Towards Requirements Analytics: A Research Agenda to Model and Evaluate the Quality of Unstructured Requirements Specifications

Patrick Kummler(✉)

Karlsruhe Service Research Institute (KSRI), Karlsruhe Institute of Technology (KIT), Kaiserstraße 89, 76133 Karlsruhe, Germany
patrick.kummler@kit.edu

Abstract. Communication between actors in a service system can be based on unstructured text. The quality of this text is crucial for the effort and output of a service system. The paper presents an approach to evaluate and model the quality by using requirements from automotive development projects as practical example. The aim is to define quality by using relevant attributes and quantifiable measures. First results include the development of an assessment tool and an initial analysis of the available dataset.

Keywords: Service systems · Software requirements · Analytics · Machine learning · Automotive software · Requirements quality · Quality metrics · Quality measurement

1 Introduction

The main economic activities in developed countries are service activities [1, 2]. The increase of the service sector is based on the tremendous outsourcing of activities, such as maintenance or tasks in research and development. According to Vargo and Lusch [3], a service is defined as "*the application of competences for the benefit of another, meaning that service is a kind of action, performance, or promise that's exchanged for value between provider and client*." In contrast to the goods-dominant logic, a service activity is based on communication and exchange with the client: "*[T]he more knowledge-intensive and customized the service, the more the service process depends critically on client participation and input"* [4, p. 72].

The main goal of a service system is to create reciprocal value – "service is exchanged for service" [1]. Therefore, a service system concludes the co-creation of value through the configuration of actors and resources. An actor is defined as a person including their knowledge and skills while resources are described as technology, information and physical artifacts. Thus, a service system is a *"socio-technical system that enables value co-creation guided by a value proposition"* [5, p. 399].

Service systems can increasingly be observed in different industries. For example, in the development of automotive software products the actors of a service system can be recognized as automotive manufacturers and suppliers. In this case the service

© Springer International Publishing AG 2017
S. Za et al. (Eds.): IESS 2017, LNBIP 279, pp. 197–209, 2017.
DOI: 10.1007/978-3-319-56925-3_16

interactions are based on communication and exchange of information and requirements between these actors.

Generally, services increasingly influence the automotive industry [6–8]. The importance of configurable services as part of current and future mobility services increases the relevance of effective co-creation of value between car manufacturers and suppliers. The creation process of service solutions, that are provided by developing software-based functionalities in products, is crucial for future competitiveness. Besides the importance of these services, the complexity of software products is growing [9, 10]. Several established quality approaches exist to describe and specify software requirements [11]. However, industrial practice is still suffering from a lack of structure regarding the interaction and cooperation between actors in a development project.

In the development process of a software product the specification of requirements is the first concrete and tangible output [12]. However, it is also the most critical part as the specification is the major source of failures for upcoming activities [13]. At the same time the specification and adjustment of requirements is an ongoing and highly agile process [14]. Thus, it must be ensured, that the quality of requirements is on a high level from the beginning of a software project.

In this work, we develop an approach to analyze communication and exchange of information based on software requirements that are specified as unstructured text by using machine learning algorithms. The target is to implement an automated solution to measure and monitor requirements quality. We contribute at two stages during the specification process: First, the quality of the specification can be checked and analyzed at the point of creation and improvements are shown directly; second, the output can be evaluated from different roles.

We portray the research in cooperation with an international automotive engineering company and relevant organizational units of car manufacturers. The research should not only improve effectiveness and efficiency of software-based services in the automotive industry. It should also offer a generalizable solution where the analysis and improvement of unstructured text is necessary for the successful communication between actors of a service system.

The quality assurance is based on the use of metrics that are analyzing the requirements. Before metrics can be applied to the text, the term *quality* must be defined and concretized. Many definitions of quality with regards to software requirements are already established [15–17]. The work in progress analyzes and aggregates these definitions and enhance the approaches with the input from experts, that are creating and receiving software requirements either for testing or implementing. The input is used in different phases of our research. An assessment phase delivers necessary data to get insights into the qualitative assessment of requirements. Based on these results we develop quantitative measures as input for machine learning techniques.

2 Related Work

The analysis of software requirements was initially conducted in the aerospace sector [12]. Due to high safety and legal provisions, software developers were forced early to guarantee reliable results. The experiences from the aerospace sector for software development should be considered in the automotive industry as the automotive manufacturers are increasingly faced with similar challenges [18, 19].

General problems, solutions and challenges of the requirements engineering are summarized and categorized in [20]. The authors analyze and evaluate different software projects in the automotive industry and indicate various results for the following categories: general issue, requirement engineering process and technical issues. These results can be used as input for the development of relevant quality attributes.

In [21], a framework to measure and improve the quality of textual requirements is presented. Especially the challenge regarding the derivation of quantitative measures from quality attributes is considered. Similar approaches of the requirements quality prediction are based on this research. However, despite sufficient approaches defining relevant quality attributes of software requirements, the industrial practice is rarely asked about the relevance and importance of the quality attributes and the assignment of quantitative measures.

Parra [22] build a machine learning technique to assess the quality of requirements automatically. The authors use the tool RQA (Requirement Quality Analyzer) [21] to provide quantitative criteria. They show that their methodology manages to evaluate the quality of requirements written in natural language in the same way an expert would do with 86.1% average accuracy. Their work focuses on correctness metrics. Other kind of metrics are left aside which offers the possibility for further and comprehensive research.

Mund [17] follows a similar research approach and investigates the influence of requirements quality in the lifecycle of software-intensive systems. This research and the subsequent contribution offer useful information especially regarding the derivation of quantitative measures from quality attributes.

Davis [15] provides a list with different quality attributes for requirements. We plan to enhance this list with current research. We analyze different approaches of quality attributes referring to requirements quality and compare and aggregate the information. In this context, we focus especially on weak phrases used in the specification of requirements [23].

3 Research Methodology

The aim of our research is to analyze and improve service interactions based on unstructured text for the co-creation of value. By using service interactions between automotive manufacturers and suppliers we want to develop an automated solution that analyzes, measures and monitors software requirements to support the specification process in a software project.

A literature review according to vom Brocke [24] helps us to systematically find relevant literature. Although a literature review is described as *"a summary of a subject field that supports the identification of specific research questions"* [25, p. 31], we already defined potential questions. Our research topic is based on personal experiences made during the interaction and collaboration with organizations and enhanced by analyzing relevant practice literature [26, 27]. In Webster [28] it is mentioned, that such experience helps to justify and support a proposition. In the following we propose our research questions:

RQ1 How can the quality of unstructured text improve the communication between actors in a service system?
RQ2 How is it possible to measure the quality of unstructured text by using qualitative attributes and quantitative measures?
RQ3 How is the quality of unstructured software requirements defined?
RQ4 Which quality attributes are relevant for the interaction between actors and which of them are consciously adapted in the development of software functions for the automotive industry?
RQ5 Which quality attributes influence the quality perception?
RQ6 Is it possible to automatically recognize the quality of unstructured requirement text?

The proposed research questions are focused on the quality of unstructured text. Thus, it is necessary to propose a resilient definition of the term quality and to evaluate it throughout the research progress. RQ1 analyzes the relation between different actors of a service system where the quality of unstructured text communication is a crucial part. We analyze how the communication quality can be improved by considering the relevant unstructured text.

To increase the communication quality in a service system different quality metrics are necessary. In the second research question, we evaluate and implement quality attributes that analyze unstructured text. Established approaches are presented and applied to the available data. The implementation of enhanced attributes is also a relevant part.

In a specific way, RQ3 gives an overview of existing quality notions regarding software requirements. For the term quality, especially in conjunction with software requirements, many different definitions exist. In our research, we want to aggregate these definitions and develop a comprehensive approach by considering input from expert interviews as well.

RQ4 is concerned about which quality attributes are relevant for the communication and interaction between actors in a service system. Referring to our practical example, we consider different actors of software development projects in the automotive industry. This research question studies which quality attributes from literature and standards are consciously adapted in the industrial practice.

Following the previous research question, RQ5 aims to identify the relevant quality attributes that influence the quality perception. The goal is to analyze quality attributes that are relevant for the actors in a service system.

Based on the previous results RQ6 questions whether an automatically recognition of requirement quality is possible. By gathering sufficient data, we want to answer this question using machine learning techniques to predict the communication quality.

In the field of quality measurement for requirements, a lot of earlier work exists. Due to the number of similar approaches regarding the quality characteristics of software requirements, such as IEEE Std. 830 [29] or ESA PSS-05 [30], it is planned to create an overview and compare the different concepts later in the research. Our research builds upon previous findings and past research and enhances these results with input from experts. This information is used to adapt an approach that focus on industrial practice.

We plan a systematic and guided literature review defined as a *"structured approach to identifying, evaluating and synthesizing research [and to] address a specific research question that guides data collection, extraction and aggregation process"*. [31, p. 9] The purposes of our review are to comprehensively understand the domains of requirements engineering, quality measurement and machine learning techniques. Additionally, we want to uncover research and feature gaps for the measurement of requirements quality. A relevant part is the identification of research methods and strategies that are commonly used for the quality measurement of requirements.

Following vom Brocke [24], we focus on different phases during our literature review. As the goal of the review should be clear in the beginning we adopt the taxonomy according to Cooper [32] in the initial phase. We start with analyzing handbooks and seminal literature about requirements engineering [26, 27]. The next phase focuses the relevant database, keywords and back- and forward search as well as an ongoing evaluation of sources. It is planned to analyze relevant journals and doing an author- and topic-based search simultaneously. We evaluate important IS conferences [33, p. 121] for similar topics and research approaches. After sufficient literature [31] is collected we analyze and synthesize relevant content into a concept matrix to evaluate relevant concepts.

In our research, we consider the Cross-Industry Standard Process for Data Mining (CRISP-DM) as standard reference model for our data mining approach [34]. Initially we start with the business understanding phase and define objectives for our data analysis. As we already have relevant data for our research topic we proceed with activities that *"enable [us] to become familiar with the data, identify data quality problems, discover first insights into the data, and/or detect interesting subsets to form hypotheses regarding hidden information"* [34, p. 10]. The initial analysis of our dataset is presented in Sect. 6.2 and shows basic syntactical measures. This analysis helps us to get familiar with the available data and to identify potential tasks regarding the pre-processing of our dataset. From the final dataset, we randomly choose requirements for the assessment tool. For the modelling phase, we choose the application of a support vector machine (SVM). According to Kivinen [35], SVM algorithms are well suited for problems with dense concepts and sparse instances. Moreover, SVM have the ability to learn independently from the dimensionality of the feature space. *"[SVMs] allow fully automatic parameter tuning without using expensive cross-validation"* [36, p. 3]. Additionally, we follow the approach of Hsu et al. [37] that offers a practical guide for support vector classification. Nonetheless, other techniques are considered to analyze the optimal approach according to the dataset.

Simultaneously to the literature review we conduct exploratory interviews with different roles: requirements engineer, developer and tester. A requirements engineer is the person that specify the requirements. This person creates the unstructured text we want to analyze and improve. A developer or a tester receive requirements and capture the unstructured text. We conduct the interviews by meeting our participants "face-to-face" [38]. A group basis is not desired, as the single participants could be biased by the answers of other participants especially regarding the questions about quality. As there are various persons with different years of profession we prefer a semi-structured interview as this type refers to qualitative research and is non-standardized [39].

With our research, we contribute in different parts of relevant scientific areas. We analyze existing approaches that assess the communication quality based on unstructured text. We use input from experts' perspective to enhance these approaches and aim at creating a tool that automatically analyzes text according to defined quantitative measures. Our solution helps to reduce misunderstanding between actors and reduces unnecessary exchange of information due to inadequate quality. We conduct the quality analysis of the unstructured text by using quality attributes acknowledge from the experts. Additionally, we conduct a structural analysis to find quantitative measures representing a well-defined requirement. By using machine learning algorithms, we want to define appropriate rules that classify requirements automatically.

4 Research Process

The research is planned in three phases: Quality Definition, Assessment & Analysis and Tool Development. Figure 1 shows the research process.

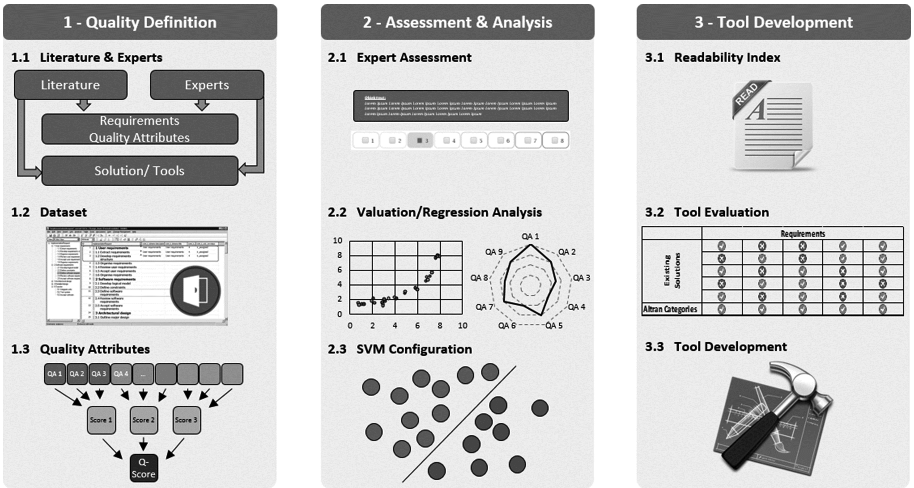

Fig. 1. Research process for requirements analytics

The first phase *Quality Definition* includes tasks to gather data and create attributes to measure the quality of unstructured text in requirement documents. We gain information about the quality measurement of requirements by considering established attributes from literature and standards. Additionally, we plan to conduct around 20 expert interviews to gain enhanced information and best practices about necessary quality attributes and quantitative measurements. The experts for the interview sessions have different roles in software projects (requirements engineer, software developer and tester). Simultaneously we capture data consisting of unstructured requirements that are used for the interaction and communication between actors of the service system. The next task covers the definition of relevant quality attributes. The outcome of the first phase contains an overview about established quality attributes and insights about necessary characteristics of qualitative requirements in the industrial practice. Besides gathering relevant data, the first phase provides a set of defined and crucial quality attributes. This step is the most important and challenging part of the research as the following phases are based on its output.

During the *Assessment & Analysis* phase, we gather additional data through the assessment of requirements to analyze the quality in two ways: first, we want to determine which quality attributes influence the overall quality perception of requirements; second, we want to implement a solution to predict the quality of requirements by using machine learning techniques. The first task in this phase contains the tool development to assess the unstructured requirement text with the defined quality attributes from the previous phase. After reaching a level of saturation - when more assessments would not introduce significantly new findings and insights [31] - the results are used to determine dependent quality attributes by analyzing the correlation for each quality attribute. We want to identify relevant quality attributes and the influence of each attribute regarding the overall quality assessment of a requirement.

The following task includes the derivation of quantitative measures from the quality attributes defined in the phase *Quality Definition* by using the input from the expert interviews as well. With these measures, it is possible to implement a support vector machine (SVM) that classifies the quality of unstructured document text.

The results of the second phase contain the assessed requirements and the application of a regression analysis which is used to gain detailed insights about the data. To predict the quality of requirements, quantitative measures are derived and assigned to each relevant quality attribute. In this way, a support vector machine analyzes and uses the data as input to predict requirements quality.

In the last phase (*Tool Development*), the tool evaluation and implementation is focused. In the first task, we consider and analyze different readability indices. It is planned to adapt syntactical features to implement and evaluate an approach for a software requirement readability index. Based on already established readability indices, such as Flesch, Dale-Chall and SMOG, a similar approach would help to identify the complexity and understandability of a software requirement. An evaluation of available tools for the qualitative measurement of software requirements based on features and metrics is necessary. The last task in this phase contains the development of a solution based on evaluated features to improve the quality of software requirements.

5 Research Issues

The analysis of textual requirements can be an *"aid for writing the requirements right, but not for writing the right requirements"* [21, p. 26]. A measurement can be an approach to improve the correct understanding of requirements. However, an overall challenge in the analysis of requirement text is to define attributes that offer insights about the content quality. In [21], the quality measurement is mainly based on the analysis of countable characteristics. Generally, a quantitative measurement approach does not necessarily give insights about the content quality as the text is only the carrier of the content. However, we are convinced that understandable, clear and unambiguous content lead to a better communication between actors and, referring to our automotive example, lead to a higher quality of software products.

In our research, requirements are assessed without detailed context. This could lead to a different analysis from experts and thus the creation of distorted rules from our machine learning algorithm. However, we believe that intelligibility is one important criterion on its own as well. Our methodology is dependent on the quantitative attributes that we define to analyze the requirements. The definition of the right and accurate metrics is a crucial point for the success of our predictions. On the other hand, our research method depends a lot on experts' inputs and evaluation. Given the numerous features in a text analysis, it might take a while to reach the saturation level of our algorithm. A challenge is also to find enough experts that are willing to share their time and knowledge for these analyses.

The development of this tool is based on several hypotheses: quality can be made objective and measurable, metrics can portray experts' quality analysis and metrics are sufficient to interpret natural language. We refer to empirical results to confirm these hypotheses. Moreover, we plan to use one set of chosen requirements to develop this tool, which might not be sufficient to apply the deduced classifying rules to every other requirement.

As mentioned in [17] another possible issue is the unawareness of related work. Metrics, its application and natural language analysis is a broad subject. Thus, we might be tempted to use others disciplines hints and at the same time we might miss some important inputs from distant fields we are unaware of.

6 First Results

In this section, we present first results regarding the development of the assessment tool and the analysis of the dataset.

6.1 Development of Assessment Tool

The assessment tool is constructed as a web survey using JavaScript and PHP scripts connected to a SQL database. Initially general information is asked: gender, role in the service system and years of experience in this role.

After a necessary data preprocessing, a random set of requirements is selected for the assessment. We want to make sure that the requirement set is evaluated several times by different roles. Thus, we take the given role as input for our SQL queries to display the requirements to be analyzed.

For each text, the expert first decides whether it is a requirement, an information or another type (e.g. headings to structure the requirements document) that is used in the specification process of requirements. In case *requirement* is selected, the assessor can detail the concrete requirement type according to the V-model [40]. This standard development process for software development offers different requirements types in each development phase. However, despite of established procedures and standards to create different documents for each requirement type, industrial practice is still suffering from a lack of clear separation between these types. Therefore, and due to reasons of simplicity requirements engineers often combine different requirement types in one document. This situation urges to ask about the concrete type in the assessment phase, such as customer requirements, architectural requirements, (non-) functional requirements or software requirements. After selecting the type, the expert assesses the overall quality of the requirement and the quality attributes that are defined in the first phase of our research process. The assessments are saved in our database through SQL queries.

6.2 Data Analysis

The assessment tool provides information about how requirements are perceived by the experts and leads to the assessment of requirements based on defined quality attributes. Besides this, we analyze objective information about the available dataset. We plan to define metrics based on lexical indicators (connective terms, imprecise terms) and analytical indicators (verbal tense) [20].

The available dataset is based on an export from several development projects and contains 32.956 object types in 54 requirements documents. Objects types are defined as headings, information and requirements. As headings are not relevant for our analysis we exclude this type from the dataset. The remaining object types are enhanced with additional information, such as language of the object text and a differentiation between software and hardware related content. Further we reduce the dataset by excluding empty content and double spaces in the relevant object text to 26.492 objects. We analyze the dataset by defining the following basic syntactical measures:

– Number of characters
– Number of words
– Average length of words

In Table 1, the analysis of the number of characters reveals that there is a remarkable difference between software and hardware related content. The mean value of hardware referring to the number of characters is 121.99 whereas for software it is 138.56. Such characteristics can also be observed in the standard deviation where the value for software content (120.68) is one third higher than for hardware content (88.32). The values for minimum and maximum number of characters lead to the necessity to invest sufficient time for a successful pre-processing of the data. As example, an object (requirement or information) consisting of only ten characters can hardly transport sufficient information. In contrast, for an object with 3900 characters it should be possible to split the content in several objects. Throughout the measures software objects contains more characters in comparison to hardware objects.

Table 1. Analysis of number of characters

Objects	Mean value	Standard deviation	Minimum value	Maximum value
Hardware	121.99	88.32	10	1876
Software	138.56	120.68	18	3900

In Table 2 the results from the analysis of number of words are shown. The mean value is similar for hardware (18.02) and software (19.50). However, the outliers for software objects (650) have to be analyzed separately. The analysis of the average length of words in Table 3 confirms the necessity of pre-processing before using the dataset for further research. The mean value for the average length of words is similar for hardware (6.00) and software (6.29). The standard deviation for both object categories corresponds as well (1.29 for hardware and 1.56 for software). Again, the software related content reveals outliers referring to the maximum value of the average length of words (hardware with 17.58 and software with 24.81).

Table 2. Analysis of number of words

Objects	Mean value	Standard deviation	Minimum value	Maximum value
Hardware	18.02	12.95	5	166
Software	19.50	16.00	5	650

Table 3. Analysis of average length of words

Objects	Mean value	Standard deviation	Minimum value	Maximum value
Hardware	6.00	1.29	1.20	17.58
Software	6.29	1.56	2.00	24.81

The first analysis of the dataset leads to the necessity to invest sufficient time in pre-processing the data in order to create a qualitative dataset. This analysis shows the range of outliers and differences between hardware and software related content and contains first results to get familiar with the available dataset. Additional tasks are necessary and presented in the work in progress.

7 Conclusion and Outlook

The paper sets up to improve communication quality between actors in a service system. We refer to the relationship between actors in development projects of automotive manufacturers and present related work that investigates metrics to assess the quality of requirements. We propose a research agenda to analyze software requirements that are specified as unstructured text by using machine learning algorithms. Quality attributes based on experts' input and relevant literature help to create an assessment tool to analyze the perceived quality of requirements. With these results, we plan to develop quantitative measures that serve as input for a support vector machine. We aim to offer a tool that automatically predicts the quality of unstructured text in the communication between actors of a service system.

The next step of our research contains a comprehensive literature review to gain information about fundamental and current research referring to communication in service systems and quality measurement of requirements. We plan to finalize the interview questionnaire near-term and start interview sessions with relevant actors soon. Simultaneously we analyze the dataset and identify potential and relevant tasks for pre-processing the data.

Acknowledgments. The author would like to thank Léa Vernisse for her general support regarding the implementation of the assessment tool and her input for the research issues as well as Gerhard Satzger for his continuous support.

References

1. Maglio, P.P., Vargo, S.L., Caswell, N., Spohrer, J.: The service system is the basic abstraction of service science. Inf. Syst. E-Bus. Manage. **7**(4), 395–406 (2009)
2. Tien, J.M., Berg, D.: On services research and education. J. Syst. Sci. Syst. Eng. **15**(3), 257–283 (2006)
3. Lusch, R.F., Vargo, S.L.: The Service-Dominant Logic of Marketing: Dialog, Debate, and Directions. M.E. Sharpe, New York (2006)
4. Spohrer, J., Maglio, P.P., Bailey, J., Gruhl, D.: Steps toward a science of service systems. Comput. (Long. Beach. Calif.) **40**, 71–77 (2007)
5. Böhmann, T., Leimeister, J.M., Möslein, K., Leimeister, J.: Service systems engineering. Bus. Syst. Eng. **6**(2), 73–79 (2014)
6. Gaiardelli, P., Songini, L., Saccani, N.: The automotive industry: heading towards servitization in turbulent times. In: Servitization in Industry (2014). http://doi.org/10.1007/978-3-319-06935-7-4
7. Verstrepen, S., Deschoolmeester, D., Berg, R.J.: Servitization in the automotive sector: creating value and competitive advantage through service after sales. In: Mertins, K., Krause, O., Schallock, B. (eds.) Global Production Management. ITIFIP, vol. 24, pp. 538–545. Springer, Boston (1999). doi:10.1007/978-0-387-35569-6_66
8. Westphal, I., Nehls, J., Wiesner, S., Thoben, K.-D.: Steigerung der Attraktivität von Elektroautomobilen durch neue Produkt-Service-Kombinationen. Industrie Manage. **5**, 19–24 (2013)
9. Broy, M.: Automotive software and systems engineering. In: Proceedings of the Second ACM and IEEE International Conference on Formal Methods and Models for Co-design, pp. 143–149 (2005)
10. Grimm, K.: Software technology in an automotive company: major challenges. In: Proceedings of the 25th International Conference on Software Engineering, pp. 498–503. IEEE Computer Society (2003)
11. Kneuper, R., Sollmann, F.: Normen zum Qualitätsmanagement bei der Softwareentwicklung. Informatik Spektrum. **18**(6), 314–323 (1995)
12. Wilson, W.: Writing effective natural language requirements specifications. In: Software Technology Conference, pp. 1–14 (1997)
13. Alshazly, A., Elfatatry, A., Abougabal, M.: Detecting defects in software requirements specification. Alexandria Eng. J. **53**(3), 513–527 (2014)
14. Juristo, N., Moreno, A.M., Silva, A.: Is the European industry moving toward solving requirements engineering problems? IEEE Softw. **19**(6), 70–77 (2002)
15. Davis, A., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A.: Identifying and measuring quality in a software requirements specification. In: IEEE Software Metrics Symposium, pp. 141–152 (1993)

16. Lochmann, K., Fernández, D.M., Wagner, S.: A case study on specifying quality requirements using a quality model. In: APSEC, pp. 577–582 (2012)
17. Mund, J.: Quality Assessment of Requirement Specifications Using Metrics–A Research Proposal (2013)
18. Bellotti, M., Mariani, R.: How future automotive functional safety requirements will impact microprocessors design. Microelectron. Reliabil. **50**(9), 1320–1326 (2010)
19. Papadopoulos, Y., Parker, D., Grante, C.: Automating the failure modes and effects analysis of safety critical systems. In: Proceedings of the Eighth IEEE International Symposium on High Assurance Systems Engineering, pp. 310–311 (2004)
20. Weber, M., Weisbrod, J.: Requirements engineering in automotive development-experiences and challenges. In: Proceedings of the IEEE International Conference on Requirements Engineering, pp. 331–340 (2002)
21. Génova, G., Fuentes, J., Llorens, J., Hurtado, O., Moreno, V.: A framework to measure and improve the quality of textual requirements. Requir. Eng. **18**(1), 25–41 (2013)
22. Parra, E., Dimou, C., Llorens, J., Moreno, V., Fraga, A.: A methodology for the classification of quality of requirements using machine learning techniques. Inf. Softw. Technol. **67**, 180–195 (2015)
23. Femmer, H., Méndez Fernández, D., Wagner, S., Eder, S.: Rapid quality assurance with requirements smells. J. Syst. Softw. **123**, 190–213 (2015)
24. Brocke, V., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A.: Reconstructing the giant: on the importance of rigour in documenting the literature search process. In: Proceedings of the 17th European Conference on Information Systems, Verona, pp. 2206–2217 (2009)
25. Rowley, J., Slack, F.: Conducting a literature review. Manage. Res. News **27**(6), 31–39 (2004)
26. Ebert, C.: Systematisches Requirements Engineering: Anforderungen ermitteln, dokumentieren, analysieren und verwalten. dpunkt.verlag, Heidelberg (2014)
27. Rupp, C., Klaus, P.: Basiswissen Requirements Engineering: Aus-und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level. dpunkt.verlag, Heidelberg (2015)
28. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: writing a literature review. MIS Q. **26**(2), xiii–xxiii (2002)
29. IEEE Std. 830-1998: Recommended Practice for Software Requirements Specification. IEEE (1998)
30. Kaindl, H., Lutz, B., Tippold, P.: Methodik der Softwareentwicklung: Vorgehensmodell und State of the art der professionellen Praxis. Springer Fachmedien Wiesbaden GmbH, Wiesbaden (1998)
31. vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R., Cleven, A.: Standing on the shoulders of giants: challenges and recommendations of literature search in information systems research. Commun. Assoc. Inf. Syst. **37**(1), 205–224 (2015)
32. Cooper, H.: Organizing knowledge syntheses: a taxonomy of literature reviews. Knowl. Technol. Policy **1**, 104–126 (1988)
33. Walstrom, K.A., Hardgrave, B.C.: Forums for information systems scholars: III. Inf. Manage. **39**(2), 117–124 (2001)
34. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: CRISP-DM 1.0 step-by-step data mining guide (2000)
35. Kivinen, J., Warmuth, M.K., Auer, P.: The perceptron algorithm versus winnow: linear versus logarithmic mistake bounds when few input variables are relevant. Artif. Intell. **97**(1–2), 325–343 (1997)

36. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998). doi:10.1007/BFb0026683
37. Hsu, C., Chang, C., Lin, C.: A practical guide to support vector classification. Department of Computer Science and Information Engineering, National Taiwan University (2003)
38. Saunders, M., Lewis, P., Thornhill, A.: Research Methods for Business Students, 6th edn. Pearson Education, Harlow (2012)
39. King, N.: Using interviews in qualitative research. In: Cassell, C., Symon, G. (eds.) Essential Guide to Qualitative Methods in Organizational Research, pp. 11–22. Sage, London (2004)
40. Dröschel, W., Wiemers, M.: Das V-Modell 97: der Standard für die Entwicklung von IT-Systemen mit Anleitung für den Praxiseinsatz. Walter de Gruyter GmbH & Co. KG (1999)