2023 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC) | 979-8-3503-1517-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICE/ITMC58018.2023.10332295

Alignment of Quality Models for Assessing Software Requirements in Large-scale Projects: A Case from Space

Patrick Beyersdorffer*, Jürgen Münch* and Marco Kuhrmann*

*Reutlingen University | Herman Hollerith Center, Reutlingen, Germany,
Email: {name.surname}@reutlingen-university.de

***Abstract*—Large critical systems, such as those created in the space domain, are usually developed by a large number of organizations and, furthermore, they have to comply with standards. Yet, the different stakeholders often do not have a common understanding of the needed quality of requirements specifications. Achieving such a common understanding is a laborious process that is currently not sufficiently supported. Moreover, such a common understanding must be aligned with the standards. In this paper, we present an approach that can be used to align the different stakeholder perceptions regarding the quality of requirements specifications. Existing quality models for requirements specifications are analyzed for equivalences, and transferred into a common representation, the so-called Aligned Quality Map (AQM). Furthermore, a process is defined that supports the alignment of different stakeholder perspectives with regard to the quality of requirements specifications using AQM, which is validated in a case study in the context of European space projects. AQM has been created and populated with an initial set of quality models. It is designed in such way that it can be extended to include further quality models. The case study has shown that an alignment of different stakeholder perspectives and the quality model of the European Cooperation for Space Standardization using AQM is feasible. The approach allows for aligning different stakeholder perspectives for a common understanding of the quality of requirements specifications in the context of standards. Furthermore, AQM supports the assessment of requirements specifications.**

***Index Terms*—Quality attribute, quality model, quality assessment, software requirements specification, aerospace**

I. Introduction

The success of software and system development projects depends to a large extent on the quality of the software requirements specification (SRS) [1], [2]. A well-written SRS is a contract between customers and suppliers that documents the requirements to be met in a project for being considered successful. Hence, a high-quality SRS is key to ensure that the final product meets the needs of the end users [3]. "High quality" also means minimizing and avoiding misunderstandings and miscommunication, which helps implement a streamlined development process [2], [4]. Nevertheless, quite often SRS contain errors that can lead to delays, rework, and a product that does not meet the users' needs. The NaPiRE survey [5] named incomplete SRS and communication flaws as frequent problems significantly impacting a project's success. Especially during software development in large-scale projects (e.g., in the domains space or automotive), such problems have to be solved and the agreed quality goals need to be met by the suppliers [6]. To guarantee that the SRS serves as a reliable means and is not affected by potential problems, quality assurance methods need to be implemented. However, these methods must adhere to different quality expectations of involved stakeholders, and also have to comply with standards. For formulating quality expectations, the use of quality models is considered a promising approach [7].

Problem Statement and Objective: Quality models often remain abstract [8], allow for multiple interpretations [2], can be incomplete [9], and might disregard context-specific aspects [10]. To deal with variability in quality modeling, our goal is to extend the body of knowledge with an approach for aligning quality models. This alignment shall minimize misunderstandings about quality and improve the assessment of SRS, especially in large-scale projects. Consensus between specific quality expectations of stakeholders, standards, domain- and company-specific constraints shall be achievable.

Contribution: We propose the *Aligned Quality-Model Map* (AQM), which combines and extends related approaches. AQM includes 10 SRS quality models that represent different, yet aligned, perceptions about SRS quality. Furthermore, we developed a process for applying AQM during SRS quality assessment. The process was evaluated using a case in the context of European space projects.

Outline: The rest of the paper is structured as follows: Section II presents the background and related work. The research design, including the research questions, the data extraction, the building of the solution model, and the description of the case used for AQM's evaluation is described in Section III. In Section IV, we present our results, which we discuss in Section V. We conclude our paper in Section VI including a discussion of future work.

II. Background & Related Work

We present the background and related work with a focus on quality assessment based on quality models. We also describe how our work extends previous research.

A. Quality Assessment of Requirements Specifications

Assessing the quality of an SRS document solely through the calculation of its defect density is not an adequate method [11]. To rigorously evaluate the quality of SRS, quality assessment focuses on assessing specific quality attributes to gain a differentiated evaluation of the complex documents. Such quality attributes are defined as inherent properties of an individual requirement or of the whole SRS [12]. During the assessment, the fulfillment of acceptance criteria per quality attribute is determined. The assessment results can then be utilized in reviews with the customer or to demonstrate compliance with regulatory constraints.

There is a wide range of quality attributes applicable to SRS [13]. Several publications utilize the "Three C's" (Completeness, Consistency, and Correctness) or ambiguity as key quality attributes. The ISO/IEC/IEEE 29148 [12] standard also addresses these attributes in the generic definition of 14 attributes. Gregory [11] comments on these quality attributes highlighting the missing objectivity and discusses the possibilities of their measurement. Consensus regarding the interpretation of the quality attributes' meaning is considered crucial for defining related metrics and for understanding the measurement results. A survey by Atoum et al. [14] on SRS quality assurance techniques confirms this need while existing assessment frameworks lack uniformity.

The ISO/IEC 25010 [15] defines a generic quality modeling framework for software development. Quality attributes are arranged in a model using hierarchical decomposition, i.e., higher-level quality attributes are refined into lower-level attributes that allow for measurement. The ISO/IEC 25020 [16] defines *base metrics* for measuring these lower-level attributes and provides *derived metrics* that aggregate measurement data for the assessment of higher-level attributes. Yet, the lack of common decomposition techniques results in varying abstraction levels in different models [7]. Aggregation methods need to be defined to use low-level measurement data for assumptions about the overall quality. Due to missing precise definitions of the quality attributes and their model-dependent abstraction levels, aggregation is challenging [7]. Eventually, lacking precision and variability in quality modeling can lead to different and/or incompatible interpretations of quality.

B. Related Work

Saavedrea et al. [17], [18] conducted a literature study on quality attributes for SRS with primary studies until 2013. They analyzed the respective meaning of the attributes and studied how these attributes impact each other. However, the authors focused on attributes at high abstraction levels that are not immediately applicable for measurement. In addition, Montgomery et al. [13] provided an updated systematic mapping study on quality attributes of SRS until 2021, resulting in a comprehensive overview of the research field. In particular, 12 higher-level attributes were identified, as described in [17], [18]. Authors also consider a lower abstraction level with 111 extra attributes. They highlighted that consensus naming of this multitude of attributes was missing, resulting in different interpretations of their meaning. Inspired by the insights of [13], our work addresses the need for a more precise and shared understanding of individual quality attributes. As [13] suggest, instead of redefining quality attributes and models, we use existing work to create a common knowledge base as reliable input for the next research steps. Our work extends this knowledge base by additionally considering quality models from standards and industry guidelines.

The *Comprehensive Quality Model Landscape* (CQML) by Kläs et al. [19] provides a meta-modeling language for categorizing quality models. The approach aims to compare existing quality models from different viewpoints, which allows for orientation in the broad area of quality models, improves communication about quality attributes, and enables analysis of quality goals that can be achieved with these quality models or if gaps remain. By utilizing this approach, the elements of a quality model can be clearly defined. We extend the meta-model to compare the meaning of quality attributes from different models and align them at a semantic level. We applied the strategy by Frattini et al. [20] and created an ontology for a meaningful organization of quality attributes. They initiated a community effort to structure the existing body of knowledge on SRS quality in a *Requirements Quality Factor Ontology* (*reqfactoront*) with which they address the problem that quality attributes can have multiple representations, and that explicit definitions of the concepts and dependencies are currently missing. Our contribution to this effort is to extend their approach by integrating and aligning whole quality models of SRS, not only individual quality attributes.

We integrate and build upon the CQML and *reqfactoront* conceptualizations to align differently modeled SRS quality attributes. We thus fill a gap in research, namely the alignment of quality models to achieve semantic consistency for finally improving the reliability of SRS quality assessment.

III. Research Design

We present our research method, which implements a multi-method approach that consists of a literature study, the creation of a solution model, and the model's evaluation using a case study. The different steps are explained in subsequent sections after providing the overall objective and research questions.

A. Research Objective and Research Questions

The overall goal is to extend the body of knowledge with an approach for aligning quality attributes across quality models and to make this alignment applicable for quality assessments of software requirements specifications (SRS) in large-scale projects. For this, we pose the following research questions:

RQ1: *Which quality models for software requirements specifications exist?* Quality models describe individual aspects of quality and can further be used to assess or predict quality [7]. However, due to varying modeling approaches and stakeholder expectations, these quality models represent varying interpretations of quality [2]. The aim of this research question is to create an overview of existing quality models to study the various definitions for SRS quality.

RQ2: *How can varying quality models be aligned at a semantic level?* Varying and contradicting interpretations of SRS quality can negatively influence the reliability of quality assurance methods [7]. Therefore, we aim to develop a solution that aligns the quality models collected in answer to *RQ1*. The resulting solution shall provide a common understanding of quality expectations and shall further be extendable to support different quality assurance scenarios to deal with variability [9] as well as context-specific aspects [2] of the requirements engineering process.

RQ3: *Can semantic alignment of quality models support the quality assessment of SRS for large-scale projects?* Based on the outcomes of *RQ2*, we define a process for applying the alignment of quality models during quality assessments. A case study in the context of European space projects shall demonstrate the feasibility of our approach, as such projects are prime examples of large-scale system development involving a large number of stakeholders and standards. The results of this research question contribute to the need that the development of quality assessment techniques faces challenges due to the plurality of quality attributes and varying quality expectations from different stakeholders [14].

B. Data Collection and Extraction

To identify relevant quality models in academia and industry, in a first step, a snowballing-based literature study according to Wohlin [21] was conducted to retrieve widely used quality models. For a full overview of the field, we refer to the secondary study of [13]. A start set used for forward and backward snowballing was created using search queries for the *ACM Digital Library*, *ScienceDirect*, and *IEEE Xplore*, as these databases contain articles as well as standards related to the research questions. Combinations of the search terms “requirements specification”, “requirements engineering”, “quality”, “model” yielded 23 papers that were, based on the analysis of the abstracts, included for full-text review. We included papers that address generic or instantiated SRS quality models. If quality models were not the main contribution, as identified during a full-text review, the papers were excluded and not considered as input for the next snowballing iterations. We explicitly did not exclude standards or industry guidelines to retrieve a variety of quality models.

The data extraction was performed together with the development of the solution model (cf. next section). All available information about the quality attributes in the analyzed models as well as their dependencies were extracted from the papers and documented in the solution model.

C. Building the Solution Model

Formalization and comparison between quality models were already achieved by the CQML approach [19], as introduced in Section II-B. We applied the process from [20] to transfer the collected data to CQML. The *Web Ontology Language* (OWL) and *Protégé* [22] were used for model building and data management, which results in a formal, explicit and shared conceptualization of domain knowledge [23]. Due to the formal logic, the formalized knowledge is automatically processable. Furthermore, the explicitness reduces misunderstandings and provides a concise knowledge base for practitioners.

We aim to contribute to the community effort of Frattini et al. [20]. For this, we reuse characteristics from their ontology development project and enriched CQML with extra details that are relevant for comparing the meaning of quality attributes. However, we do not reuse the whole conceptualization for two reasons: first, our research focuses on the semantic alignment of quality models. In contrast, Frattini et al. built a knowledge base with a focus on quality measurement approaches and related datasets. Therefore, the conceptualization domains do not intersect in all areas and, therefore, we decided to exclude the measurement details at the current state of our work. Second, our work focuses on quality models rather than on quality attributes. Frattini et al. have only indirectly addressed the hierarchical composition of quality attributes in a model by assigning quality attributes to preselected (so-called) higher-level quality aspects. To allow for a more differentiated quality modeling, we use the CQML concepts for hierarchical modeling.

To conceptualize semantic equivalence between quality attributes from different quality models, the combined approaches need to be extended by a new relation, as described in Section IV.

D. Case Study

The feasibility of the solution model is demonstrated using a case study in the context of large-scale European space projects, which are regulated by the constraints of the *European Cooperation for Space Standardization* (ECSS). Due to the large customer-supplier chain, which is common in this domain, individual expectations about quality from involved companies need to be aligned and, furthermore, need to comply with the ECSS standards [6].

Case Description: The following case is used for the feasibility analysis: During a requirements review between the customer and supplier, the responsible requirements and quality engineers want to agree on quality attributes for implementing suitable quality assessment methods for the SRS. As both parties already use internal quality models, quality attributes with the same meaning must be identified to facilitate the alignment of the individual quality interpretations. Also, the different interpretations need to be aligned with the requirements of the ECSS standard’s quality model. Furthermore, the quality assessment definition shall be improved by an overview of the scope of the agreed-on quality attributes, which provides the quality engineers with a reference of quality attributes that are currently not considered in the assessment. An overview of measurement possibilities for the agreed-on quality attributes is needed to allow for the assessment.

Case Execution: To support requirements- and quality engineers accessing the solution model, we developed a web-based tool. The tool hides the complex interaction with the ontology utilizing SPARQL queries and provides single-click functionalities to retrieve the needed information. A researcher, who was not involved in the development of the solution model, used the tool to execute the case defined above. The goal of this case was to study whether the alignment of quality models with a particular focus on the ECSS standard can be achieved with the solution model by using the developed tool. This includes finding proper mappings of the different quality perceptions as well as identifying gaps, i.e., which parts of the ECSS quality model are not (yet) covered in other models and, therefore, require an extra agreement by the parties involved. Also, a successful mapping of the solution model contents with the ECSS standard allows for checking and confirming a joint interpretation of quality in space projects.

IV. Results

We present the results structured according to the research questions as described in Section III-A. The discussion of our findings can be found in Section V.

A. RQ1: SRS Quality Models

After three iterations, the snowballing procedure saturated and yielded only papers already included in our paper set[1] so that the termination condition was reached [21]. In total, forward and backward snowballing identified 10 quality models for SRS [12], [24]–[32]. Table I provides an overview of the selected quality models. The majority of the papers found was of scientific nature. Papers were not included if: (i) the scope was quality requirements for a product; (ii) the main contribution was formal languages for SRS to eliminate quality deficits caused by natural language; (iii) general quality measurement of SRS without a reference to a quality model, or to a model that was considered in our paper set. The industry perspective is represented by the ISO/IEC/IEEE 29148:2018 [12], as this standard provides a generic SRS quality model applicable for multiple domains. Also, the handbook *Education and Training for Certified Professionals for Requirements Engineering* [27] was identified.

In total, the 10 identified quality models define 135 quality attributes. Only four of the selected models provide means to explicitly model hierarchical dependencies between quality attributes to formalize a differentiated understanding of SRS quality. As noted in Section III-B, the data extraction using these 10 quality models, happened in the course of building the solution model, which is described in the following.

B. RQ2: Alignment of Quality Models

The extracted data from the literature study was used to instantiate the solution model[2]. To align the quality models at a semantic level, existing conceptualization approaches are combined and a new semantic equivalence relation was defined as described in subsequent sections.

[1] Details on the literature study can be found in the replication package: https://gitlab.reutlingen-university.de/anuki/AQM-Interaction

[2] Detailed descriptions of the conceptualization, the formalization process of the identified quality models, and the equivalence analysis are documented in the replication package.

TABLE I
Collected Quality Models for SRS (incl. number of quality attributes (QA), number of dependencies (Dep)

Author(s)	Type	Reference	#QA	#Dep
Berry et al.	Sci. Paper	[25]	7	3
Belfo	Sci. Paper	[24]	7	-
Davis et al.	Sci. Paper	[26]	23	-
Glinz et al.	Handbook	[27]	12	-
Halligan	Sci. Paper	[28]	10	-
Heck & Zaidman	Sci. Paper	[29]	10	7
ISO/IEC/IEEE	Standard	[12]	14	-
Medeiros et al.	Sci. Paper	[30]	13	-
Satio et al.	Sci. Paper	[31]	12	8
Takoshima & Aoyama	Sci. Paper	[32]	27	19
	Σ		*135*	*37*

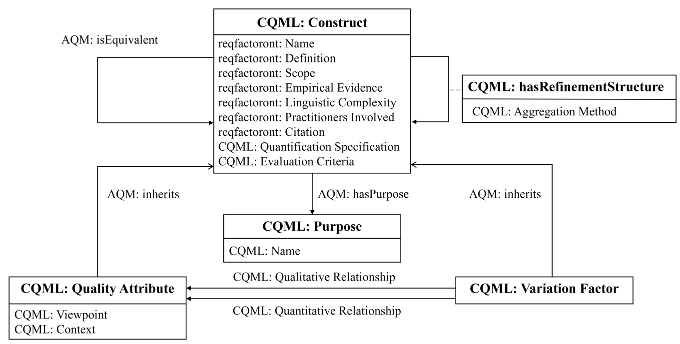

Fig. 1. Integration and extension of the conceptualizations into AQM

1) Conceptualization: The extracted data was formalized as shown in Fig. 1. According to Kläs et al. [19], a quality model consists of generic *Constructs*. A *Construct* is represented as OWL class with OWL annotations for the respective characteristics. A *Quality Attribute*, as one element of a quality model describing a specific quality aspect, inherits these characteristics. It can have a specific *Viewpoint* from which the quality aspect is described, e.g., to understand the perspective of a specific stakeholder. A *Domain* to which an attribute refers to is described, e.g., a specific process phase or application domain.

To measure a certain *Quality Attribute*, *Quantification Specification* methods can be added and, if possible, compared to *Evaluation Criteria*, such as thresholds or benchmarks. If a lower-level *Quality Attribute* contributes to a higher-level *Quality Attribute*, this hierarchical composition is conceptualized through the *Refinement Structure* property. Measurement data can be aggregated from lower to higher abstraction levels via *Aggregation Methods* to retrieve an overall metric from differentiated quality aspects. Finally, quality can be influenced by external factors, expressed through *Variation Factors* on *Quality Attributes*, that can be quantitatively or qualitatively formalized. Note, that these *Variation Factors* were also extracted from the identified quality models, but are not the main contribution of this paper.

The CQML approach was built based on the assumption that depending on which *Constructs* a quality model contains,

TABLE II

RESULTS OF THE EQUIVALENCE ANALYSIS FOR 10 QUALITY MODELS (INCL. THE SCOPES: DOCUMENT (D) OR REQUIREMENT (R))

Equivalent Quality Attributes	**Scope**	**Label from [17]**
Able to be validated [12]; Verifiability and Validity [24]	D	-
Achievable [26]; Achievable [32]; Feasible [12]; Feasibility [24]	D	Achievable
Annotated by Version [26], Change History [30]	D	Annotated by Version
Non-redundant [32]; Singular [12]; Singularity [28]	R	Atomic
Backward Traceability [32]; Traced [26]	D	Backward Traceability
Complete [26]; Complete [12]; Complete [32]; Complete [27]; Completeness [24]; Completeness [25]	D	Complete
Complete [12]; Complete [27]; Completeness [28]	R	Complete
Clarity [28]; Understandable [27]; Understandability [30]	R	Comprehensible
Completeness [29]; Completeness “agile” [30]; Descriptive Completeness [32]	R	-
Concise [26]; Simplicity [30]	D	Concise
Connectivity [28]; Preparation of glossary [31]	D	-
Confirm to Goal [31]; Correspondence to project goals [31]; Project objective [32]	D	-
Conformant [27]; Template [32]; Template usage [31]; Uniformity [29]	D	-
Consistency / Correctness [29]; Correct [26]; Correct [32]; Correctness [25]; Correctness [24]	D	Correct
Adequate [27]; Correct [12]	R	Correct
Cross-Referenced [26]; Cross-Referenced [32]	D	Cross-Referenced
Feasible [12]; Feasibility [28]	R	Feasible
Forward Traceable [29]; Forward Traceability [32]; Identifiability [31]; Traceable [26]; Unique Identifier [29]	D	Forward Traceability
Consistent [12]; Consistent [32]; Consistent [27]; Consistency [28]; Consistency [25]; Internally Consistent [26]	D	Internally Consistent
Label [32]; Presence of Identifier [31]	D	-
Modifiable [26]; Modifiable [32]; Modifiable [27]; Modifiability [28]	D	Modifiable
Necessary [27]; Necessary [12]	R	-
Non-functional Requirements [30]; Non-functional Requirements [29]	D	-
Not Redundant [26]; Non-Redundant [27]	D	Not Redundant
Organized [26]; Searchable [32]	D	Organized
Precise [26]; Quantitative [32]	D	Precise
Annotated by Relative Importance [26]; Priority [29]	D	Ranked by Importance
Traceable [32]; Traceable [27]; Traceability [30]	D	Traceable
Clearness/Unambiguously [24]; Freedom from ambiguity [32]; Uniguity [25]; Unambiguous [26]; Unambiguous [32]	D	Unambiguous
Clarity [30]; Non-Ambiguity [28]; Unambiguous [12]; Unambiguous [27]	R	Unambiguous
Comprehensible [12]; Readability [25]; Understandable [26]; Understandability [24]; Understandability [25]; Understandability [29]	D	Understandable
Testability [25]; Verifiable [26]; Verifiable [32]	D	Verifiable
Acceptance Criteria [29]; Acceptance Criteria [30]; Testability [28]; Verifiable [12]; Verifiable [27]	R	Verifiable

the Purpose of the respective model or model elements can be determined. For example, if the *Quality Attributes* have a *Quantification Specification* and related *Evaluation Criteria*, the attribute fulfills the purpose of assessing quality. If *Variation Factors* are quantifiable and relationships to *Quality Attributes* are modeled, quality can be predicted. If only *Refinement Structures* exist, then the model specifies the understanding of quality. In total, nine different *Purposes* are defined in CQML [19], which are present in our conceptualization as individual classes that allow for analyses based on the *Purposes* of interest. The conceptualization is further enriched with elements from the *reqfactoront* approach [20], like *Scope* (e.g., word, sentence or the full SRS document) and the *Empirical Evidence* on which a formalized *Construct* relies.

2) Semantic Alignment: To formalize semantic equivalence between quality attributes from different quality models, we extended CQML [19] and *reqfactoront* [20] conceptualizations with the property *isEquivalent* as shown in Fig. 1. This symmetric and transitive relation shall allow users or the reasoning engine of the ontology to find *Quality Attributes* or *Variation Factors* that share the same meaning, but are differently labeled or hierarchically decomposed due to the respective quality model's content and structure. The analysis of the meaning per extracted *Quality Attribute* and *Variation Factor* from the different quality models was performed based on the extracted information from the selected publications. If an equivalent meaning was found, the *isEquivalent* property was added. Semantic equivalence is defined as two *Constructs* sharing a *Definition* with the same meaning, if they define a common assessment *Scope* and *Context*, and if they are modeled from the same *Viewpoint*. If a quality model does not provide all this information, the missing characteristics were excluded. Note, that the extracted characteristics are also the main elements of the goal template of the Goal-Question-Metric (GQM) schema, which is formalized in CQML [19]. Thus, quality goals are directly aligned if they refer to equivalent *Quality Attributes*.

The result of this equivalence analysis of the 10 collected quality models is documented in Table II. We used the study by Saavedrea et al. [17] on the meaning of high-level quality attributes as a baseline and mapped equivalent attributes to these common labels. Equivalent quality attributes with no corresponding label in [17] are also listed. In total 108 out of 135 quality attributes from different models could be semantically aligned, resulting in 34 mappings on a common meaning (Table II). For 28 out of 135 attributes no equivalent attribute in the landscape was found. The large number of 108 quality attributes with at least one corresponding attribute

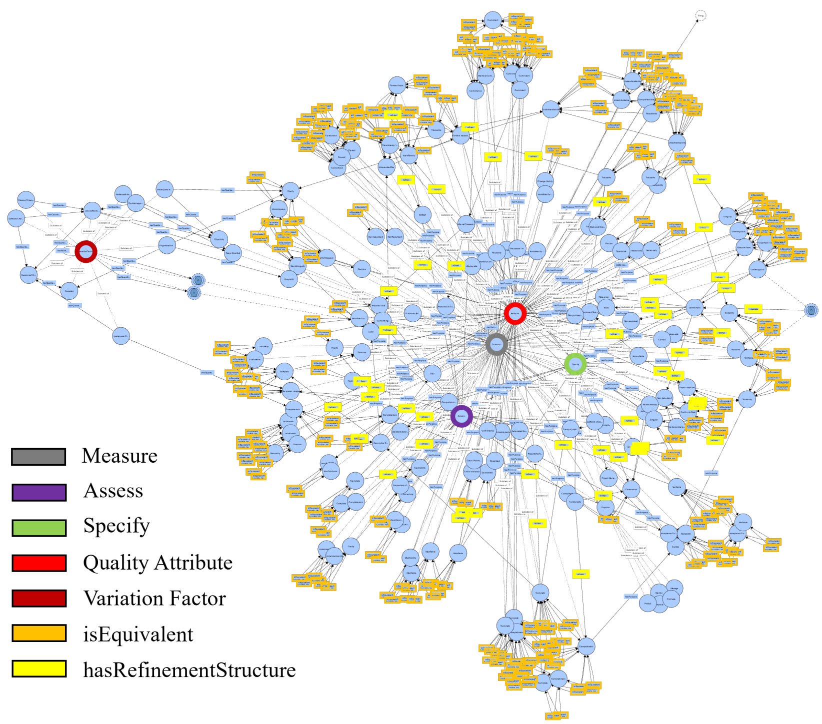

Fig. 2. Overview of the AQM as WebVOWL graph [33]. Colored circles indicate classes to which the surrounding instances are associated.

with the same meaning from another quality model allows for aligning these models at a semantic level, whereas mainly attributes at higher abstraction levels share their meaning.

As shown in Table II, the alignment is not directly achievable by simply comparing the quality attribute labels. It is necessary to consider the individual definitions of the quality attributes. For example, [29], [32] address the descriptive completeness of an SRS, i.e., that all textual and structural elements are present. However, they do not refer to the quality attribute of full completeness, meaning that also all requirements were elicited to fully meet the stakeholders' needs. However, Heck and Zaidman [29] labeled this quality attribute as "Completeness". That is, there is a risk of misinterpretation when stating that an SRS was assessed to meet the quality attribute of completeness, but the assessment only evaluated descriptive completeness, referring to another quality model. Furthermore, quality attributes with the same label can define and assess quality with a different scope, e.g., considering a single requirement or the SRS document as a whole as further detailed in Table II.

3) Aligned Quality-Model Map: To reduce misunderstandings, we propose the *Aligned Quality-Model Map* (AQM[3]) for aligning the quality attributes. Fig. 2 provides an overview of AQM containing the 10 quality models, using the previously described equivalency relations. High-level quality attributes that share the same meaning through multiple quality models are grouped at the corner of the map and are further refined by lower-level quality attributes in the middle. These lower-level attributes share to some degree equivalence relationships or are independent of each other. Through the semantic alignment in common high-level attributes, quality models can be directly extended with these lower-level attributes by following the equivalence relations. The variation factors located on the left side can influence quality attributes from multiple quality models.

The latest version of AQM consists of 62 unique quality attributes. Thereby, unique means that equivalent attributes are semantically aligned and now represent one common attribute, as examined in the previous subsection. Due to the openness of ontologies, AQM is not closed and can be enriched with additional concepts, resulting in an iterative and incremental formalization of the domain of interest as applied by Frattini et al. [20].

C. RQ3: Quality Assessment of SRS in large-scale Projects

AQM shall improve the quality assessment of SRS in large-scale projects. Therefore, as a first step, we define a process for using AQM during the quality assessment. The feasibility of the process was studied in a case as described afterwards.

[3]The ontology file can be accessed via the replication package.

1) Quality Assessment Process: We present a process for using AQM during the quality assessment of SRS. The specific technique as well as the achievable degree of automation depend on the domain, the individual quality attributes to be measured, and the existing metrics [14]. The following process is defined for applying AQM independently from these factors:

I. Instantiation phase

a. Check, if all quality models, relevant for a domain or project are already formalized.
b. Add missing quality models to AQM.

II. Execution phase

a. Select quality attributes that are relevant for contractual agreements, regulatory approval, or the definition of quality goals.
b. Develop an understanding of the assessment scope of the selected quality attributes and, if necessary, extend the scope with further dependent attributes.
c. Use dependent quality attributes with measurement purposes from equivalent quality models for quality assessment.

The assessment finally provides a consensus status report for quality assurance, which is based on aligned quality attributes in case multiple quality models are used. This status report includes an overview of the scope of the considered quality attributes in comparison with the other attributes in the whole AQM. This allows for discussion if additional quality attributes need to be considered or if the scope is sufficient to meet project-specific as well as regulatory constraints.

The AQM process was designed for iterative, incremental application during the requirements engineering process. Technically, the search for quality attributes and their dependencies in AQM can be carried out visually, e.g., using the *Visual Notation for OWL Ontologies* (VOWL) or query-based using SPARQL commands. These techniques can further be integrated into specialized quality assessment tools.

2) Case Study Results: The case study follows the procedure described in Section III.D and applies the quality assessment process defined in the previous subsection. During the instantiation phase, AQM is extended with the ECSS quality model as defined in the ECSS-Q-HB-80-04C handbook [34][4]. The 10 quality models, which are already available in AQM, are used for simulating the varying interpretations of quality of different stakeholders in a large-scale space project. Each of these quality models now represents an interpretation of quality in the multi-level customer-supplier chain that needs to be aligned with the other interpretations and with the regulatory constraints of the ECSS.

The executed AQM process resulted in an alignment with a subset of the ECSS quality model attributes that address the desired quality of an SRS. Three of the ECSS quality attributes could be matched with other attributes of the AQM, namely: "Completeness", "Testability" can be aligned with "Verifiable" [17], and "Correctness" can be aligned with Halligan's [28] interpretation, which means that an SRS is errorfree. These equivalent attributes from other quality models are directly retrievable using our tool[5] with supportive references to the respective quality models in a visual representation based on VOWL as shown in Fig. 3. The nine ECSS quality attributes framing the SRS quality of European space projects cover only 13% of the current version of AQM. This scope analysis from the regulatory viewpoint in comparison with the full AQM is also visualized by the tool. Additionally, AQM offers the possibility to select quality attributes with the purpose of quality measuring as formalized by Kläs et al. [19]. That is, a subgraph can be generated that provides the user with quantification and evaluation criteria. Overall, the defined quality assessment process, based on the AQM, and in combination with the capabilities of the developed tool results in the complete execution of the defined case study.

V. Discussion

The contributions of the paper at hand, the limitations, and the threats to validity are discussed in this section. We start with answering the research questions.

A. Answering the Research Questions

Our work is grounded in the identified quality models for SRS. To answer *RQ1*, we provide an overview of 10 quality models extracted from a snowballing-based literature study. Different viewpoints on quality and different formalization approaches have been identified. For analyzing equivalences between the selected quality models, multiple aspects must be considered for comparing the meaning of the respective quality attributes. These aspects relevant for equivalence analysis were extracted from two existing quality model conceptualizations [19], [20]. For modeling equivalence relationships, these conceptualizations have been extended. To *RQ2*, the AQM model was proposed to deal with varying quality models and to enable quality model alignment at a semantic level. The developed conceptualization of equivalences between quality models can support quality assessment of SRS as proven in a case study for answering *RQ3*. Different labels of quality attributes do not cause misinterpretations anymore and stakeholder-specific quality expectations can directly be compared with regulations from standards. The opportunities for improving the SRS quality assessment was demonstrated by applying AQM supported by a tool in the context of large-scale European space projects.

B. Limitations

The formalization of measurement approaches and data sets by Frattini et al. [20] could be added to AQM to directly allow for quality measurement. For this, additional work is needed for aggregating low-level measurement data into assumptions about the related higher-level quality attributes. Furthermore, a systematic search for variation factors is required to allow for

[4]Note that the related standards [35], [36] provide further recommandations for improving the quality of SRS for space projects. Transferring these recommandations to the quality model is outstanding and could be beneficial for the domain. However, this transfer is not part of the present work.

[5]The AQM applicator tool was added to the replication package.

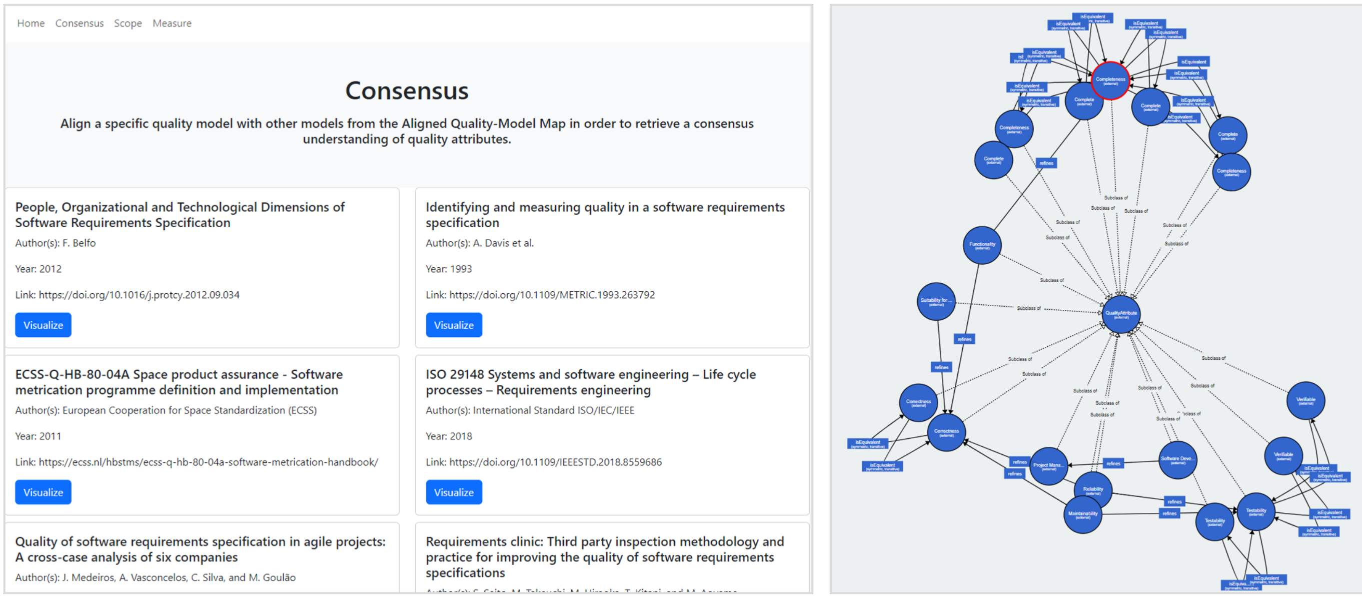

Fig. 3. Snapshot of the developed tool to select a quality model for alignment with other models (left). The example shows the ECSS quality model with equivalent quality attributes from other quality models (right).

indirectly measuring SRS quality. FernÃ¡ndez et al. [5] collected various factors for decreased SRS quality. Atoum et al. [14] highlight that besides internal quality attributes, external factors like management errors needs to be considered. The latest version of AQM can be extended with these variation factors as well as their impact on quality attributes.

AQM can support the automatic quality assessment. However, the majority of quality attributes are defined at the linguistic complexity level of pragmatism, meaning that contextual knowledge is required to assess the SRS according to the quality attributes. That is, an automatic measurement is currently applicable to "less complex" quality attributes at lower abstraction levels.

C. Threats to Validity

We discuss the threats to validity according to Wohlin et al. [37]. First, the meaning of a quality attribute could be misinterpreted and therefore incorrectly formalized. To reduce the impact on the reliability of AQM and to increase the *internal validity*, one researcher compared the meaning of the quality attributes twice with one week temporal distance. During the case study, another researcher, who was not involved in the development of AQM evaluated the resulting equivalency dependencies again. Furthermore, we compared and mapped the collected quality attributes to the former study by Saavedra et al. [17] to reduce ambiguity in labels and definitions. With this step, the *construct validity* is increased. Also, the reuse of already approved conceptualization approaches strengthens the clearness of the concepts. Considering the *external validity*, we do not claim completeness of the collected quality attributes. Even though AQM was explicitly designed to support varying quality models, and even though the underlying CQML [19] was applied for various quality models, we still cannot claim generalizability for AQM, since a practical evaluation has yet to come as part of future work. This future work should also include industry-hosted case studies as the currently provided proof of concept limits the *conclusion validity*. Nonetheless, with AQM, we laid the theoretical foundation and provided a proof of concept to be used and refined in future work.

D. Match & Contribution

The results of this work contribute to the *Management Science and Practices* field of interest by defining and implementing an *Engineering Management* approach with a specific focus on the complex task of reliable requirements engineering in large-scale projects.

VI. Conclusion

In order to use the software requirements specifications (SRS) as a reliable means in large-scale projects, it is crucial to align different interpretations of quality among stakeholders and, at the same time, including quality-related regulations defined by standards of a domain. We propose the *Aligned Quality-Model Map* (AQM) for aligning individual quality models to help achieve consensus about quality attributes during quality assessment of SRS. The application of AQM was demonstrated for SRS quality assessment in the domain of European space projects. Individual quality models were aligned with the regulations of the ECSS-integrated quality model. Due to the generic formalization, AQM can be extended and adopted, as defined by the AQM quality assessment process.

Future research based on this work will address application of quality metrics in large-scale projects. Therefore, the AQM can be used to select appropriate measurement options and collect measurement data depending on the viewpoint and on the context within a large-scale project. Based on the formalized dependencies between quality attributes, measurement results should automatically be aggregated to provide insights about the overall SRS quality. The paper at hand lays the foundation for a sophisticated toolchain that helps assess and predict the success of large-scale European space projects by automatically analyzing measurement data.

ACKNOWLEDGMENT

This work is partly funded by the German Aerospace Center (Project: ANuKI, funding number: 50RM2100A).

REFERENCES

[1] E. Knauss, C. El Boustani, and T. Flohr, "Investigating the impact of software requirements specification quality on project success," in *Lecture Notes in Business Information Processing*, ser. Lecture notes in business information processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 28–42.

[2] H. Femmer and A. Vogelsang, "Requirements quality is quality in use," *IEEE Softw.*, vol. 36, no. 3, pp. 83–91, May 2019.

[3] P. Thitisathienkul and N. Prompoon, "Quality assessment method for software requirements specifications based on document characteristics and its structure," in *2015 Second International Conference on Trustworthy Systems and Their Applications*. IEEE, Jul. 2015.

[4] W. Hu, J. C. Carver, V. K. Anu, G. S. Walia, and G. Bradshaw, "Detection of requirement errors and faults via a human error taxonomy," in *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. New York, NY, USA: ACM, Sep. 2016.

[5] D. M. Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M.-T. Christiansson, D. Greer, C. Lassenius, T. Männistö, M. Nayabi, M. Oivo, B. Penzenstadler, D. Pfahl, R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. Spinola, A. Tuzcu, J. L. de la Vara, and R. Wieringa, "Naming the pain in requirements engineering," *Empir. Softw. Eng.*, vol. 22, no. 5, pp. 2298–2338, Oct. 2017.

[6] C. R. Prause, M. Bibus, C. Dietrich, and W. Jobi, "Managing software process evolution for spacecraft from a customer's perspective," in *Managing Software Process Evolution*. Cham: Springer International Publishing, 2016, pp. 137–163.

[7] S. Wagner, *Software Product Quality Control*, 2013th ed. Berlin, Germany: Springer, Jul. 2013.

[8] M. Klas, C. Lampasona, and J. Munch, "Adapting software quality models: Practical challenges, approach, and first empirical results," in *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE, Aug. 2011.

[9] H. Al-Kilidar, K. Cox, and B. Kitchenham, "The use and usefulness of the ISO/IEC 9126 quality standard," in *2005 International Symposium on Empirical Software Engineering, 2005*. IEEE, 2005.

[10] H. Femmer, J. Mund, and D. M. Fernandez, "It's the activities, stupid! a new perspective on re quality," in *2015 IEEE/ACM 2nd International Workshop on Requirements Engineering and Testing*. IEEE, May 2015.

[11] S. Gregory, "Requirements engineering: The quest for meaningful metrics: Time for a change?" *IEEE Softw.*, vol. 36, no. 6, pp. 7–11, Nov. 2019.

[12] "Iso/iec/ieee international standard - systems and software engineering – life cycle processes – requirements engineering," *ISO/IEC/IEEE 29148:2018(E)*, pp. 1–104, 2018.

[13] L. Montgomery, D. Fucci, A. Bouraffa, L. Scholz, and W. Maalej, "Empirical research on requirements quality: a systematic mapping study," *Requir. Eng.*, vol. 27, no. 2, pp. 183–209, Feb. 2022.

[14] I. Atoum, M. K. Baklizi, I. Alsmadi, A. A. Otoom, T. Alhersh, J. Ababneh, J. Almalki, and S. M. Alshahrani, "Challenges of software requirements quality assurance and validation: A systematic literature review," *IEEE Access*, vol. 9, pp. 137 613–137 634, 2021.

[15] "Iso/iec international standard - systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models," *ISO/IEC 25010:2011*, 2011.

[16] "Iso/iec international standard - systems and software engineering – systems and software quality requirements and evaluation (square) – quality measurement framework," *ISO/IEC 25020:2019*, 2019.

[17] R. Saavedra, L. Ballejos, and M. Ale, "Quality properties evaluation for software requirements specifications: An exploratory analysis," *Anais do WER 2013 - Workshop em Engenharia de Requisitos*, pp. 6–19, Jan. 2013.

[18] R. Saavedra, L. C. Ballejos, and M. A. Ale, "Software requirements quality evaluation: State of the art and research challenges," 2013.

[19] M. Kläs, J. Heidrich, J. Münch, and A. Trendowicz, "Cqml scheme: A classification scheme for comprehensive quality model landscapes," in *2009 35th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, Aug. 2009.

[20] J. Frattini, L. Montgomery, J. Fischbach, M. Unterkalmsteiner, D. Mendez, and D. Fucci, "A live extensible ontology of quality factors for textual requirements," in *2022 IEEE 30th International Requirements Engineering Conference (RE)*. IEEE, Aug. 2022.

[21] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, USA: ACM, May 2014.

[22] M. A. Musen and Protégé Team, "The protégé project," *AI Matters*, vol. 1, no. 4, pp. 4–12, Jun. 2015.

[23] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data Knowl. Eng.*, vol. 25, no. 1-2, pp. 161–197, Mar. 1998.

[24] F. Belfo, "People, organizational and technological dimensions of software requirements specification," *Procedia Technol.*, vol. 5, pp. 310–318, 2012.

[25] D. M. Berry, A. Bucchiarone, S. Gnesi, G. Lami, and G. Trentanni, "A new quality model for natural language requirements specifications," in *Proceedings of the international workshop on requirements engineering: foundation of software quality (REFSQ)*, 2006.

[26] A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, G. Kincaid, G. Ledeboer, P. Reynolds, P. Sitaram, A. Ta, and M. Theofanos, "Identifying and measuring quality in a software requirements specification," in *[1993] Proceedings First International Software Metrics Symposium*. IEEE Comput. Soc. Press, 2002.

[27] M. Glinz, H. van Loenhoud, S. Staal, and S. Bühne, "Handbook for the CPRE Foundation Level according to the IREB Standard," 2020.

[28] R. J. Halligan, "Requirements metrics: the basis of informed requirements engineering management," in *Proc. Complex Systems Engineering Synthesis and Assessment Technology Workshop*, 1993.

[29] P. Heck and A. Zaidman, "A systematic literature review on quality criteria for agile requirements specifications," *Softw. Qual. J.*, vol. 26, no. 1, pp. 127–160, Mar. 2018.

[30] J. Medeiros, A. Vasconcelos, C. Silva, and M. Goulão, "Quality of software requirements specification in agile projects: A cross-case analysis of six companies," *J. Syst. Softw.*, vol. 142, pp. 171–194, Aug. 2018.

[31] S. Saito, M. Takeuchi, M. Hiraoka, T. Kitani, and M. Aoyama, "Requirements clinic: Third party inspection methodology and practice for improving the quality of software requirements specifications," in *2013 21st IEEE International Requirements Engineering Conference (RE)*. IEEE, Jul. 2013.

[32] A. Takoshima and M. Aoyama, "Assessing the quality of software requirements specifications for automotive software systems," in *2015 Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, Dec. 2015.

[33] S. Lohmann, S. Negru, F. Haag, and T. Ertl, "Visualizing ontologies with VOWL," *Semant. Web*, vol. 7, no. 4, pp. 399–419, May 2016.

[34] E. C. for Space Standardization (ECSS), "Space product assurance – software metrication programme definition and implementation," ESA Requirements and Standards Division, Noordwijk, Netherlands, ECSS-Q-HB 80-04A, 2011.

[35] ——, "Space engineering – technical requirements specification," ESA Requirements and Standards Division, Noordwijk, Netherlands, ECSS-E-ST 10-06C, 2009.

[36] ——, "Space product assurance – software product assurance," ESA Requirements and Standards Division, Noordwijk, Netherlands, ECSS-Q-ST 80C, 2017.

[37] C. Wohlin, M. Höst, and K. Henningsson, "Empirical research methods in software engineering," in *Empirical Methods and Studies in Software Engineering*, ser. Lecture notes in computer science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 7–23.