INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.257

Tailored metrics for assessing the quality of MBSE models

Iris Graessler [1], Dominik Wiechel [1,✉], Deniz Oezcan [1] and Patrick Taplick [2]

[1] Heinz Nixdorf Institute, Paderborn University, Germany, [2] Behr-Hella Thermocontrol GmbH, Germany

✉ dominik.wiechel@hni.upb.de

Abstract

System models are used to merge relevant engineering artefacts and relationships. Therefore, a high model quality must be ensured. Currently, there is no method for defining company-specific metrics to assess system model quality. In a six-step research approach, a method is defined: (1) literature analysis on quality assessment approaches, (2) derivation of success factors, (3) evaluation of approaches, (4) development of a method, (5) application in automotive industry, and (6) evaluation. The method supports system engineers to derive tailored metrics to objectively assess the model quality.

Keywords: model-based systems engineering (MBSE), system models, model quality, design methods

1. Introduction

Engineering of complex technical systems faces the challenge of efficiently orchestrating different development activities in interdisciplinary development teams, aiming to meet customer and market requirements (David D. Walden et al., 2015). In systems engineering, models are used to merge engineering artefacts and to consider relationships between them (Friedenthal et al., 2015). The purpose of these models is to meet the specific needs of the engineers or to answer their upcoming questions (Hause, 2011). Examples are engineering models of an electrical vehicle series, which are relevant for participating engineers and simultaneously for compliance with automotive regulations and standards such as the UN ECE regulations (United Nations, 2021; Graessler et al., 2023) or A-SPICE conformity (VDA QMC Working Group 13 / Automotive SIG, 2017). Due to the increasing importance of models, model quality is a key factor for the successful completion of engineering projects (Lange and Chaudron, 2005). In Model-Based Systems Engineering (MBSE), a central system model is created that represents and links interdisciplinary engineering artefacts at different levels such as requirements, functions, logical elements, and physical elements, verification and validation (RFLPV²) (Graessler et al., 2022; Gräßler and Oleff, 2022) The system model is the authoritative source of information. Thus, ensuring the quality is a fundamental part of maximising the benefits of the system model (Friedenthal and Burkhart, 2015). The data and information collected needs to be highly accurate; otherwise, data analytics, applications, or business processes is unreliable (Kilkenny and Robinson, 2018). Metrics to assess the quality are used to support effective decision making (Stvilia et al., 2007). A relevant aspect in assessing the model quality is that the quality of system models depends on the object of observation (Giraldo et al., 2018). Models are at different stages of maturity depending on the underlying model approach and the duration of the development project, which means that the quality of the objective of observation varies (Bansiya and Davis, 2002). The model quality of the system model must therefore be ensured in the different phases of engineering, starting with the early phases of the engineering project (Debbabi et al., 2010). Determining the information quality of engineering artefacts is a demanding and labour-intensive activity (Parra et al. 2019). The improvement and assurance of model quality is perceived as an independent management task due to the time-consuming resources involved (Rohweder et al., 2021). Quality assurance meetings can be conducted, to assess the model quality (Sargent, 2013). As a varying group of experts need to be involved, the quality of the system model needs to be ensured from different points of view. For evaluation, the experts cannot use existing standards and norms, because these are too general to determine the quality of specific artefacts (Parra et al., 2019). The currently occurring aspects are illustrated in Figure 1.

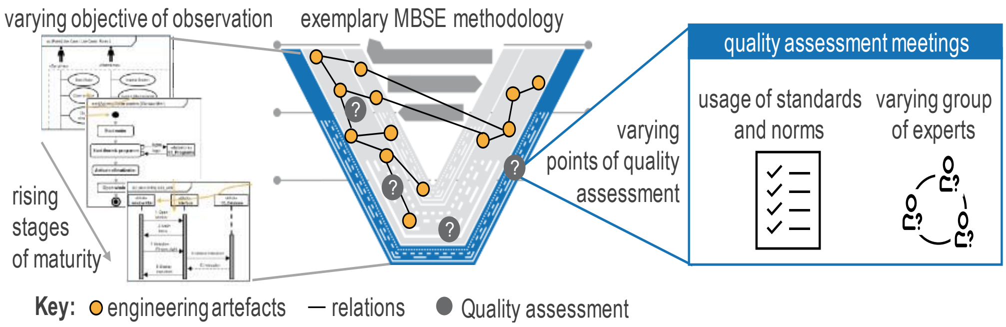

Figure 1. Quality assurance of system models in the V-Model (VDI/VDE, 2021)

2. Problems and challenges

In software engineering, quality assurance of software artefacts has been established for many years (Doan and Gogolla, 2019). Ensuring the model quality of interdisciplinary artefacts in system models requires new assessment approaches (Lehner et al., 2022). Approaches for assessing the model quality of system models exist, for example with reference to the Unified Modelling Language (UML) (Debbabi et al., 2010). Still, there are three main shortcomings of current approaches:

1. The approaches focus on standardised metrics without considering company-specific and role-specific needs.
2. The model quality metrics are not tailored to the different phases of the engineering project, as the information needed to assess quality may vary by the maturity of the system model.
3. Most approaches focus on metrics for the assessment of information quality, not for the assessment of the quality of an interdisciplinary system model.

Currently, there is no approach for defining company-specific metrics for assessing the model quality of interdisciplinary system models. As a result, there is a need for methods, models and tools, which help engineers to define company-specific metrics from the perspective of all participating stakeholders and integrate them in the engineering processes. In this paper, a method is adapted for defining company-specific metrics for assessing the model quality of system models. This objective is addressed by two research questions (RQ):

- RQ1: Which methodical steps are necessary to define company- and role-specific metrics and integrate them in the model processes of a company?
- RQ2: How can the defined metrics be mapped to different phases and maturity levels of a system model?

3. State of the art

The state of research is divided into three essential sections: Model-Based Systems Engineering is analysed, as well as model quality and metrics for evaluating the model quality.

Model-Based Systems Engineering

Model-Based Systems Engineering (MBSE) is defined as the application of formal modelling to support system requirements engineering, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases (International Council on Systems Engineering, 2007). Key activities in system modelling are model creation, model visualisation, model analysis and model management (Friedenthal and Burkhart, 2015). In MBSE, the system model is the central engineering artefact of the development process. A system model is created and maintained using three main components: a modelling language, a modelling tool and a modelling method (Delligatti, 2014). Different modelling languages exist, which support modelling engineers to create system models. Existing modelling languages like UML (Object Management Group, 2004) and SysML (Object Management Group, 2019) are used to create a system model in a formalised way. In addition, modelling tools can be used to create a system model, for example Cameo Systems Modeler from D'assault Systèmes, IBM Rhapsody or Enterprise Architect from Sparx System. Modelling methods support engineers in creating a system model. Exemplary modelling methods using SysML are OOSEM (Friedenthal et al., 2015) and SysMOD (Weilkiens, 2014). All main components have an impact on the model quality (Mohagheghi and Aagedal, 2007).

Model quality

Before defining an approach to assess the model quality, it is important to define the term "quality" consistently within an organisation (Ernadote, 2018). Quality is defined as the degree to which a set of inherent characteristics of an object fulfils requirements (DIN, 2015). Information quality (IQ) is defined as the degree of usability of information for the respective purpose (Krcmar, 2015). Based on existing studies (Wang and Strong, 1996) the German Society for Data and Information Quality has defined fifteen quality dimensions (ISO/IEC), including the categories maintainability, efficiency, usability, reliability, functionality and portability. In engineering, the IQ of information artefacts is relevant for the project success. Underlying artefacts have an inherent uncertainty that affects the quality of the information, for example an unclear and incomplete requirements specification (Pottebaum and Gräßler, 2020). A system model is the central information artefact in interdisciplinary engineering projects (Friedenthal et al., 2015). Therefore, the quality of system models and IQ have intersections. The necessary model quality also depends on the purpose, for example to strengthen the communication, to serve as documentation or to support decision-making (Mohagheghi and Aagedal, 2007). The quality and validity of a model is determined with respect to that purpose and can be done in model accreditation based on specific defined criteria (Sargent, 2013).

Metrics for evaluation of model quality

A metric is defined as a quantitative measure of the degree to which a system, component, or process possesses a given attribute (*610.12-1990 IEEE Standard Glossary of Software Engineering Terminology*). The use of metrics to assess the model quality is an important part of the development process (Henderson et al., 2023). By using metrics, engineers from different disciplines can ensure that the models they create and use are accurate, resilient and usable. The use of metrics to assess the model quality is an important part of the MBSE processes (Henderson et al., 2023). An exemplary framework is provided by Friedenthal and Burkhart (2015), who define key effectiveness measures for systems modelling, including expressiveness, preciseness, interoperability, manageability, usability, and customisability (Friedenthal and Burkhart, 2015). When deriving metrics, different requirements have to be fulfilled Kaiser et al. derive general requirements for the definition of quality metrics, which include interpretability, feasibility, adaptivity, normalisation and an interval scale (Kaiser et al., 2007).

4. Scientific approach

In order to answer the research questions, a six-step scientific approach is defined (based on (Ulrich, 1982)). In step 1, a systematic literature research is conducted to identify practical relevant approaches to measure the quality of system models. In step 2, success factors are derived to compare and evaluate the identified approaches. At this point, the literature-based success factors are completed from a practical perspective by expert interviews with four modelling experts from the automotive industry. In step 3, the approaches are categorised into three categories: frameworks, methods and application examples. The categorized approaches are compared and evaluated based on the success factors with a three-level scheme (1 fulfilled, 0.5 partially fulfilled, 0 not fulfilled). In step 4, based on the most suitable approach, a method is adapted and a model is developed to define company- and role-specific metrics to assess the model quality of system models and to integrate these into the company's processes. In step 5, the resulting method is applied for a case example from the automotive industry. In the last step, an application evaluation and a success evaluation are conducted. For the application evaluation, the results of step 5 are reviewed and discussed. For the success evaluation, the derived success factors are evaluated by four model experts from the automotive industry. The scientific approach is illustrated in Figure 2.

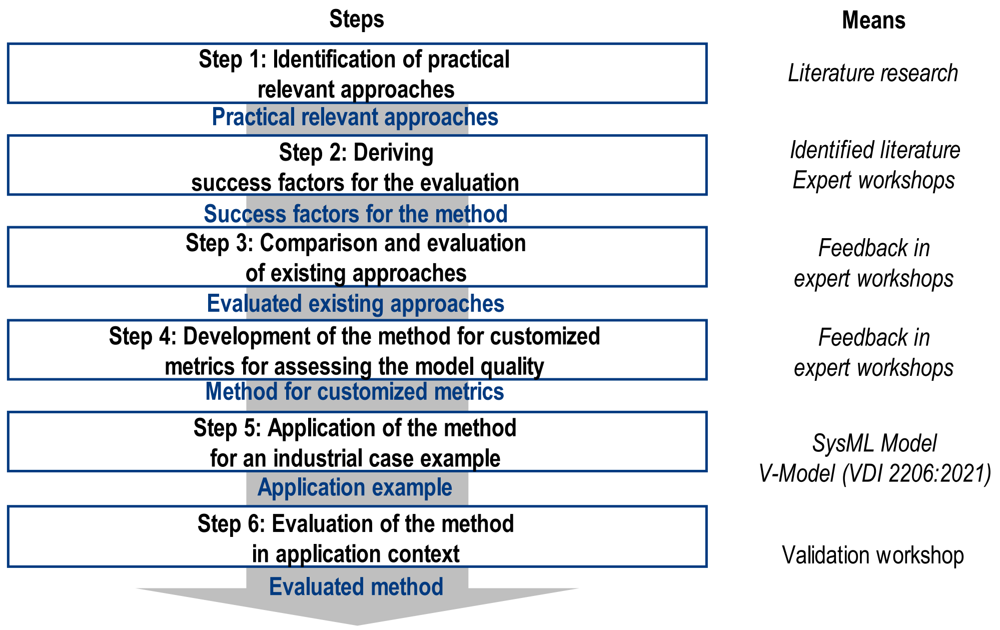

Figure 2. Scientific approach to derive the method for the assessment of system model quality

5. Method and Model for assessing the quality of system models

With reference to the identified research questions (section 2), search strings are defined for the systematic literature research. The strings are based on identified keywords, appropriate synonyms, and Boolean operators. They are primarily used to search the scientific databases of Scopus and Web of Science. In order to ensure completeness of the search, Google Scholar is considered subsequently. The identified elements are reduced by using selection criteria such as the applicability for interdisciplinary models and for other case examples. The number of identified documents and selected documents are shown in Table 1. Afterwards, the relevant approaches are described briefly and success factors are derived.

Table 1. Literature research

	IEEE Explore		Web of Knowledge		Google Scholar	
ID = identified documents, SD = selected documents	ID	SD	ID	SD	ID	SD
"Quality Metrics" AND ("MBSE" OR "Model-Based Systems Engineering")	184	7	20	3	171	5
"Model Quality" AND ("MBSE" OR "Model-Based Systems Engineering")	55	2	8	0	267	8
"Quality Assurance" AND ("MBSE" OR "Model-Based Systems Engineering")	140	2	2	1	1100	5
"SysML" AND "Model Quality"	0	0	0	0	355	11
"Quality Metrics" AND "System Model" AND "SysML"	0	0	0	0	92	2
Selected documents	11		4		31	

Lange and Chaudron (2005) describe a four-level approach defining a quality model for UML by mapping characteristics to the purpose and primary use of the artefact. They also describe a tool called QualityView to visualise the metrics. Based on this, Lange et al. (2007) describe the tool SDMetrics, which calculates metrics from UML models and visualises them in graphs or tables. Dromey (1995) proposes a model for assessing software product quality by developing quality-carrying properties to link software product characteristics with quality attributes. Bansiya and Davis (2002) describe a methodology for defining a quality model for object-oriented design (QMOOD), based on Dromey (1995). The methodology is tested for applicability of the object-oriented modelling language C++. Weiss and Basili (1985) and Basili et al. (1994) propose the Goal/Question/Metric (GQM) paradigm, which describes how to derive metrics from questions to be answered in order to achieve a specific goal. Chidamber and Kemerer (1994) define six general metrics for software development with the aid of object-oriented design. Basili et al. (1996) adapted five of these metrics to predict class fault-proneness in early stages of the software system lifecycle. Lindland et al. (1994) describe a framework based on quality goals and the means to achieve them in conceptual modelling. Their framework is divided into three linguistic concepts: Syntax, Semantics and Pragmatics. Briand et al. (1996) define software measurement concepts like size, length, complexity, system cohesion and system coupling. These concepts are compared with other relevant concepts, for example the complexity measures of Weyuker (1988). Ernadote (2018) defines quality entities that indicate the assessed quality state of model artefacts and can be visualised in a dashboard. Stvilia et al. (2007) define a framework for information quality assessment that can serve as a starting point to develop a company-specific information quality framework. Doan and Gogolla (2019) define model metrics for UML models using OCL (Object Constraint Language) to implement quantifying design metrics and identify bad design decisions. The approaches are compared based on success factors, which are derived from literature and completed from a practical point of view within a workshop with four modelling experts.

These approaches are compared based on the following success factors, which are derived from literature (Kaiser et al., 2007; Watts et al., 2009; Rohweder et al., 2008) and from expert interviews:

Adaptability: The metrics must be adaptable within the application context.

Feasibility: The metrics must be based on existing input parameters that can be determined from modelling approaches.

Interpretability: The results of the metrics must be interpretable by the affected stakeholders.

Reproducibility: Metrics must provide reproducible results regardless of people, timing and interpretation.

Conformity to standards: The metrics must be usable to demonstrate compliance with standards.

MBSE Application: The metrics must be applicable for information from MBSE system models.

Maturity level: The metrics should be adapted to the maturity level of the system model.

Table 2. Comparison matrix with metrics approaches and success factors

Success factors / Metric approaches	Adaptability	Feasibility	Interpretability	Reproducibility	Conformity to standards	MBSE Application	Maturity level	Evaluation score
Quality View (Lange and Chaudron, 2005)	◐	●	◐	◐	○	●	◐	4
SD Metrics (Lange et al., 2007)	◐	◐	◐	◐	○	●	●	4
Software product quality metrics (Dromey, 1995)	◐	◐	◐	◐	●	○	◐	3,5
QMOOD (Bansiya and Davis, 2002)	◐	◐	●	◐	◐	●	◐	4,5
GQM (Weiss and Basili, 1985; Basili et al., 1994)	●	◐	●	◐	●	●	◐	5,5
6 metrics (Chidamber and Kemerer, 1994)	◐	◐	●	◐	◐	○	○	3
Metrics for class fault proneness (Basili et al., 1996)	◐	◐	●	●	◐	○	○	3,5
Framework for quality goals (Lindland et al., 1994)	◐	●	◐	◐	○	◐	◐	3,5
SW measurement concepts (Briand et al., 1996)	◐	◐	●	●	◐	○	◐	4
Quality entities (Ernadote, 2018)	◐	◐	●	●	◐	●	◐	5
Framework IQ assessment (Stvilia et al., 2007)	◐	●	◐	●	◐	●	◐	5
UML Model metrics (Doan and Gogolla, 2019)	◐	◐	●	●	◐	●	●	4,5

In the comparison of all approaches, the GQM process (Weiss and Basili, 1985; Basili et al., 1994) has the highest evaluation score and fulfils the success factors adaptability, interpretability, conformity, MBSE adoption. In addition, the process is generic, understandable for appliers and independent of specific disciplines. However, the GQM process has shortcomings regarding the adaptability, processual integration, and conformity with standards. Therefore, the underlying GQM paradigm is chosen as foundation to define a new method for assessing the quality of system models using company-specific metrics. Based on the GQM paradigm and the concept of control questions from the VDI 2206:2021 (VDI/VDE, 2021), four new methodical steps are defined:

- **Identify relevant MBSE processes**: In this step, the company's current modelling methodology is analysed. The methodology is divided into process steps. If there is no MBSE methodology used, relevant processes from existing methodologies like OOSEM or SysMOD can be used as starting point. The result of this step is a list of relevant process steps along a timeline, which can be used to evaluate the model quality.
- **Define control points within the processes**: Based on the relevant process steps, control points are defined that are embedded in the process to check the model quality. For each control point, the stakeholder group responsible for the assessment is defined. The control point can be mandatory or non-mandatory, depending on their relevance to the test. In addition, a target is defined by the specified stakeholders that describe the required model quality and maturity at the point in time. The result is a set of control points, associated targets, and the definition of responsible persons.
- **Derive metrics for each control point**: Based on the objective of the control point, a set of control questions is derived and metrics are associated to establish a quantitative evaluation of the system model. All relevant stakeholders are involved in deriving the metrics, as they want to describe and check relevant quality characteristics from their point of view. The result of this step are a set of control questions per control point and a set of metrics per control question.
- **Define guidelines for a standardised quality assessment**: To ensure a standardised quality assessment, framework conditions and rules are defined. These include the type of quality assessment. For example, in a design review this refers to the definition of necessary persons and the definition of the necessary information. The result of this step is a standardised framework to be followed when performing the quality assessment.

To ensure the applicability, the main elements are comprised in a template, which is applied in the case example in section 6. The four steps of the methods are illustrated in Figure 3.

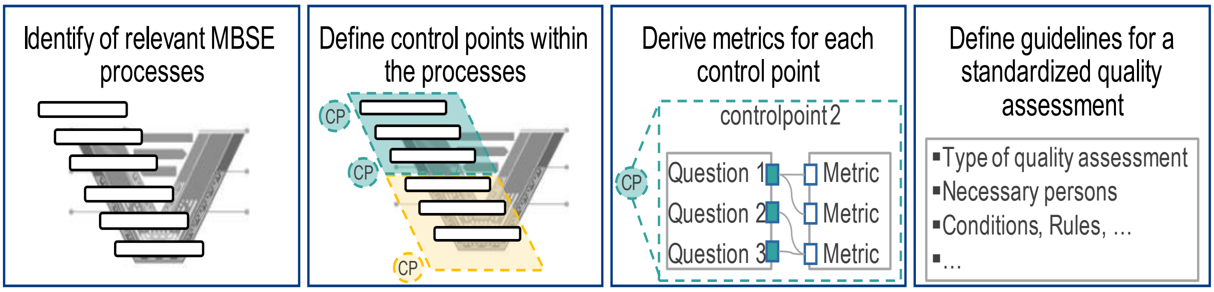

Figure 3. Method for assessing the information quality of MBSE system models

6. Industrial application

The method was applied to a company in the German automotive industry. In the first step, the company's existing modelling method was recorded and analysed using Input-Process-Output (IPO) Diagrams (David D. Walden et al., 2015). The IPO diagrams show how the necessary inputs are transformed into outputs through the activities of the process. The artefacts in the IPO diagrams are represented by underlying model types need to be identified, for example a requirement table in IBM DOORS or a SysML block definition diagram (bdd) in IBM Rhapsody. In the second step, the identified processes of the modelling method are mapped to the V-model of the VDI/VDE guideline 2206:2021 (VDI/VDE, 2021). Mapping to the V-model helps engineers in identifying checkpoints when an assessment of the model quality of the MBSE system model is required. In addition, the V-model serves as a common basis for understanding the development process from different stakeholders perspectives. Based on the mapping, the company's modelling process is divided into two parts, an acquisition phase and a development phase. In the acquisition phase, a first draft of the system model must be developed to convince the customer of the technical feasibility and abilities to receive the development order from the OEM. In the development phase, the draft of the system model is further specified to ensure that the technical maturity level is met.

Based on mapping and separating phases, four control points have been defined, which are derived in step 2. The first one is after the acquisition, the second is before the start of the development, the third is between the logical architecture and the physical architecture and the fourth is after completing the physical architecture. Based on the GQM method, for every control point a goal is derived and responsible roles are defined. For example, the goal of control point 1 is to deliver an initial system model with all relevant features to the OEM in a short time. The focus of the modelling in this phase is to identify every feature on a high level in a limited time period. In addition, responsible roles for the first control point are sales manager, project manager, system architect and feature owner. Each of these roles has the ability to derive control questions from their point of view. For example, the project manager might ask, "Are all approved requirements linked to features and epics?". In the third step, metrics are derived for each control question. As example, to evaluate the question from above, a useful metric would be the percentage of requirements that have links to a feature. The fourth step is to define the framework for the standardised implementation of the company-specific quality assessment is defined. The quality assessment has to be conducted in design reviews for each of the system models, which is at one of the four control points. Based on the quality assessment, a standardised model quality can be ensured. In addition, metrics can be reused to answer other questions, for example if the modelling process will be adapted in the future. In figure 4, the result of the application of the method for the first control point is demonstrated. As mentioned above, to apply the method, a template is developed which can be filled in for each control point. The template contains categories' goal, mapping to the V-model, control questions, metrics and additional information. The goal definition specifies the purpose, the artefact under consideration, the process, the roles involved and the timing. In the additional information, the date of the test, the name of the test procedure, the project, the document number, the path in the database and other comments can be stored. The template serves as a generic form and can be tailored for (SW tool) formats.

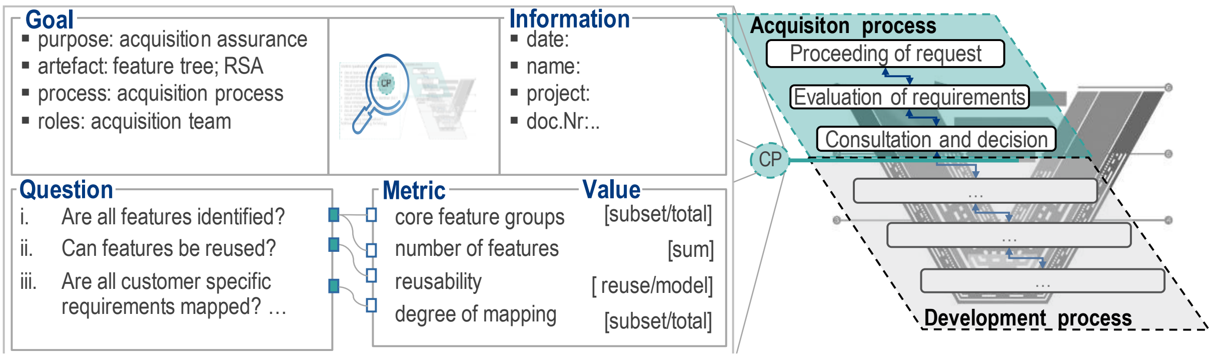

Figure 4. Model with control questions and metrics after the acquisition process

7. Evaluation of the method

The evaluation is divided into two phases, the application evaluation and the success evaluation (Blessing and Chakrabarti, 2009). Application evaluation aims to determine whether the method can be used for the intended task, while the success evaluation aims to determine whether the support has the expected impact for the user (Blessing and Chakrabarti, 2009). For the application evaluation, the results from section 6 are evaluated. The method was applied in industrial context. During three workshops with system architects, 50 control questions were derived and associated to four control points. Based on the set of control questions 71 metrics were derived, which are used to assess the information quality of the system model in different phases. The additional template supports the user in dealing with the number of metrics required to address the concerns of all participants and to provide the necessary transparency. For the success evaluation, the modelling experts are interviewed and asked to evaluate the derived success factors. The modelling experts confirm that all success factors are fulfilled: The *adaptability* of the generic method is ensured by defining company-specific control questions as the basis for deriving metrics. For the same reason, all the parameters needed to calculate the metrics can be derived from the information of the company-specific modelling process. *Feasibility* is ensured by the simplicity and short application time. The 71 metrics were derived in just three hours, based on the company's and role's specific needs. All stakeholders concerned are consulted in step 2 of the method, which ensures *interpretability*. The metrics are explicitly described in terms of formulae and values, which ensures *reproducibility*. In addition, the process integration supports reproducibility, as the point of assessment depends on the point of time and the model maturity. In the case study, the necessary evidence of compliance with A-SPICE was considered. This shows that the derived metrics can support the *conformity to standards* and regulations in the automotive industry. In addition, the central artefact for calculating the metrics is the MBSE system model, which was considered at different *levels of maturity* depending on the recurrence point of time of the checkpoints.

8. Conclusion

The paper describes a method for defining company-specific goals, questions, and metrics for assessing the quality of MBSE system models. The derived research questions are answered by a six-step scientific approach. Seven success factors for the systematic derivation of metrics are derived from literature and completed by industrial experts. Based on the derived success factors, a method for assessing the model quality of MBSE system models is derived. The method consists of four steps that need to be carried out for a successful derivation of company-specific metrics (RQ1): Identification of relevant MBSE processes, definition of control points within the process, derivation of metrics for each control point, definition of a standardised quality assessment. All inputs for the metrics evaluation come from existing information of the modelling process. The tailoring of the method to different phases and maturity levels of the system models are ensured by the anchoring of the control points to the MBSE processes (RQ2). After the development, the method is applied in a German automotive company and evaluated by the four associated modelling experts, which ensures the applicability of the method. The success evaluation underlines that the method fulfils all success factors. The feedback from the industry experts underlines that the results from the application project represent a direct added value for the company and that the method will also be used in the future. Overall, the method supports engineers in independently deriving company-specific or even project-specific metrics for evaluating the information quality of their system models. Due to the increasing relevance of system models, competitive advantages can be developed in this way. Future research includes further detailing of the process-related integration, for example through role models (Gräßler et al., 2021), as well as the transferability to support decisions regarding technical changes (Gräßler and Wiechel, 2023; Gräßler et al., 2023).

References

610.12-1990 IEEE Standard Glossary of Software Engineering Terminology, IEEE / Institute of Electrical and Electronics Engineers Incorporated, s.l.

Bansiya, J. and Davis, C.G. (2002), “A hierarchical model for object-oriented design quality assessment”, IEEE Transactions on Software Engineering, Vol. 28 No. 1, pp. 4–17, https://dx.doi.org/10.1109/32.979986.

Basili, V.R., Briand, L.C. and Melo, W.L. (1996), “A validation of object-oriented design metrics as quality indicators”, IEEE Transactions on Software Engineering, Vol. 22 No. 10, pp. 751–761.

Basili, V.R., Caldiera, G. and Rombach, H.D. (1994), “The goal question metric approach”, in Encyclopedia of software engineering.

Blessing, L.T.M. and Chakrabarti, A. (2009), DRM, a Design Research Methodology, 1. Auflage, Springer London, Guildford, Surrey, https://dx.doi.org/10.1007/978-1-84882-587-1.

Briand, L.C., Morasca, S. and Basili, V.R. (1996), “Property-based software engineering measurement”, IEEE Transactions on Software Engineering, Vol. 22 No. 1, pp. 68–86, https://dx.doi.org/10.1109/32.481535.

Chidamber, S.R. and Kemerer, C.F. (1994), “A metrics suite for object oriented design”, IEEE Transactions on Software Engineering, Vol. 20 No. 6, pp. 476–493, https://dx.doi.org/10.1109/32.295895.

David D. Walden, Garry J. Roedler and Kevin Forsberg (2015), “INCOSE Systems Engineering Handbook Version 4: Updating the Reference for Practitioners”, INCOSE International Symposium, Vol. 25 No. 1.

Debbabi, M., Hassaïne, F., Jarraya, Y., Soeanu, A. and Alawneh, L. (2010), Verification and validation in systems engineering: Assessing UML/SysML design models, 1. ed., Springer, Berlin, Heidelberg.

Delligatti, L. (2014), SysML distilled: A brief guide to the systems modeling language, Addison-Wesley, Upper Saddle River, NJ.

DIN (2015), DIN EN ISO 9000: Qualitätsmanagementsysteme - Grundlagen und Begriffe (ISO 9000:2015); No. 9000, 2015th ed., Beuth, Berlin.

Doan, K.-H. and Gogolla, M. (2019), “Quality Improvement for UML and OCL Models Through Bad Smell and Metrics Definition”, in 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Munich, Germany, 15.09.2019 - 20.09.2019, IEEE.

Dromey, R.G. (1995), “A model for software product quality”, IEEE Transactions on Software Engineering, Vol. 21 No. 2, pp. 146–162, https://dx.doi.org/10.1109/32.345830.

Ernadote, D. (2018), “A Framework for Descriptive Models Quality Assessment”, in 2018 IEEE International Systems Engineering Symposium (ISSE), Rome, 01.10.2018 - 03.10.2018, IEEE, pp. 1–7.

Friedenthal, S. and Burkhart, R. (2015), “Evolving SysML and the System Modeling Environment to Support MBSE”, INSIGHT, Vol. 18 No. 2, pp. 39–41, https://dx.doi.org/10.1002/inst.12020.

Friedenthal, S., Moore, A. and Steiner, R. (2015), A practical guide to SysML: The systems modeling language, 3. ed., Elsevier; Morgan Kaufmann, Amsterdam, Waltham, Mass.

Giraldo, F.D., España, S., Pastor, Ó. and Giraldo, W.J. (2018), “Considerations about quality in model-driven engineering”, Software Quality Journal, Vol. 26 No. 2, pp. 685–750, https://dx.doi.org/10.1007/s11219-016-9350-6.

Graessler, I., Wiechel, D. and Oleff, C. (2022), “Extended RFLP for complex technical systems”, in 2022 IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, 24.10.2022 - 26.10.2022, IEEE.

Gräßler, I. and Oleff, C. (2022), Systems Engineering: Verstehen und industriell umsetzen, Springer, Berlin, Heidelberg, https://dx.doi.org/10.1007/978-3-662-64517-8.

Gräßler, I. and Wiechel, D. (2023), “Customized impact analyses for technical engineering changes”, in 2023 18th Annual System of Systems Engineering Conference (SoSe), Lille, France, 14.06.2023 - 16.06.2023, IEEE.

Gräßler, I., Wiechel, D., Koch, A.-S., Sturm, T. and Markfelder, T. (2023), “Methodology for Certification-Compliant Effect-Chain Modeling”, Systems, Vol. 11 No. 3, p. 154, https://dx.doi.org/10.3390/systems11030154.

Gräßler, I.; Özcan, D.; Preuß, D.: "AI-based extraction of requirements from regulations for automotive engineering DS 125: Proceedings of the 34th Symposium Design for X (DFX2023), 14 2023, S. 163–172.

Gräßler, I., Wiechel, D. and Pottebaum, J. (2021), “Role model of model-based systems engineering application”, IOP Conference Series: Materials Science and Engineering, Vol. 1097 No. 1, p. 12003.

Hause, M. (2011), ““Are we there yet?” Assessing Quality in Model Based Systems Engineering”, INCOSE International Symposium, Vol. 21 No. 1, pp. 510–522, https://dx.doi.org/10.1002/j.2334-5837.2011.tb01221.x.

Henderson, K., McDermott, T., van Aken, E. and Salado, A. (2023), “Towards Developing Metrics to Evaluate Digital Engineering”, Systems Engineering, Vol. 26 No. 1, pp. 3–31, https://dx.doi.org/10.1002/sys.21640.

International Council on Systems Engineering (INCOSE) (2007), INCOSE Systems Engineering Vision 2020, Seattle, WA.

ISO/IEC, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models No. 25010:2011, ISO.

Kaiser, M., Klier, M. and Heinrich, B. (2007), “How to Measure Data Qualityß - A Metric-Based Approach”, in International Conference on Information Systems (ICIS).

Kilkenny, M.F. and Robinson, K.M. (2018), “Data quality: "Garbage in - garbage out"”, Health information management journal of the Health Information Management Association of Australia, Vol. 47 No. 3.

Krcmar, H. (2015), Informationsmanagement, 6., überarb. Aufl., Springer Gabler, Wiesbaden.

Lange, C. and Chaudron, M. (2005), “Managing Model Quality in UML-Based Software Development”, in 13th IEEE International Workshop on Software Technology and Engineering Practice (STEP'05), Budapest, Hungary, IEEE, pp. 7–16, https://dx.doi.org/10.1109/STEP.2005.16.

Lange, C.F., Wijns, M.A. and Chaudron, M.R. (2007), “MetricViewEvolution: UML-based Views for Monitoring Model Evolution and Quality”, in 11th European Conference on Software Maintenance and Reengineering (CSMR'07), IEEE, https://dx.doi.org/10.1109/csmr.2007.32.

Lehner, D., Vamberszky, S., Wieland, K. and Siegl, D. (2022), “Git-basiertes Qualitätsmonitoring und von Systems Engineering Modellen”, in Koch, W., Wilke, D., Dreiseitel, S. and Kaffenberger, R. (Eds.), Tag des Systems Engineering 2022.

Lindland, O.I., Sindre, G. and Solvberg, A. (1994), “Understanding quality in conceptual modeling”, IEEE Software, Vol. 11 No. 2, pp. 42–49, https://dx.doi.org/10.1109/52.268955.

Mohagheghi, P. and Aagedal, J. (2007), “Evaluating Quality in Model-Driven Engineering”, in International Workshop on Modeling in Software Engineering (MISE'07: ICSE Workshop 2007), Minneapolis, 2007.

Object Management Group (2004), “OMG: Unified Modeling Language Infrastructure Specification, Version 2.0”, available at: https://www.omg.org/spec/UML/2.0/Superstructure/PDF (accessed 12 October 2020).
Object Management Group (2019), “OMG Systems Modeling Language (OMG SysML™)”, available at: https://sysml.org/.res/docs/specs/OMGSysML-v1.6-19-11-01.pdf.
Parra, E., Alonso, L., Mendieta, R. and La Vara, J.L. de (2019), “Advances in Artefact Quality Analysis for Safety-Critical Systems”, in 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Berlin, Germany, 27.10.2019 - 30.10.2019, IEEE, pp. 79–84, https://dx.doi.org/10.1109/ISSREW.2019.00047.
Pottebaum, J. and Gräßler, I. (2020), “Informationsqualität in der Produktentwicklung: Modellbasiertes Systems Engineering mit expliziter Berücksichtigung von Unsicherheit”, Konstruktion, Vol. 2020 No. 11-20.
Rohweder, J.P., Kasten, G., Malzahn, D., Piro, A. and Schmid, J. (2008), “Informationsqualität — Definitionen, Dimensionen und Begriffe”, in Hildebrand, K., Gebauer, M., Hinrichs, H. and Mielke, M. (Eds.), Daten- und Informationsqualität, Vieweg+Teubner, Wiesbaden, pp. 25–45, https://dx.doi.org/10.1007/978-3-8348-9266-9_2.
Rohweder, J.P., Kasten, G., Malzahn, D., Piro, A. and Schmid, J. (2021), “Informationsqualität – Definitionen, Dimensionen und Begriffe”, in Hildebrand, K., Gebauer, M. and Mielke, M. (Eds.), Daten- und Informationsqualität, Springer Fachmedien Wiesbaden, Wiesbaden, pp. 23–43.
Sargent, R.G. (2013), “Verification and validation of simulation models”, Journal of Simulation, Vol. 7 No. 1, pp. 12–24, https://dx.doi.org/10.1057/jos.2012.20.
Stvilia, B., Gasser, L., Twidale, M.B. and Smith, L.C. (2007), “A framework for information quality assessment”, Journal of the American Society for Information Science and Technology, Vol. 58 No. 12, pp. 1720–1733.
Ulrich, H. (1982), “Anwendungsorientierte Wissenschaft”, Die Unternehmung, Vol. 36, pp. 1–10.
United Nations (2021), Regulation - Software update and software update management system No. 156.
VDA QMC Working Group 13 / Automotive SIG (2017), Automotive SPICE: Process Reference Model Process Assessment Model No. Automotive SPICE Version 3.1, 656th ed.
VDI/VDE (2021), VDI 2206 Entwicklung mechatronischer und cyber-physischer Systeme No. 2206, Beuth Verlag GmbH, Düsseldorf.
Wang, R.Y. and Strong, D.M. (1996), “Beyond Accuracy: What Data Quality Means to Data Consumers”, Journal of Management Information Systems, Vol. 12 No. 4, pp. 5–33, https://dx.doi.org/10.1080/07421222.1996.11518099.
Watts, S., Shankaranarayanan, G. and Even, A. (2009), “Data quality assessment in context: A cognitive perspective”, Decision Support Systems, Vol. 48 No. 1, pp. 202–211, https://dx.doi.org/10.1016/j.dss.2009.07.012.
Weilkiens, T. (2014), Systems Engineering mit SysML/UML: Modellierung, Analyse, Design, 3rd ed., dpunkt, Heidelberg.
Weiss, D.M. and Basili, V.R. (1985), “Evaluating Software Development by Analysis of Changes: Some Data from the Software Engineering Laboratory”, IEEE Transactions on Software Engineering, SE-11 No. 2.
Weyuker, E.J. (1988), “Evaluating software complexity measures”, IEEE Transactions on Software Engineering, Vol. 14 No. 9, pp. 1357–1365, https://dx.doi.org/10.1109/32.6178.