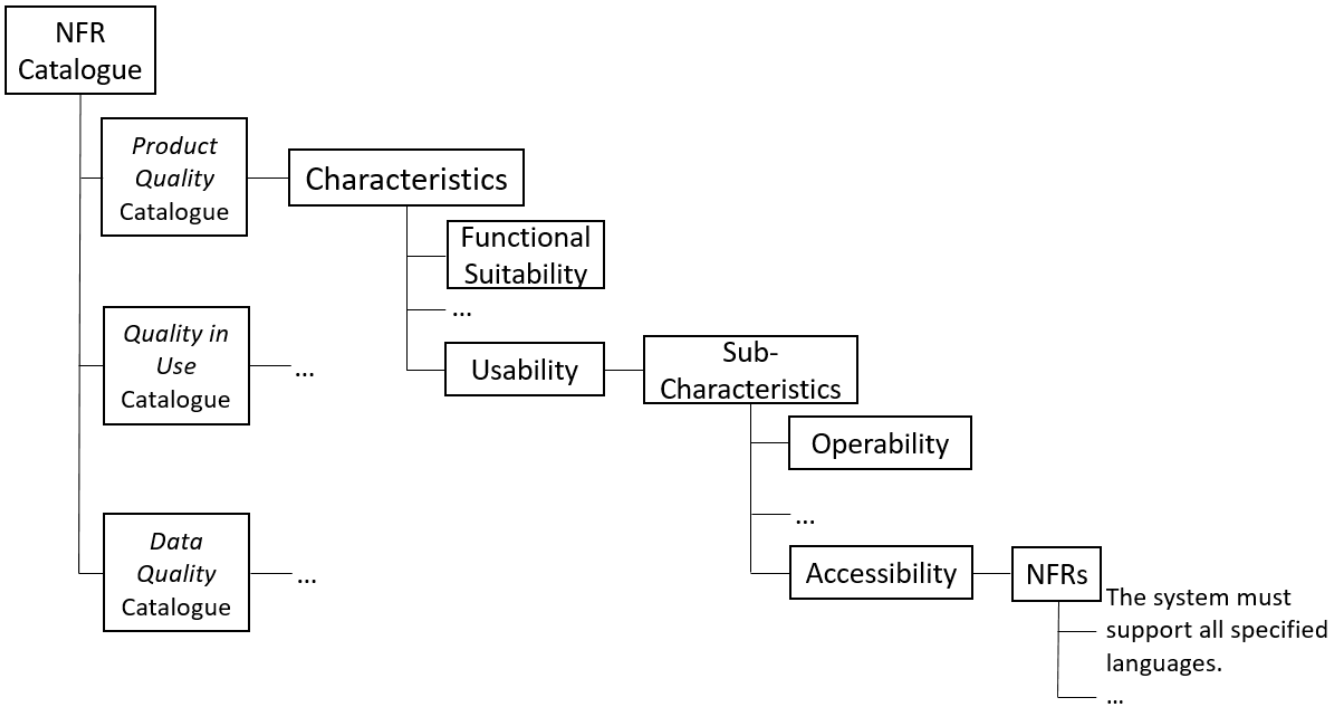

Fig. 1. Structure of the NFR catalogue corresponding to the ISO SQuaRE series of standards

II. Catalogue of NFR

The procedure described in this paper is based on the use of a NFR catalogue, which provides a knowledge repository of preset NFRs. These NFRs are each assigned to specific quality criteria within the catalogue structure. The NFR catalogue contains NFRs described in generic form, which can be enriched with context information for use in a concrete system development project. As the basis for the catalogue structure the ISO 250xx series of standards is used. Nevertheless the structure is flexible, expandable and can be iteratively enriched with new knowledge.

A. ISO 250xx

The ISO standard 25000 is the first in the series of standards for "System and Software Quality Requirements and Evaluation (SQuaRE)" [4]. ISO standard 25010 describes the quality models for *Product Quality* and *Quality in Use* [5]. The ISO 25012 standard also describes a model for *Data Quality* [6]. These quality models specify individual characteristics that relate to the respective quality aspect of the model. For *Product Quality* and *Quality in Use* each characteristic is subdivided into individual sub-characteristics. Together, the characteristics of the three models comprehensively describe the quality of a software product. ISO standards also provide measures to quantify the characteristics of the three quality models, ISO standard 25022 for *Quality in Use* [7], standard 25023 for *Product Quality* [8] and standard 25024 for *Data Quality* [9].

B. Building the catalogue

Figure 1 shows the structure of the NFR catalogue, with the categorization of NFRs into individual sections according to the three quality models, as well as the structure of characteristics and sub-characteristics. The complete list of characteristics can be found in the corresponding ISO standards. The measures described in the ISO standards 25022, 25023 and 25024 serve as a basis for the formulation of a first version of generic NFRs. At least one NFR, which quantifies the requirement in terms of the quality characteristic, is derived from each measure. The first version of generic NFRs effectively call for a positive fulfilment of the quality measure they were derived from. The NFRs obtained in this way are sorted into the structure of the NFR catalogue according to the ISO quality characteristic the measure referred to. The version of the catalogue thus created contains 191 generic NFRs. In the first version of the catalogue, the terms of the English ISO standards have been adopted and the NFRs are also formulated in English.

Each generic NFR in the catalogue refers to a subject or object within the development project, such as 'system', 'service' or 'database'. To define the priority of an NFR, the verb within the NFR is used (must, should, could, won't). An example NFR from the catalogue is: "The system must support all specified languages" (Product Quality - Usability - Accessibility); where both subject and verb can be defined context-specifically by the user, for example "The user interface should support all specified languages". Furthermore, the user must set concrete values in some quantifiable NFRs: "The system

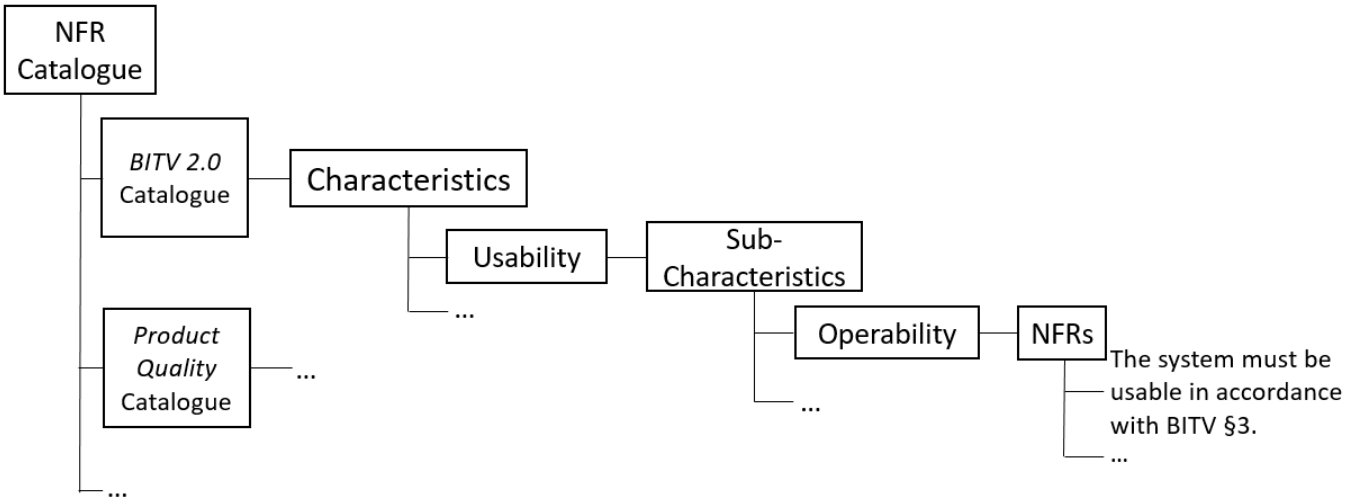

Fig. 2. Expanding the catalogue with BITV 2.0

must have a test coverage of at least [X]%" (Product Quality - Reliability - Maturity). This allows the NFR catalogue to integrate context specific information from the development project.

Not all ISO measures are specific enough to formulate a detailed NFR for their respective quality characteristics. In these cases, knowledge from former quality projects has been used to formulate accurate descriptions of the first iteration of the respective NFRs. This was achieved by choosing existing NFRs from successful past development projects that relate to the ISO measure and generalizing them for an inclusion into the catalogue. Additional work has been done to expand upon the catalogue, which will be described in the following subsection.

C. Expanding the catalogue

The initial catalogue can be expanded in two ways. New NFRs can be added into the existing structure, using the functionality of the catalogue as a knowledge storage. This continuous building of the catalogue must be carried out carefully. Before adding new NFRs, a duplicate check should be performed and the level of detail in the catalogue should be kept constant. In addition, the structure of the catalogue can be expanded by defining new sub-catalogues with corresponding characteristics and sub-characteristics. This allows recurring, project-specific contexts to be included. This is relevant for special regulations in the context of modernizing the federal budget procedures.

The first extension of the catalogue is the addition of the Barrier-free Information Technology Ordinance (BITV 2.0) [10]. The BITV 2.0 regulates the accessibility of electronic services of public authorities and electronic administrative procedures in Germany. NFRs obtained from the analysis of the BITV 2.0 are collected in a separate BITV 2.0 catalogue due to the special level of detail. Figure 2 illustrates the extension of the NFR catalogue by a special sub-catalogue for BITV 2.0 NFRs. Including this extension, the currently used version of the catalogue contains a total of 207 generic NFRs that can be used to elicit NFRs for a given project context.

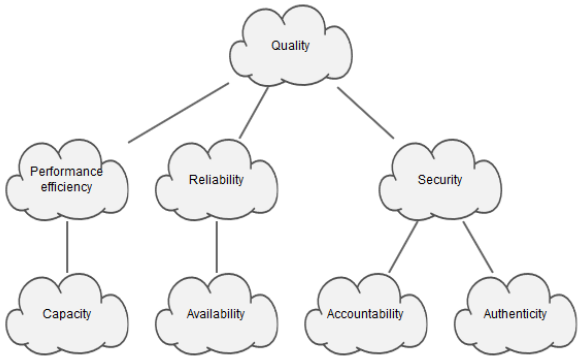

Fig. 3. Softgoal tree example

III. NFR ELICITATION USING THE CATALOGUE

The next step is to use the catalogue in a given context to create concrete NFRs for that project. With the continuous expansion of the catalogue, a method for filtering it is necessary. In this method the total set of NFRs in the catalogue is filtered in a sensible way and offered to the user for selection. The filtering takes into account the structure of the catalogue and allows a basic pre-selection according to relevant sub-characteristics, thus reducing the number of NFRs to be considered. Both architecture and process models can be used as input for filtering the catalogue. The input of a process description is particularly relevant for automation and the alignment of business processes and IT. Softgoal modelling is used to describe and display the filtering, an example is given in Figure 3.

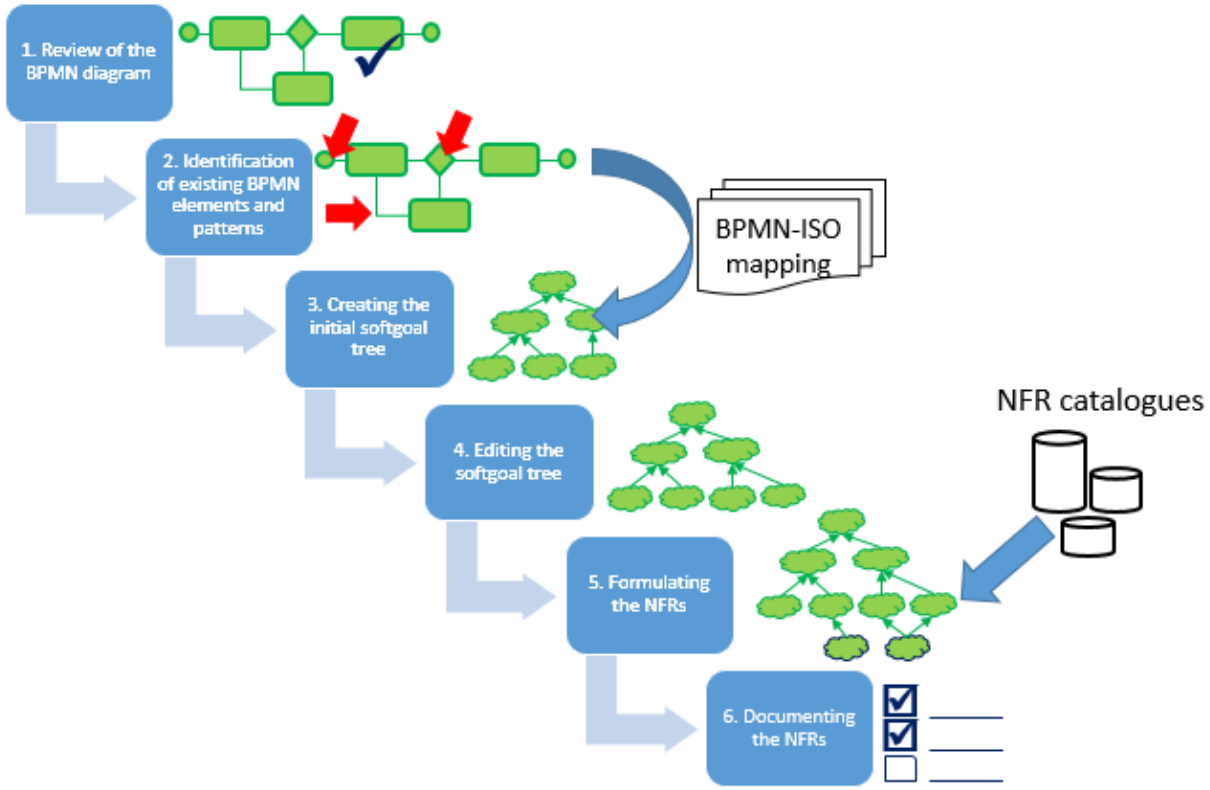

Fig. 4. Steps of the BPMN-NFR method

A. Model-based elicitation

Softgoal Interdependency Graphs (SIG) are used to model softgoals, which are goals that cannot be clearly fulfilled. SIG visualise softgoals with their mutual influences, e.g. whether a softgoal interferes with or supports other softgoals [11]. Softgoals are used to model the filter criteria when using the NFR catalogue. A tree structure of softgoals is created for this purpose. This softgoal tree depicts on its lowest level the sub-characteristics which have been identified as relevant in the course of a methodical procedure. The following subsection describes this method based on BPMN diagrams. The sub-characteristics are used as filter criteria for the catalogue. At the top level of the softgoal tree is the softgoal "Quality", to which the ISO characteristics are subordinated according to their structure. Figure 3 displays an example of such a tree structure. Based on the final softgoal tree, NFRs can be selected from the filtered catalogue.

Modelling using the softgoal tree allows representing the influence of NFRs selected from the catalogue on the softgoals. Thus, potentially negative influences between elements of the softgoal tree can be modeled if the fulfillment of one ISO sub-characteristic negatively affects another. This is illustrated in Figure 5, which shows two softgoals at the lowest level of the tree structure with an NFR selected from the catalogue. These influences are currently modeled manually but it is planned to store them in the NFR catalogue in future iterations.

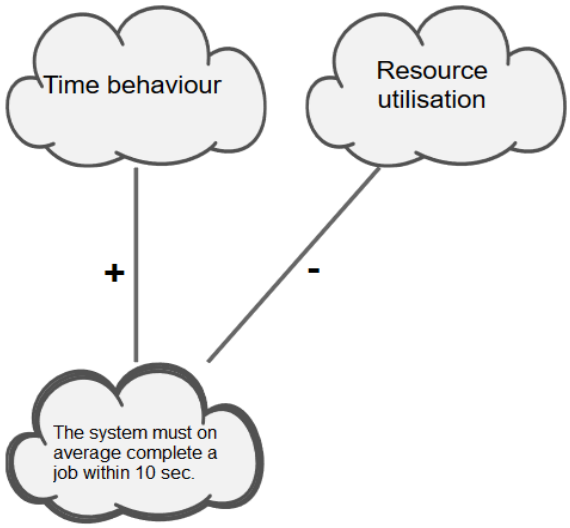

Fig. 5. Influences of an NFR onto softgoals

B. BPMN-NFR

In this section, the BPMN-NFR method for the collection of non-functional requirements out of the NFR catalogue using process models is presented. The "Business Process Model and Notation" (BPMN) of the Object Management Group (OMG) is commonly used to model business processes [12].

Among others, the REMO method of Vieira et al. uses BPMN diagrams as input for requirements elicitation [13]. NFRs are collected in REMO using brainstorming based on activities and message events [14]. In contrast, the method presented here uses elements and patterns present in the BPMN diagram to build a softgoal tree and then filter the NFR catalogue based on the sub-characteristics presented in the softgoal model. Figure 4 shows the steps of the method.

In summary, the method consists of six successive steps:

1) Review of the BPMN diagram
2) Identification of existing BPMN elements and patterns
3) Creating the initial softgoal tree
4) Editing the softgoal tree
5) Formulating the NFRs
6) Documenting the NFRs

Basic requirement for the application of the method is a contentwise correct and up-to-date BPMN diagram, which is checked in the first step. Thereupon, predefined BPMN patterns and elements are identified in the diagram and marked accordingly. Selected ISO sub-characteristics have been assigned to each pattern/element. This general assignment is a basic component of the BPMN-NFR method and is of particular importance for the successful implementation of the NFR elicitation. With the help of the defined assignment an initial softgoal tree is generated automatically. When using a tool implementation of the method, the third step is therefore without manual effort.

The generation of the initial softgoal tree is based on the identification of predefined patterns in the input. For each specified pattern there are potentially relevant NFR categories assigned. If a certain pattern is found in the input during the analysis, the corresponding categories are inserted into the softgoal tree. The van der Aalst Workflow Patterns are a collection of patterns that can occur in process models [15]. Wohed et al. analyzed to what extent the workflow patterns can be represented in BPMN [16]. Taking into account this analysis, the patterns as well as the individual BPMN elements were examined for their relevance to the elicitation of NFRs. This examination is based on the ISO measures of the quality models. To generate the initial assignment from workflow pattern to ISO sub-characteristics an analysis was carried out for all workflow patterns to determine whether an implementation of the pattern could theoretically influence the respective measure. For selected workflow patterns two to four potentially relevant ISO sub-characteristics were identified.

This mapping was used, checked and refined in several application examples. For each use case the BPMN-NFR method was executed. If extensive manual changes were necessary in step four, then the initially generated softgoal tree was potentially insufficient. The necessary changes of the different use cases were compiled and compared, which allowed conclusions to be drawn about the quality of the initial assignment of workflow patters to ISO sub-characteristics. As a result necessary adjustments could be identified and the mapping was iteratively improved. The revised result is listed in Table I.

TABLE I
MAPPING OF WORKFLOW PATTERN TO PRODUCT QUALITY CHARACTERISTICS

Workflow Pattern	**ISO sub-characteristics**
Parallel Split	Capacity; Co-existence; Modularity
Synchronisation	Resource utilisation; Interoperability
Exclusive Choice	Maturity; Modularity
Multiple Choice	Capacity; Co-existence; Maturity; Modularity
Discriminator	Resource utilisation; Interoperability; Fault tolerance
Arbitrary Cycles	Resource utilisation; Maturity
Multiple Instances	Time behaviour; Resource utilisation; Capacity
Deferred Choice	Time behaviour; Maturity; Modularity
Cancel Activity	User error protection; Maturity; Recoverability
Cancel Case	Maturity; Recoverability; Accountability; Analysability

After the automated generation in step 3 of the BPMN-NFR method, the softgoal tree is edited by a project expert using the existing context information. Individual softgoals can be added or removed. The manual editing is step 4 of the method, according to the procedure in Figure 4. The automatic generation of the initial softgoal tree is based on general assumptions and therefore not necessarily fully transferable to the context of every project. The manual editing step allows the inclusion of context-specific aspects. On the one hand, elements existing in the initial softgoal tree are rejected if they are not relevant to the concrete application. On the other hand, softgoals are added manually, whose necessity can result from further project information not specifically modeled in BPMN.

The softgoal tree is then used to filter the NFR catalogue for potentially relevant NFRs. The final selection of NFRs from the catalogue is made by using the existing context information, e.g. the system and expert knowledge available about the process. Finally, the NFRs selected from the catalogue must be documented. Due to the model-based approach, detailed traceability information is made available for the documentation.

C. Tool-support

We have implemented the described BPMN-NFR method in a tool chain framework based on the components ModelBus [17] and ModelICE.io [18]. These components allow the editing of all models in the web browser and common modelling tools (such as Enterprise Architect, Eclipse Modelling, Papyrus), additionally libraries are integrated and applicable via adaptors. The given framework was used to implement a tool support for the BPMN-NFR method due to the possibility to display different diagrams on the modelling canvas. This allows us to display the process diagram and softgoal tree together with gained traceability information. In the course of the model-based approach, special attention should be given to traceability information. Traceability must be possible across

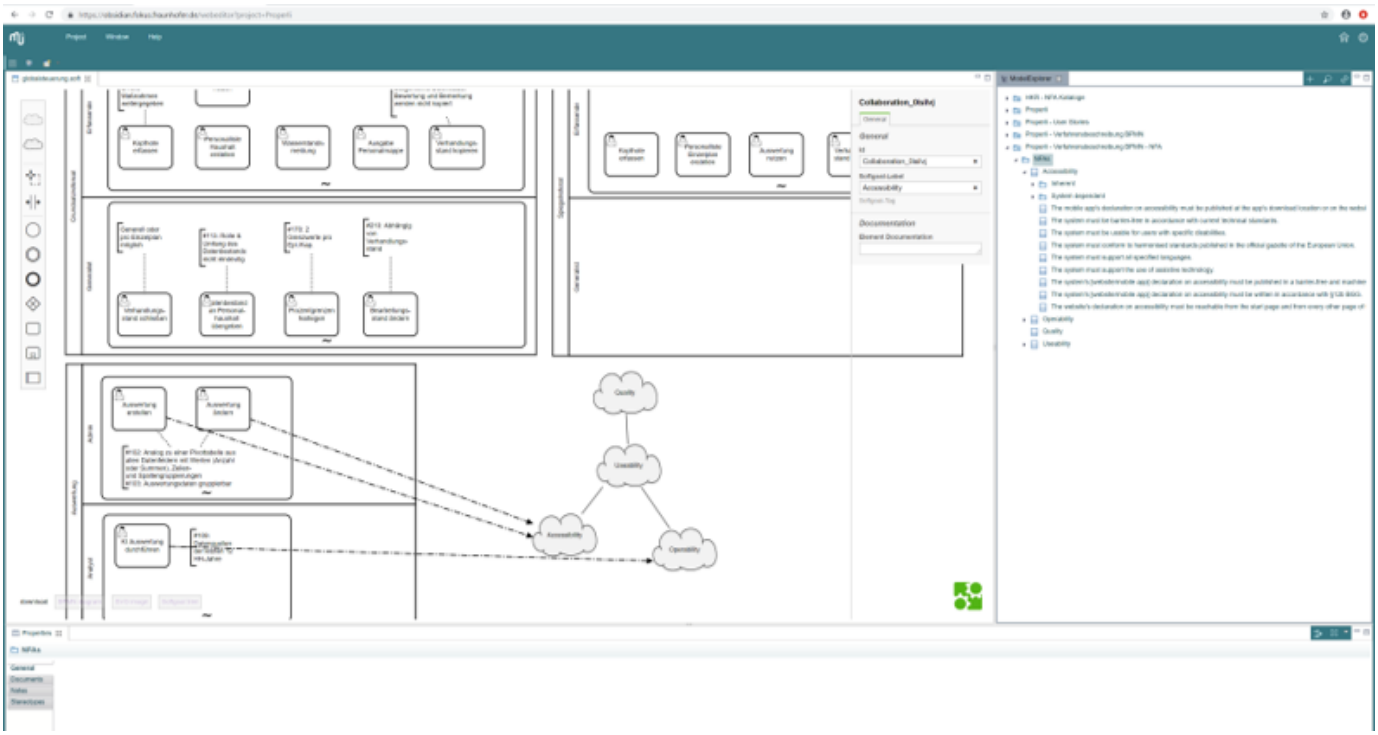

Fig. 6. Editing BPMN and softgoals with ModelICE.io

model boundaries because of dependencies between process descriptions (in BPMN), architecture descriptions (mainly in SysML) and softgoal trees.

For the implementation of the method, we integrated an extension of bpmn-js into the tool ModelICE.io to allow the description of the processes and their relation to softgoal trees. Figure 6 shows a screenshot of the implementation. In addition to the representation of the softgoal trees, we have developed an extension, which filters the NFR catalogue for concrete NFRs using the softgoals. The use of ModelICE.io makes it possible to have this functionality available as a service in the tool chain and thus use it in other tools such as Enterprise Architect.

IV. Use Case

The developed BPMN-NFR method is used in the context of the modernization of the federal budget procedures for the collection of NFRs. For this purpose, the described tool chain is used to partially automate it. In the following section the application of the BPMN-NFR method to the example use case ProPerLi is described. BPMN models of the process to be implemented and the mapping of BPMN to the *Product Quality* sub-characteristics described in the previous chapter are used.

A. ProPerLi project

The ProPerLi (production of personnel lists) project is intending to create a software system to support the preparation of the federal personnel budget (positions of civil servants and of employees covered by collective agreements). The aim is to develop a continuous, IT-supported procedure that will cover the entire process. All process segments from the recording of all personnel demands, the assessment of these demands, the presentation of negotiation results within the federal government and of changes from the parliamentary procedure to the production of personnel lists and the transfer of the data stock to an existing database are to be implemented. The process is currently carried out manually, supported by office software. It is characterised by data discontinuity and is inspection-intensive due to missing automated error and plausibility checks. The processes are made available as structured BPMN diagrams within the project context.

With the help of the modelled BPMN diagrams the NFRs for ProPerLi are collected. The individual steps of the BPMN-NFR method are shown in Figure 4. The collection of the NFRs is done by using the NFR catalogues. A total of 29 process-related NFRs from the *Product Quality* catalogue were collected for ProPerLi using the BPMN-NFR method. A softgoal tree with a total of 18 sub-characteristics was used. Selected NFRs are listed as examples in Table II.

B. Evaluation

The presented BPMN-NFR method and the use of a NFR catalogue for requirements elicitation are evaluated below. The results of a predecessor project of ProPerLi, in which NFR for the process of personnel lists were collected by an external consulting firm, serve as comparison data. The consulting company created an NFR collection based on existing, unstructured original requirements in the federal procedures. This NFR collection was prepared for application in the

TABLE II
EXAMPLE NFRS FOR THE PROPERLI PROJECT ELICITED WITH BPMN-NFR

ID	Sub-characteristic	NFR
1	Capacity	The system must allow at least 5 users to use the system simultaneously.
14	Accountability	The system must audit all user access to the system or data.
18	Availability	The database must be available for 95% of the scheduled system operational time.
22	Operability	The system must support all appropriate input devices.
29	Analyseability	The system must implement all specified diagnosis functions.

context of the modernization of the federal budget procedures and structured according to the categories of the ISO 9126 standard. The NFR collection of the consulting firm is in the following referred to as comparative NFRs.

In this paper, a structured NFR catalogue was presented, based on the quality characteristics and the measures of the ISO 250xx series of standards. The catalogue closes content gaps which existed in the comparative NFR collection. Furthermore, the NFR catalogue was created with the use of fewer resources than the comparative NFRs. The NFR catalogue thus fulfils the general requirements that process participants placed on a NFR collection and which were not sufficiently covered by the external consulting company. The NFR catalogue fulfils its role as a knowledge collection and enables the reuse of NFRs in other projects, mainly within the context of the budget procedures modernization. Nonetheless, further work is possible to extend the catalogue for other specialised application contexts.

The structured application, using softgoal modelling based on BPMN, enabled a comprehensible overview of NFRs for all stakeholders in the ProPerLi process. By structuring the procedure, coverage of the problem area - specifically the process descriptions - is ensured. The tool implementation of the method allows a partial automation of the steps. In addition to the extended coverage of the problem area, the method ensures that all stakeholders understand the process flows with the help of BPMN modelling.

Finally, the presented method is to be evaluated quantitatively. The comparative NFRs comprise 19 NFRs defined as generally applicable. Based on the categorization according to ISO standard 9126, a comparison with the *Product Quality* sub-catalogue is preferable. It contains a total of 91 generic NFRs at a more detailed level of abstraction, of which 29 were elicited for the ProPerLi process using the BPMN-NFR method. An analysis of the contents shows that the requirements of the *Product Quality* catalogue are more concrete than the comparison NFRs. In total 17 of the 19 comparison NFRs are covered by the 29 ProPerLi NFRs elicited. The comparative NFRs are described in more detail by the new BPMN-NFRs. The two comparison NFRs not covered address the anonymization of test data and the maintenance concept. These two aspects are not covered by the process view. Accordingly, the elicitation could be extended by an architecture analysis to check the system context for further missing NFRs. A backward comparison considering the categories of the ISO 9126 standard shows that seven BPMN NFRs are not mapped at all in the comparison NFRs and five others are not sufficiently mapped. Thus, 24% of the BPMN-NFRs elicited are completely new for the project context and a total of 41% represent an extension of the previously existing NFRs. These new BPMN-NFRs were chosen by process experts and developers to be used in the implementation of ProPerLi instead of the formerly available comparison NFRs.

V. RELATED WORK

The reuse of known requirements in new software projects can increase the efficiency of requirements elicitation. Toval et al. propose the SIREN method for this purpose. SIREN uses a spiral approach, where requirement catalogues are used, e.g. a catalogue for security profiles [19]. The NFR Framework provides catalogues in the form of Softgoal Interdependency Graphs (SIG) [11]. Cysneiros provides further catalogues, also mapped in SIG, and the i* system for systematic use of catalogues [20]. In the literature SIG for usability [21], performance [22] and invisibility [23] are mentioned among others. The NFR catalogue described in this paper sets itself apart by its structure based on ISO standards and an uncomplicated extensibility.

A structured survey of NFRs can be carried out using, for example, the SENoR method, which uses NFR templates based on the characteristics of the ISO 25010 standard. This catalogue of templates is used in brainstorming sessions during the method to support the stakeholders in the NFR survey [24]. The PABRE system provides a collection of requirement patterns for some of the ISO 25010 characteristics [25]. The BPMN-NFR method collects structured NFRs using the presented NFR catalogue. A similarly structured approach is also possible for architectural models. With the partial automation of work steps, manual effort is reduced. This automation can be extended within the framework of tool support.

VI. CONCLUSION

With the help of the expandable catalogue of quality criteria and the procedure for building the softgoal tree, concrete NFRs can be derived for architecture and process models. In this paper the application of the method for process models was presented and examined. Due to the systematic procedure, a comprehensive and at the same time precisely fitting set of NFRs is created. The web-based tool we implemented for the creation of BPMN and softgoal models enables interactive

team-spanning work, which is helpful for revising the models with the help of experts.

Promising results have been achieved in the application of the method so far. In particular, if the collection of non-functional requirements is started from scratch, it is relatively easy to generate a comprehensive design of NFRs for the different parts of a system in order to enter into a discussion with stakeholders at an early stage. The structured allocation of the generated NFRs to the ISO characteristics used in the catalogue also facilitates the further processing of the requirements in tenders and in the technical implementation.

So far, the method has been mainly used in the specific context of the federal budget procedures presented in this paper. Further use and application to other development contexts is necessary to extend the catalogue further and to identify potentially new mappings between workflow patterns and ISO characteristics.

A. Outlook

The first results of previous applications of the method in the course of the modernization of budgetary procedures and especially in the sub-module ProPerLi (production of personnel lists) yielded positive results in the software development and in the eyes of the project stakeholders. The knowledge gained will be used to expand the catalogue and improve the method. Currently the patterns are partly marked manually, therefore the extensive automation of pattern recognition through improved tool support is our next goal. Besides the improvements to the existing BPMN-NFR method, a worthwhile and necessary expansion is the inclusion of architecture models into the analysis. The method is to be used in different application areas in the future, which will allow the currently used patterns to be checked and further refined. In addition to the application in the ProPerLi submodule, the method will also be used in the future to support the modernization of other federal budget procedures.

References

[1] J. Doerr, D. Kerkow, T. Koenig, T. Olsson, and T. Suzuki, "Non-functional requirements in industry – three case studies adopting an experience-based nfr method," in *Proceedings of the 13th International Conference on Requirements Engineering*, 2005.

[2] L. Chung and B. A. Nixon, "Dealing with non-functional requirements: Three experimental studies of a process-oriented approach," in *17th International Conference on Software Engineering*, 1995.

[3] M. Rahman and S. Ripon, "Elicitation and modeling non-functional requirements – a pos case study," in *International Journal of Future Computer and Communication*, 2014.

[4] ISO, *ISO/IEC 25000:2014 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*. International Organization for Standardization, 2014.

[5] ——, *ISO/IEC 25010:2011 System and software quality models*. International Organization for Standardization, 2011.

[6] ——, *ISO/IEC 25012:2008 Data quality model*. International Organization for Standardization, 2008.

[7] ——, *ISO/IEC 25022:2016 Measurement of quality in use*. International Organization for Standardization, 2016.

[8] ——, *ISO/IEC 25023:2016 Measurement of system and software product quality*. International Organization for Standardization, 2016.

[9] ——, *ISO/IEC 25024:2015 Measurement of data quality*. International Organization for Standardization, 2015.

[10] B. 2.0, *Barrierefreie-Informationstechnik-Verordnung - BITV 2.0*, 09 2011.

[11] L. Chung and J. Leite, "On non-functional requirements in software engineering," in *Conceptual Modeling: Foundations and Applications*. Springer, 01 2009, pp. 363–379.

[12] OMG, *Business Process Model and Notation Specification Version 2.0*. Object Management Group, 2011.

[13] S. Vieira, D. Viana, R. Nascimento, and T. Conte, "Using empirical studies to evaluate the remo requirement elicitation technique," *SEKE 2012 - 24th International Conference on Software Engineering and Knowledge Engineering*, 2012.

[14] ——, "Evaluating a technique for requirements extraction from business process diagrams through empirical studies," *38th Latin America Conference on Informatics, CLEI 2012 - Conference Proceedings*, pp. 1–10, 2012.

[15] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, and A. Barros, "Workflow patterns," *Distributed and Parallel Databases*, p. 5–51, 2003.

[16] P. Wohed, W. M. P. Aalst, M. Dumas, A. Ter, and N. Russell, "On the suitability of bpmn for business process modelling," *Lecture Notes in Computer Science*, 2006.

[17] "Fraunhofer fokus: Modelbus, https://www.modelbus.org," 2019, online, zuletzt aufgerufen: 13.10.2019. [Online]. Available: https://www.modelbus.org

[18] "Ingrano solutions: Modelice, https://www.modelice.io," 2019, online, zuletzt aufgerufen: 13.10.2019. [Online]. Available: https://www.modelice.io

[19] A. Toval, B. Moros Valle, J. Nicolás, and J. Lasheras, "Eight key issues for an effective reuse-based requirements process," *Computer Systems Science and Engineering*, vol. 23, pp. 373–385, 11 2008.

[20] L. M. Cysneiros, "Evaluating the effectiveness of using catalogues to elicit non-functional requirements," in *WER*, 2007.

[21] L. Cysneiros, V. Werneck, and A. Kushniruk, "Reusable knowledge for satisficing usability requirements," in *13th IEEE International Conference on Requirements Engineering (RE'05)*, 2005, pp. 463–464.

[22] Y. Yu, J. Mylopoulos, E. Yu, J. Leite, L. Linda, and L. Liu, "Software refactoring guided by multiple soft-goals," in *1st workshop on Refactoring: Achievements, Challenges, and Effects (REFACE'03)*, 2003.

[23] R. M. Carvalho, R. M. Andrade, K. M. Oliveira, and C. Kolski, "Catalog of invisibility requirements for ubicomp and iot applications," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, 2018, pp. 88–99.

[24] S. Kopczynska and J. R. Nawrocki, "Using non-functional requirements templates for elicitation: A case study," *2014 IEEE 4th International Workshop on Requirements Patterns (RePa)*, pp. 47–54, 2014.

[25] C. Palomares, C. Quer, and X. Franch, "Requirements reuse with the pabre framework," *Requirements Engineering Magazine*, 01 2014.