INTERNATIONAL DESIGN CONFERENCE – DESIGN 2020
https://doi.org/10.1017/dsd.2020.52

CONCEPTUAL MODEL FOR (SEMI-) AUTOMATED DERIVATION OF EVALUATION CRITERIA IN REQUIREMENTS MODELLING

D. Horber ✉, B. Schleich and S. Wartzack

Friedrich-Alexander Universität Erlangen-Nürnberg, Germany
✉ horber@mfk.fau.de

Abstract

Requirements act as a limitation of the solution space, which represents the stakeholders' needs and guides the whole product development process. Therefore, forgotten requirements can lead to wrong decisions when using them as a basis for decision-making. This contribution introduces a novel approach to link the requirement and evaluation criteria models to address this problem. For setting up those criteria consistently, the requirements are classified using natural language processing and derived by a ruleset based on a developed mapping between requirement classes and criteria types.

Keywords: requirements management, decision making, design evaluation, natural language processing

1. Introduction

For the development of marketable and innovative products, requirements are the central element guiding not only the design itself but also the entire development process. Not considering all necessary requirements adequately leads to products being developed, which do not fulfill the customer's needs and thus fail in the market. The task of requirements engineering (RE) is therefore to identify the relevant requirements of the various stakeholders who have an interest in the product to be developed (Rupp, 2014). Starting at the beginning of the product development process, requirement specification is one of the first steps to clarify the projects' main goal and as the project progresses, iteratively extending or adapting requirements can be necessary in order to set up a comprehensive requirement model. To gain more benefit from a consistent and continuous requirement model, it is possible to link it with other product models as described by model-based systems engineering (Walden et al., 2015).
During development, product developers are confronted by various decision situations, such as the decision for further detailing of one product concept. Requirements limit the solution space and have to receive consideration also within decision-making. Therefore, product requirements can be seen as evaluation criteria in a certain way and must be transformed into these criteria (Horber et al., 2019). Consistent criteria models are necessary in order to prevent from wrong decisions caused by insufficient consideration of single criteria. To counter the described problem, this paper presents a concept for the (semi-) automated derivation of evaluation criteria in consideration of all product requirements. This enables developers to receive feedback on the purpose of requirements in decision-making as early as the requirement specification. On one hand, the concept ensures the continuity of the models and on the other hand minimizes the risk of considering criteria insufficiently. Especially when it comes to large projects dealing with complex systems and a high number of requirements, a lot of effort is required to

set up and maintain all product models (Shea, 2017). The presented approach addresses this issue and assists in the linkage of requirement and criteria model, in order to reduce this effort and improve the quality of the overall decision model. Within this contribution, the state of the art is described at first, and the relevant research questions are derived next. Subsequently, the concept as well as an exemplary demonstrator are presented, followed by a critical discussion of both.

2. Background and state of the art

As stated before, taking every element of a requirement model into account is of high importance for successful product development and therefore minimizes the risk of miss-developed products. In the literature on RE the fundamental knowledge and methods are described (Fernandes and Machado, 2016). The RE process begins with the clarification of the task and continues with the identification of relevant requirements as well as their specification. Kamata and Tamai (2007) state, that requirement quality has a major impact on the market success or failure of the product and is therefore necessary for a successful development process. Requirements quality can be differentiated into the quality of single requirements as well as the quality of a set of requirements (Requirements Working Group, 2017). To improve the quality of these sets, factors like completeness, consistency or feasibility are necessary and have to be considered. Requirement statements contained in these sets have to be (amongst others) necessary, appropriate and complete to ensure a sufficient level of quality (Requirements Working Group, 2017). Taking these quality criteria into account are the primary challenges while defining the specifications (Knauss and Boustani, 2008). For this purpose, various checklists are available in literature (for example (Shea, 2017)).

However, a high quality of requirements alone does not necessarily lead to the success of a product development project (Tamai and Kamata, 2007). In addition, an essential task is to embed the relevant requirements in a suitable model, which can possibly change or need revision over time, in order to manage them and ensure their availability throughout the development process. In model-based systems engineering, the modeling of product requirements allows their use as a fundament for the entire product development process as well as their connection with elements from other product models (Walden et al., 2015). Linking those models with each other benefits the clarity of the entire system and generates knowledge about the connections within the system. For example, the relevant requirements can be linked with a product function and in the event of a requirement change, the influenced elements in the functional model can be identified. With this information, conclusions can be drawn about the necessary effort when changing elements in the requirement model (Goknil et al., 2014).

Requirements are not only the basis for the development activities itself, but also for the decision-making process (Regnell et al., 2001). Product requirements represent the boundary conditions in decision problems and have to be checked for fulfilment within the process of decision-making. Identified alternatives have to satisfy all requirements at a minimum. If they do not, they are not relevant for the decision-making process. Since several different criteria are used to identify the best alternative, the methods of multi-criteria decision-making are used for this purpose (Sen and Yang, 1998). In multi-criteria decision-making, decision problems can be classified according to different dimensions. Luft et al. (2015) provide a suitable method to characterize decision problems in product development and classify them, for example, according to their degree of difficulty or the composition of the solution space. When selecting an alternative in a given decision problem, all evaluation criteria have to be considered. Because of the stated connection with requirements, a link between the criteria model for decision-making and the requirements model is necessary. Breiing and Knosala (1997) stress that criteria specifications have to be defined while considering several quality criteria just as requirement specifications. Although the criteria model is based on the requirements model, both have to be set up and maintained individually. As a result, developers have to invest more effort and errors are more likely to occur. As stated by Regnell et al. (2001), bad decision may finally lead to products that do not fit for purpose and therefore decisions have to be supported by methods and tools. Less wrong decisions are essential for product success and may reduce the time spent on iterations concerning error corrections. Quality improvements of decision models are necessary in order to reduce the number of wrong decisions, especially when it comes to large projects with a high amount of information. The state of the art shows that there is currently no method or tool available to derive evaluation criteria based on product requirements, although Breiing and Knosala (1997) describe its necessity.

https://doi.org/10.1017/dsd.2020.52 Published online by Cambridge University Press

3. Research questions

Requirements are of high importance throughout the entire product development as well as evaluation process and therefore, a consistent derivation into evaluation criteria using a suitable method is essential. Such method is only practicable, if the additional effort to use does not exceed the benefits gained. As stated before, benefits of better decision-making are various, but the lack of applicable approaches leads to unused potential. Therefore, improving the overall quality of decision-models is central goal of our approach. Effects such as the reduction of iterations in development processes are secondary and need further investigation. Because of these considerations, the research questions to be answered in this paper are as follows:

- RQ1: How can requirements be linked with evaluation criteria in order to make them usable throughout the decision-making process?
- RQ2: Which techniques and methods can be used to ensure a consistent linkage between these two models and what manual modelling effort remains with the user?

4. Requirement classification for derivation of evaluation criteria

Throughout the product development process, developers are confronted by various decision situations. In multi-criteria decision-making, suitable evaluation criteria are required in order to compare the different alternatives and identify the best solution. As stated by Horber et al. (2019), derivation is possible through assigning the requirements to different classes (Figure 1). The relevant requirement classes and different types of evaluation criteria are explained within this section.

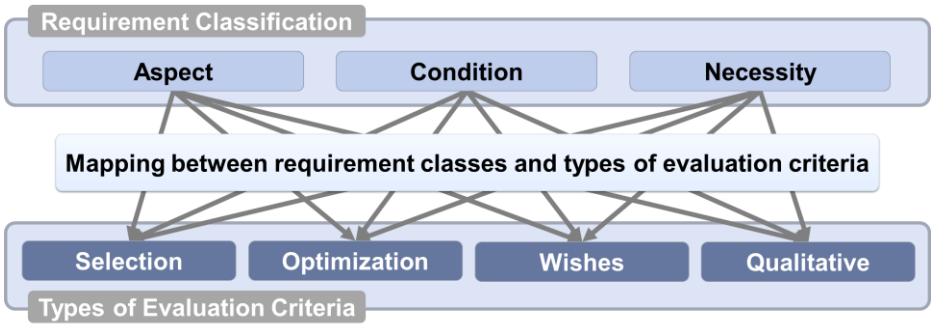

Figure 1. Concept to link requirements and evaluation criteria (Horber et al., 2019)

4.1. Requirement classes

In the requirement definition process, developers specify the way they limit the solution space, which is the main task of requirements (Rupp, 2014). Developers need knowledge of the different types of limitations to guide the development process as desired. Every linguistic specification therefore differs in the interpretation of these limitations. As stated from Horber et al. (2019), the requirement purpose can be represented by different classes according to their *necessity*, *aspect* as well as *condition* and later be used for the derivation of evaluation criteria.

4.1.1. Necessity

As described by Rupp (2014), a good requirement is always necessary and should never be optional. However, uncertainness about the relevance of single requirements can occur especially in early stages of product development. Therefore, a differentiation between *wishes* and *demands* is needed. *Wishes* do not have to be implemented under any circumstances. However, the implementation of a requirement classified as *demand* is necessary.

4.1.2. Aspect

The requirement specification can differ from the assigned type of value. Some requirements can only be specified by linguistic variables. These so-called *qualitative* requirements can therefore not be used in a

https://doi.org/10.1017/dsd.2020.52 Published online by Cambridge University Press

mathematical multi-criteria decision analysis without a corresponding quantitative value that represents the qualitative specification. Apart from that, *quantitative* requirements have a numerical value assigned. These values are either countable, measurable, weighable or calculable.

4.1.3. Condition

Another way to classify the requirements is by their function in the whole development process. On one hand, requirements can be seen as a *hurdle*, if fulfilling the requirement is necessary at a minimum, but the product does not benefit from a higher degree of fulfilment. On the other hand, there are *optimization* requirements. The better these requirements are met, the more the alternatives gain from it.

4.2. Types of evaluation criteria

For multi-criteria decision-making, evaluation criteria are necessary (Breiing and Knosala, 1997). These criteria differ fundamentally in their purpose within the decision-making process (Figure 2). For example, in terms of their evaluation character or their necessity. These types of evaluation criteria are used for the derivation from requirements in order to link the requirement and criteria model with each other. The identified basic types are explained in detail in this section.

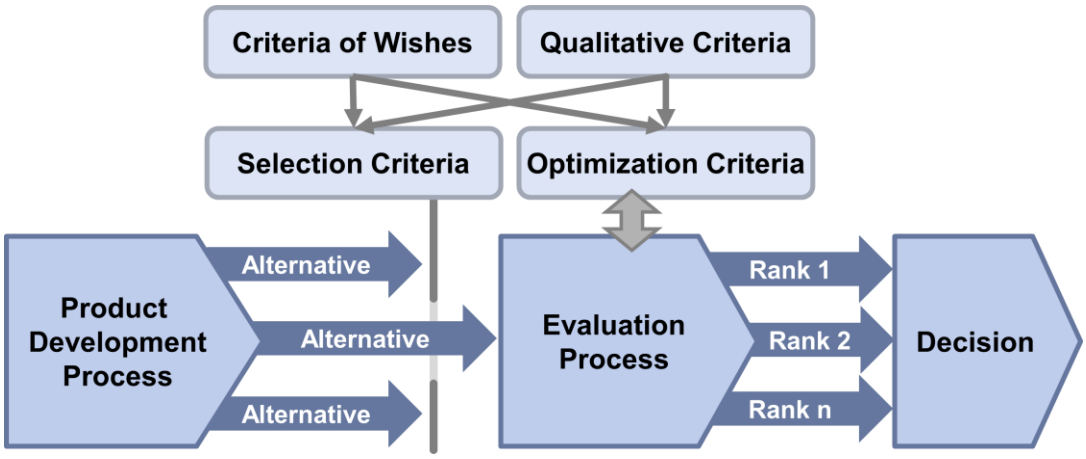

Figure 2. Types of evaluation criteria and their purpose in decision making

4.2.1. Selection criteria

The fulfillment is necessary, but overfilling these criteria does not lead to a higher ranking in the evaluation results. If these criteria are not met, the respective alternative is not taken into account in the next steps of the evaluation process, where the actual ranking is calculated.

4.2.2. Optimization criteria

After selecting only the relevant alternatives by checking the fulfilment of all selection criteria, the optimization criteria are then used for the actual ranking of the alternatives. A better degree of fulfilment by an alternative result in a higher rank of the respective alternative compared to those alternatives with a lower degree of fulfillment.

4.2.3. Criteria of wishes

This type of criteria represent those requirements that are classified as a wish. As stated before, wishes are optional and their implementation is not mandatory. Even though some authors like Breiing and Knosala (1997) recommend that wishes should not be considered in decision making, wishes have to be taken into account where necessary. Especially in early stages of product development, a certain opinion about the necessity of a requirement is not always possible. Therefore, wishes are included within the approach. In case of a criterion of wish, an argumentation why this wish is necessary has to be done. If a wish needs to be realized, the criteria has to be derived in either a selection or an optimization criterion.

https://doi.org/10.1017/dsd.2020.52 Published online by Cambridge University Press

4.2.4. Qualitative criteria

As stated by Rupp (2014), requirement specifications cannot be described by quantitative values in general. For example in early stages of product development, when there is lack of information. Quantification is achievable either in later stages, when enough information are gathered, or methods like value tables or fuzzy-logic are used (Rupp, 2014). To calculate the ranking of the alternatives with mathematical methods, a quantitative value is necessary. Qualitative criteria have to be derived in selection or optimization criteria, but with an explanation, how the quantification is done.

5. (Semi-) automated derivation of evaluation criteria

In the previous section, the basic requirement classes and types of evaluation criteria are described (see section 4). For the (semi-) automated derivation of evaluation criteria, a link between these two models is necessary. This derivation has to be explicit to ensure a consistent transition from requirements into evaluation criteria. As shown schematically in Figure 2, the derivation is on a mapping between both domains. In this section, this basic mapping is explained first and then the concept for the (semi-) automated derivation of evaluation criteria is introduced. Based on this concept, a demonstrator for an assistance system is then presented, which supports developers during the requirements definition process with an automated consideration of the requirement´s purpose as an evaluation criterion.

5.1. Mapping between requirement classes and evaluation criteria types

The requirement classes and criteria types described in section 4 have to be linked to ensure a consistent derivation of the requirements into evaluation criteria. A modified type of truth table is used for this purpose (Table 1), since truth tables are well suited for the representation of logical links (Posthoff and Steinbach, 2019). The table provides an summary of the mapping for the derivation. It is composed of the two areas, one mentioning the classes of requirements and one considering the types of criteria. These are subdivided analogously to the described contents in section 4. The mapping combines both areas and defines which combination of requirement classes lead to which type of criterion. Table entry ‘1’ visualizes the direct mapping, whereas indirect mapping is symbolized by entry ‘1’ in brackets.

Table 1. Truth table representing the mapping between requirement classes and evaluation criteria types (cells containing zeros are intentionally left blank because of better clarity)

Classes of requirements						Types of evaluation criteria			
Aspect		Condition		Necessity					
Quantitative	Qualitative	Hurdle	Optimization	Demand	Wish	Selection criteria	Optimization criteria	Criteria of wishes	Qualitative criteria
1		1		1		1			
1		1			1	(1)		1	
1			1	1			1		
1			1		1		(1)	1	
	1	1		1		(1)			1
	1	1			1	(1)		1	(1)
	1		1	1			(1)		1
	1		1		1		(1)	1	(1)

Based on the mapping done within the truth table, rules for deriving evaluation criteria from requirements can be developed by querying the information stored in the table. As can be seen within Table 1, the mapping is explicit for the selection criteria and optimization criteria. Criteria of wishes as well as qualitative criteria need a conversion either to selection or optimization criteria in a

https://doi.org/10.1017/dsd.2020.52 Published online by Cambridge University Press

subsequent step, as mentioned in section 4.2.3 and 4.2.4. In two cases, however, the mapping has two different indirect entries. Therefore, a one-time derivation is not sufficient in those cases. This problem has to be addressed during the development of rulesets. Otherwise, this could lead to an inconsistent model.

5.2. Ruleset for requirement derivation

In order to derive the evaluation criteria from the requirement model, the classification of the requirements can be used in conjunction with the mapping described in section 5.1. The mapping is used to develop the necessary rules for each type of criteria, which then leads to the derivation of the requirements. An exemplary pseudocode illustrates the functionality of these derivation rules, as in this case for deriving selection criteria:

```
For Requirement in Requirement_Dataset {
    If Aspect == 'Quantitative' AND Condition == 'Hurdle' AND Necessity == 'Demand'
        Criterion_Type = 'Selection Criteria'
    End If
}
```

The query for deriving optimization criteria is analogous to the query for selection criteria shown above. Differences exist when it comes to criteria of wishes and qualitative criteria. In these two cases, it is necessary to transform those into selection and optimization criteria in order to make them accessible to the decision-making process. For this purpose, a decision has to be made whether the wish has to be implemented or not (see section 4.2.3). Assessing a wish as no longer necessary leads to no transformation of the requirement and the status 'unnecessary wish' is applied. The requirement should yet be documented and stored for subsequent phases of the product development process in order to preserve traceability. The decision for or against a wish requirement should be done by an expert, because it is necessary to consider the effort, costs and benefits for the realization. The same applies to the quantitative criteria. These have to be converted into selection or optimization criteria as well. Here, however, the use of a suitable method for quantifying the characteristic is required. This enables the developers to evaluate the alternatives in a quantifiable manner within decision-making (see section 4.2.4). For example, the quantification of linguistic variable is possible by the help of a value table and the degree of fulfilment of each alternative is assessed by a user survey.

5.3. Conceptual model

As a main goal of this contribution, the defined research questions have to be answered (see section 3). Whereas section 5.1 deals primary with RQ1, this section examines the question about what a model for deriving evaluation criteria could look like. For this reason, a model (Figure 3) was developed based on the findings on the classification of requirements and the derivation of evaluation criteria. This provides developers an opportunity to transform the relevant requirements into evaluation criteria consistently. As a result, the product developer receives all the necessary criteria and can use them to rank the developed alternatives within the multi-criteria decision-making process.

The structure of the model is divided into steps for preprocessing requirements, classification and derivation into evaluation criteria. The starting point is the input of a natural language requirement, which is prepared within the framework of preprocessing with the help of algorithms out of the field of natural language processing (NLP). This outputs the preprocessed requirement, which is then prepared for the classification process. The classes are assigned to the requirement based on its semantic characteristics that are the outcome of the preprocessing beforehand. By using the mapping and the defined rule-sets for deriving requirements, the assessment of the individual types of evaluation criteria is done to the requirement. As mentioned in section 0, especially those requirements that are derived in wish or qualitative criteria have to be transformed with manual effort into selection or optimization criteria. At this point, there has to be argued whether the realization of a wish is necessary or how a qualitative requirement can be quantified for multi-criteria decision-making. Afterwards, the criteria model is used for the decision analysis on which alternative is the best and which one is to choose.

https://doi.org/10.1017/dsd.2020.52 Published online by Cambridge University Press

The requirements are therefore consistently derived in evaluation criteria and this enables developers to receive feedback on the purpose of each requirement in the decision-making process based on the natural language specification (Figure 2). The direct connection of the requirement and evaluation criteria model is the main advantage of the concept. Due to this, changes in the specification of single requirements automatically lead to a corresponding change in the evaluation criterion. In addition, new requirements are derived (semi-) automatically, therefore there is no need for a separate process, which reduces the workload. Thus, the product developer has to maintain only one model, which reduces the effort considerably. Nevertheless, decisions can be made based on all relevant requirements and evaluation criteria. This reduces the risk of forgotten requirements, which leads to a higher quality of the decision model as well as less wrong decisions.

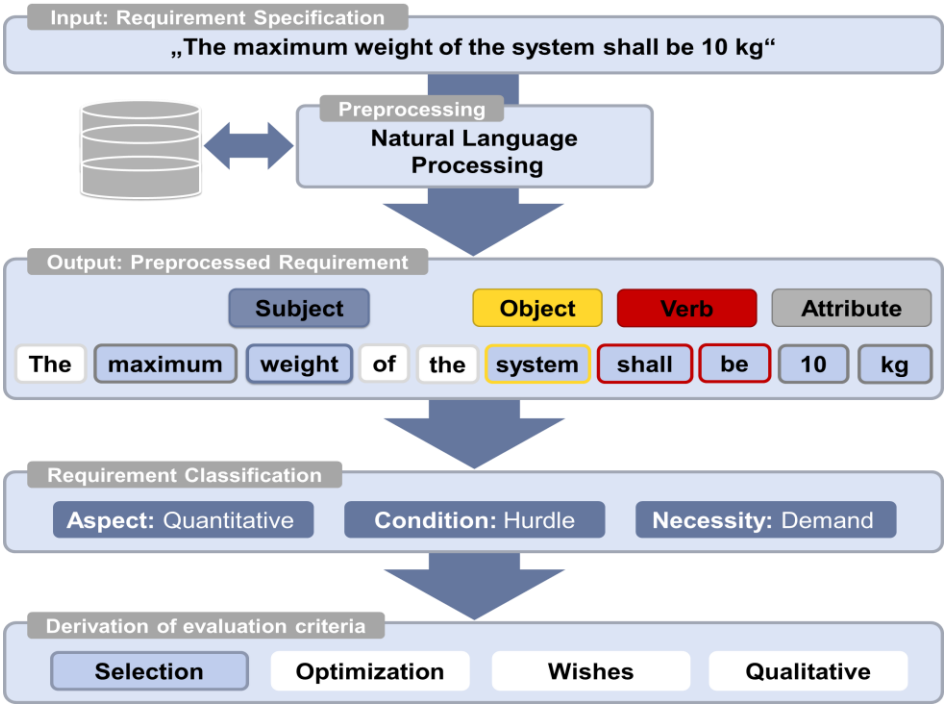

Figure 3. Model for (semi-) automated derivation of requirements into evaluation criteria

However, the model requires an efficient and explicit classification of the requirements. This is accomplished by using the classes defined as aspect, condition and necessity and can be done with the help of the NLP-algorithms. The application of NLP in the field of RE is getting more popularity by time and is used in this context to extract elements of concerns out of requirement statements (Nazir et al., 2017). An overview about the current state of the art can be found, for example, in Meth et al. (2013). NLP can be a suitable method for supporting even experienced developers in their work with natural language requirement statements (Ryan, 1993) as well as in early stages of the development. This reinforces the idea that the model presented in this contribution can also be applied at the beginning of product development process. The concept for preprocessing requirements with the help of NLP is based on the segmentation of natural language requirement specifications into several parts of the sentence, which are then used to classify the requirements. A review by Binkhonain and Zhao (2019) shows that the most commonly used techniques of NLP considering the described kind of task are stemming, stop word removing, part-of-speech tagging, tokenization and lemmatization. Concerning the presented approach, the mentioned techniques have to be used within the preprocessing and classification model where applicable, in order to ensure the performance and accuracy of the classification. This is necessary for improving the quality of decision models.

5.4. Demonstrator

The consistent derivation of evaluation criteria is essential for a holistic multi-criteria decision analysis based on all product requirements. However, the model developed and presented makes this accessible

https://doi.org/10.1017/dsd.2020.52 Published online by Cambridge University Press

for developers. To check the applicability of the concept, a demonstrator was developed with the basic functions implemented (Figure 4).

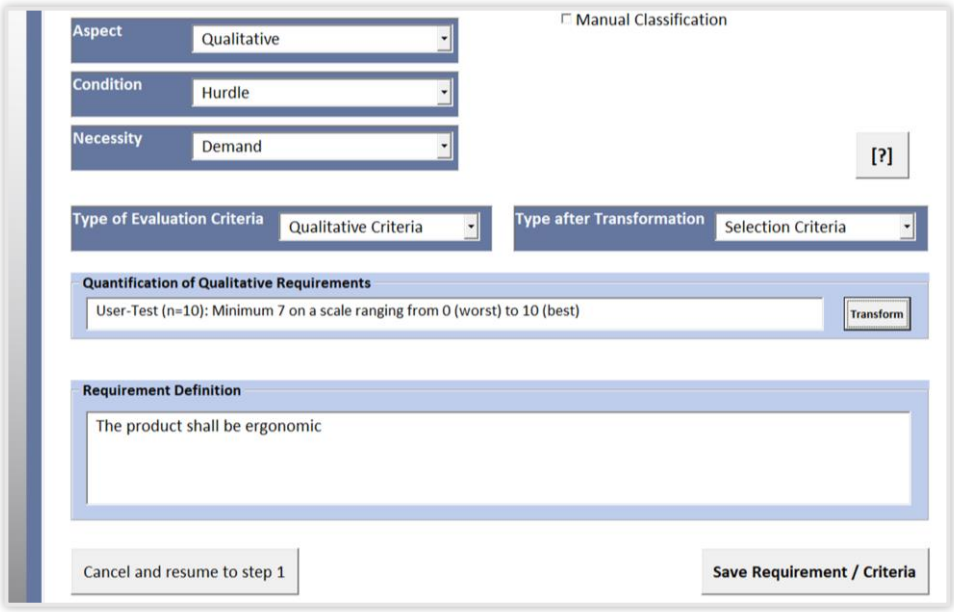

Figure 4. Demonstrator for the concept of deriving evaluation criteria from requirements

The demonstrator serves as a tool for the continuous definition and modification of requirements. Product developers can enter their requirements in the appropriate field. While typing in the requirement specification, the requirement is iteratively preprocessed and classified with the help of NLP. Developers receive immediate information about the classification that was done based on their input. By applying the mapping rules, the assigned type of evaluation criteria is also displayed. Consequently, developers obtain an instant feedback concerning the purpose of the specified requirement in later decision-making situations. For a holistic decision-making based on every relevant product requirement, all of them have to be either selection or optimization criteria (see Figure 2). If the tool recognizes a criterion of wishes or a criterion that is difficult to quantify, another input field opens up. In here, developers have to provide appropriate information whether a wish should be implemented or not or how to quantify a qualitative criterion. After entering this information, the use of the 'transform' function initiates a new classification request. The corresponding type of criteria is then assigned to the requirement. If a wish is not going to be implemented, the status 'deleted' is noted. It is also possible for developers to classify the requirement manually. However, only experienced users should use this option in order to avoid errors or inconsistencies in the criteria model.

The exemplary requirement statement in Figure 4 requires a fictitious product to be ergonomic. The tool first classifies this requirement as a qualitative requirement, since this linguistic expression is not quantified. In addition, it is assigned to the class 'hurdle' regarding its condition. Therefore, only its fulfilment or non-fulfilment is checked during the multi-criteria decision-analysis. Because of the word 'shall', the tool also recognizes a 'demand'. Due to its qualitative specification, a quantification of the is required. In case of the displayed example, a user test with a sample size of 10 is required in order to achieve the quantification. When interviewing users that have used the product, they have to assess its ergonomics on a scale ranging from zero to 10. A median score of at least seven has to be reached to successfully meet this criterion. The requirement can then be derived again and is finally transformed into a selection criterion.

5.5. Discussion

Reliable and good decisions in multi-criteria decision-analysis require the consideration of all evaluation criteria within a development project. These criteria have to correspond with the requirements, which have been specified at the beginning of the development process and those, which have been developed

https://doi.org/10.1017/dsd.2020.52 Published online by Cambridge University Press

later as well as extended or modified over time. In the worst case, forgotten requirements and therefore also evaluation criteria can lead to wrong decisions. The presented model contributes to solve this problem as it enables the consistent derivation of evaluation criteria from the requirement model.
Along with the demonstrator, it was shown that the concept could also be implemented in a software tool that supports developers' work considering requirements and evaluation criteria throughout the entire development process. The instant feedback of the requirement's purpose in decision-making enables a change in thinking about writing requirement specifications. If this function of the demonstrator is additionally extended to the calculation of a quality score (see also (Jergus, 2019)), the quality of a requirement specification can already be ensured in the definition phase and the consistent derivation of evaluation criterion is supported. The product developer does no longer have to check the quality of the requirement statement in a subsequent quality check with a checklist. This increases the overall quality of the requirements model, by which the entire development process benefits (Tamai and Kamata, 2007). Furthermore, the effort for developers is reduced, because of the (semi-) automated derivation of the evaluation criteria model mainly the requirement model has to be kept up to date.
Thus, the demonstrator only shows the basic functions of the concept so far since the implementation of the NLP process is only rudimentarily. At this point, it is necessary to invest further research effort. The application of NLP requires a well-trained model to ensure a high performance of the classification. Therefore, it is important to train it with suitable datasets. Nevertheless, potential errors may occur by NLP and thus incorrect classifications may be possible. In our approach, developers who define requirements need knowledge about the different types of criteria as well as their purpose in decision-making. If a requirement is incorrectly classified, this contradicts the intended purpose by the developer. As a result, either a revision of the requirement definition or a manual classification is necessary. In case of wrong classifications, this manual effort remains with the user. The main objective of the presented approach is the quality improvement of decision models. As stated, secondary effects, such as positive impacts on the number of iterations and therefore saving time in projects, need further investigation.
Until now, only the functionality of the tool has been checked. For this reason, a general user test is essential to evaluate its usability. The demonstrator itself covers the process from the input of a requirement specification to the output of a respective evaluation criterion. Regarding the multi-criteria evaluation process, additional functions are conceivable to support developers in even more activities concerning decision-making. For example, a functionality to add mathematical value functions to every criterion already within the framework of requirement specification.

6. Conclusion and future work

A conscientious handling of requirements is essential for successful product development. Forgotten requirements may lead to products, which are not concordant to the users' needs or other stakeholders' expectations. Especially when using requirements as evaluation criteria within decision-making, these can result in wrong decisions. Therefore, a concept was presented in this contribution, which enables the consistent derivation of evaluation criteria from the requirements model. The derivation is realized (semi-) automated, because there is still a manual effort for the transformation of wish and difficult to quantify criteria into selection and optimization criteria necessary. The basic functionality of the concept was then verified with a developed demonstrator and confirmed with first tests.
At the beginning of this paper, we elaborated the research questions based on the state of the art. These questions deal with the linkage of the requirement model and criteria model (RQ1) on one hand and on the other hand with the realization of a suitable model to ensure this derivation (RQ2). The first question can be answered with the help of the mappings, which connect the defined requirement classes with the different types of evaluation criteria and enable the derivation. To answer the second question, the overall concept was introduced in detail, which extends the connection between the two models by preprocessing the requirements statements as well as the (semi-) automated derivation.
Finally, a demonstrator was presented, which illustrates the basic functionality of the concept and the overall method was critically discussed. In the future, an evaluation of the tool with the help of user tests will be necessary to receive a better impression of its usability. An open point is the further development of the classification algorithm with NLP, as it is currently implemented only rudimentarily.

https://doi.org/10.1017/dsd.2020.52 Published online by Cambridge University Press

Acknowledgement

This research and development work is funded by the German Research Foundation (DFG). The authors gratefully acknowledge the financial support of project WA 2913/33-1.

References

Binkhonain, M. and Zhao, L. (2019), "A review of machine learning algorithms for identification and classification of non-functional requirements", *Expert Systems with Applications: X*, Vol. 1 pp. 1-13. https://doi.org/10.1016/j.eswax.2019.100001

Breiing, A. and Knosala, R. (1997), *Bewerten technischer Systeme: Theoretische und methodische Grundlagen bewertungstechnischer Entscheidungshilfen*, Springer, Berlin/Germany. https://doi.org/10.1007/978-3-642-59229-4

Fernandes, J.M. and Machado, R.J. (2016), *Requirements in engineering projects, Lecture notes in management and industrial engineering*, Springer, Cham/Switzerland. https://doi.org/10.1007/978-3-319-18597-2

Goknil, A. et al. (2014), "Change impact analysis for requirements. A metamodeling approach", *Information and Software Technology*, Vol. 56 No. 8, pp. 950-972. https://doi.org/10.1016/j.infsof.2014.03.002

Horber, D., Schleich, B. and Wartzack, S. (2019), "Ein Klassifizierungssystem zur Anforderungssystematisierung", *Proceedings of the DfX 2019 / 30th Symosium Design for X* , Jesteburg, Germany, September 18-19, The Design Society, Glasgow, pp. 227-238. https://doi.org/10.35199/dfx2019.20

Jergus, D. (2019), *Presentation on Requirements Quality Assistant (RQA) - Intelligent Requirements Management with IBM Watson*, Requirements Engineering Conference 2019, Munich.

Kamata, M.I. and Tamai, T. (2007), "How Does Requirements Quality Relate to Project Success or Failure?", *Proceedings of the RE 2007 / 15th IEEE International Requirements Engineering Conference*, New Delhi, India, October 15-19, IEEE Computer Society, Los Alamitos/USA, pp. 69-78. https://doi.org/10.1109/RE.2007.31

Knauss, E. and Boustani, C.E. (2008), "Assessing the Quality of Software Requirements Specifications", *Proceedings of the RE 2008 / 16th IEEE International Requirements Engineering Conference*, Barcelona, Spain, September 08-12, IEEE Computer Society, Los Alamitos/USA, pp. 341-342. https://doi.org/10.1109/RE.2008.29

Luft, T., Schneider, S. and Wartzack, S. (2015), "A methodical approach to model and map interconnected decision making situations and their consequences", *Proceedings of the ICED 15 / 20th International Conference on Engineering Design*, Milan, Italy, July 27-30, The Design Society, Glasgow/Scotland, pp. 329-340.

Meth, H., Brhel, M. and Maedche, A. (2013), "The state of the art in automated requirements elicitation", *Information and Software Technology*, Vol. 55 No. 10, pp. 1695-1709. https://doi.org/10.1016/j.infsof.2013.03.008

Nazir, F. et al. (2017), "The Applications of Natural Language Processing (NLP) for Software Requirement Engineering - A Systematic Literature Review", *Proceedings of the ICISA 2017 / 8th iCatse International Conference on Information Science and Applications*, Macau, China, March 20-23, Springer, Singapore/Malaysia, pp. 485-493. https://doi.org/10.1007/978-981-10-4154-9_56

Posthoff, C. and Steinbach, B. (2019), *Logic Functions and Equations: Binary Models for Computer Science*, Springer, Cham/Switzerland. https://doi.org/10.1007/978-1-4020-9595-5

Regnell, B. et al. (2001), "Requirements Mean Decisions! Research issues for understanding and supporting decision-making in Requirements Engineering", *Proceedings of the SERP 01 / First Swedish Conference on Software Engineering Research and Practice*, Ronneby, Sweden, October 25-26, Blekinge Instiute of Technology, pp. 49-52.

Requirements Working Group (INCOSE) (2017), *Guide for Writing Requirements*, International Council on Systems Engineering, San Diego/USA.

Rupp, C. (2014), *Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil*, 6th edition, Carl Hanser, Munich/Germany.

Ryan, K. (1993), "The role of natural language in requirements engineering", *Proceedings of the RE 1993 / IEEE International Symposium on Requirements Engineering*, San Diego, USA, January 04-07, IEEE, Los Alamitos/USA, pp. 240-242. https://doi.org/10.1109/ISRE.1993.324852

Sen, P. and Yang, J.-B. (1998), *Multiple Criteria Decision Support in Engineering Design*, Springer, London/England. https://doi.org/10.1007/978-1-4471-3020-8

Shea, G. (2017), *NASA Systems Engineering Handbook*, 2nd edition, National Aeronautics and Space Administration, Washington.

Tamai, T. and Kamata, M.I. (2007), "Impact of Requirements Quality on Project Success or Failure", *Proceedings of the Design Requirements Workshop, Cleveland*, USA, June 03-06, Springer, Berlin/Germany, pp. 258-275. https://doi.org/10.1007/978-3-540-92966-6_15

Walden, D.D. et al. (Eds.) (2015), *Systems engineering handbook: A guide for system life cycle processes and activities*, 4th edition, Wiley, Hoboken/USA.

https://doi.org/10.1017/dsd.2020.52 Published online by Cambridge University Press