Quality Assurance for Machine Learning – an approach to function and system safeguarding

Alexander Poth
Volkswagen AG
Wolfsburg, Germany
alexander.poth@volkswagen.de

Burkhard Meyer
Audi AG
Ingolstadt, Germany
burkhard.mayer@audi.de

Peter Schlicht
Volkswagen AG
Wolfsburg, Germany
peter.schlicht@volkswagen.de

Andreas Riel
Grenoble Alps University
Grenoble, France
andreas.riel@grenoble-inp.fr

Abstract— In an industrial context, high software quality is mandatory in order to avoid costly patching. We present a state of the art analysis of approaches to ensure that a specific Artificial Intelligence (AI) model is ready for release. We analyze the requirements a Machine Learning (ML) system has to fulfill in order to comply with the needs of an automotive OEM. The main implication for projects relying on ML is a holistic assessment of possible quality risks. These risks may stem from implemented ML models and spread into the delivery. We present a methodological quality assurance (QA) approach and its evaluation.

Keywords— artificial intelligence, machine learning, quality management, quality assurance, risk management

I. INTRODUCTION

Software engineering as well as corresponding QA and testing approaches have been developed and enhanced over decades. Unlike traditional software, the outcome of a connectionist ML system is highly intractable and non-transparent due to its mathematical complexity, as well as the dependency of its behavior on both the training and in-use data and processes. This gives rise to the necessity of new measures for understanding and explaining ML to the level of rigor required by QA requirements [1]. As a holistic approach (“a system is more than just the assembly of its parts”), a connectionist ML algorithm is locally deterministic, well understood and explainable, however unforeseen behavior may emerge at a global level.

Investigating modern ML development organizations and approaches, we made the following key observations:

- ML is a new and emerging type of software still missing adequate quality measurement metrics, control and assurance techniques [2].
- The general class of software systems with no reliable test oracle available is sometimes known as “non-testable programs”. ML-based software belongs to this class of systems.
- Creating mature high-quality products and services is very hard. QA principles should be pro-actively incorporated by design beforehand, instead of reacting on quality claims during product/service use. Quality has to be inherent to the product/service by design instead of being an add-on introduced at later life-cycle stages.

These key observations imply four essential questions to be addressed by a QA approach to ML software:

1. How to identify and estimate quality risks?
2. How to define adequate quality assurance activities to mitigate or reduce quality risks?
3. How to assure being the “right track” to generate customer confidence in the AI/ML software/model at release time?
4. How to deal with non-deterministic behavior of algorithms in QA approaches?

Our systematic methodical approach presented in this article helps to answer these questions systematically. The proposed evAIa (evaluate AI approaches) is mainly based on the current state of the art, as well as a mindset to future development in the AI domain especially for its sub-domain ML and their QA and test methods. As currently many industrial products and services are developed with ML-based components, the quality aspect for these new type of data driven functionality and behavior needs adequate safeguarding and QA. The ML-based components can have the objective to add a feature or functionality to a product or service. Furthermore, the ML-based components can be a core part of the product or service offer. In both cases, adequate safeguarding is needed. Depending on the ML-based component, the safeguarding scope has to be on the function or system level. Currently established systematic approaches on industry level are available like the ISO/IEC/IEEE 29119 series for software testing standard, but they are missing a link to AI and ML safeguarding and QA. Our approach to AI QA has been designed to meet the following key requirements:

- Support the life cycle from development to production;
- Fulfill business needs as well as technical aspects;
- Be independent of a particular development or operation process model (in order to assure its applicability in both e.g. V-model and Scrum contexts);
- Be usable based on a hands-on guide by the responsible teams;
• Reflect the state-of-the-art;
• Extend easily to new insights or future technologies.

In order to validate the completeness and practical relevance of our approach, we evaluated our questionnaire on a series of case and field studies (section 4) within the Volkswagen Group. The feedback of these AI and ML experts reflects the different working methods and technologies, which are used in the different brands and domains. Currently the approach is offered via the group wide AI working group and their knowledge base. The approach is periodically reflected and iteratively enhanced with the feedback of AI and ML experts.

Section 2 introduces related work, section 3 presents the evAla method, section 4 evaluates evAla in the Volkswagen AG and section 5 concludes and give some outlook.

II. RELATED WORK

Recently, research on the software development process and the data processing of AI pipelines has received considerable attention when researchers attempted to understand and improve AI approaches. So far, software quality management is not yet commonly established in this domain. In this section, we present a selection of relevant quality aspects from published work in order to gain an impression of the state of the art in QA and testing in the context of AI approaches. This state of the art will be the base for a generic QA method for AI approaches with focus on ML and their products.

To identify and collect the state of the art, we conducted a literature analysis in the first quarter of 2018 focusing on practical approaches and methods and aligned with [3]. The search was conducted in IEEE Explore, Springer and Elsevier. The used search term was (“AI” or “artificial intelligence” or “ML” or “machine learning”) and (“safeguarding” or “quality assurance” or “QA”). We filtered the results by the following criteria:

• Can the content be integrated in a generic QA approach?
• Is the content proven in use?
• Can aspects of the content be rephrased into practical questions or offer a way of measurement or indicator?

The relevant results are consolidated in Table 1 and subsequent tables in chapter 3. To summarize the insights of the literature analysis, we can conclude that for specific aspects of ML and AI quality related approaches exist. However, we could not find any holistic safeguarding approach for ML-based components. Especially the holistic QA in the context of the product or service life-cycle of ML-based components apparently has not been addressed systematically. This result gave the impetus for the development of an approach to address our demand.

TABLE I. QA TOPICS REGARDING AI

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description/Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test your features and data</td>
<td>Identify quality risks in the product stemming from AI and ML models [4].</td>
</tr>
<tr>
<td>Know your product’s quality risks</td>
<td></td>
</tr>
</tbody>
</table>

For conventional software, several QA metrics may be applied to evaluate a system. Examples are coverage metrics like C0 (statement coverage) or C1 (branch coverage). However, for AI, these metrics are only of limited use because the model performance is mostly driven by learning data it is difficult to
use these metrics or transfer them easily to the ML-based components. Thus this work does not focus on any specific metric, however if a metric is available, it will be useful to measure the complexity and quality of a system. To demonstrate the value of our approach evAla compared to classic quality assurance approaches, we focused on the topics listed in Table 1.

III. THE evAla METHOD

A. Context and Overview of evAla

The evAla (evaluate AI approaches) method reflects state of the art approaches for QA in AI projects and presents current recommendations to data scientists and engineers for ensuring the capabilities of their AI models and making limitations transparent. EvAla is based on product risk evaluation, a questionnaire about the used AI-approaches, QA method recommendations to mitigate specific product risks caused by the AI-approach, as well as a transparency report to show the mitigation and residual risk reached with the evAla approach.

![Fig. 1. The application sequence of evAla](image)

Additionally, the questionnaire is mapped to the ISO/IEC 25010:2011 to show the systematic refinement of the standardized characteristics for the AI and ML domain-based products and services as required in formal and regulated environments. The mapping of each question to the related main characteristic of the standard is made in Table 5. As some questions easily can be mapped to more than one characteristic, only the best fitting one has been mapped for simplification. The mapping shows that not all ISO characteristics are addressed by evAla because security and usability are not specific to ML-based components. However, these topics are also relevant for product and service deliveries. We recommend aligning project work with the applicable established domain standards.

B. Detailed evAla Method Description

The evAla method is designed to support quality engineers and developers to realize quality by design in four sequential steps (Figure 1). In the first step, the product/service risks are evaluated against potential quality issues and a mitigation scope is set. In the second step, the product/service team answers the evAla questionnaire to systematically check weaknesses of the ML learning components of their product or service. In the third step, decisions are made about mitigation actions. These include mitigation by design (quality by design) or specific tests for verification and validation. In the last step, the defined actions are tracked and documented to ensure adequate compliance documentation for the entire product/service life cycle.

C. Risk Evaluation

Product/service teams can make a systematic product quality risk (PQR) evaluation with the PQR method [11] and a more elaborated version for practical use as a workshop kit. Volkswagen offers a workshop kit to all product teams for effective PQR elaboration [12]. With the systematically derived quality risks and their classification, the relevant functions or features of the product or service can be identified for focusing on mitigation of the product/service specific quality risks. The PQR approach is used during the first step of the evAla sequence (figure 1). The outcome of the PQR analysis is used to focus the questionnaire on the product/service team-specific quality risks (step two in the evAla sequence). This risk-based QA approach helps to align safeguarding resources with the most relevant quality issues.

D. Questionnaire

The following tables show the questionnaire, which is used to reflect the relevant AI-based functions or features of the product or service. The tables go through the three core AI lifecycle phases of data pre-processing (Table 2), implementing (Table 3), and serving (Table 4). Volkswagen provides the quality engineers and developers spreadsheets to evaluate, comment and remark each question for an adequate documentation of the ML safeguarding.

The tables are based on literature which has elaborated QA or safeguarding approaches in the ML domain and on the experience of the ML experts who are worked and reviewed the design and development of the evAla approach. The development was oriented on the design science research approach [13].

<table>
<thead>
<tr>
<th>Topic</th>
<th>Aspect (indicators)</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adequacy</td>
<td>Degree of involvement of validation experts in the design phases of the interacting systems, deployed (GPU-) hardware and software modules; definition of validation criteria and scenarios for AI and ML algorithm requirements; existence of both requirements (includes stories)</td>
<td>1.1 Have algorithms and training- and validation-data been co-designed?</td>
</tr>
<tr>
<td></td>
<td>1.2 Are the requirements to training- and validation-data clearly defined?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.3 Which parts of the system shall be</td>
<td></td>
</tr>
</tbody>
</table>
### TABLE III. QUESTIONNAIRE FOR AI BASED PRODUCTS/SERVICES – TEST YOUR IMPLEMENTATION

<table>
<thead>
<tr>
<th>Topic</th>
<th>Aspect (indicators)</th>
<th>Questions</th>
<th>Topic</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training and testing chain</td>
<td>Code and configurations are under version control, test-suites for the code are established; architecture and requirements are documented; the established software development process is fulfilled [17].</td>
<td>1.4 Is the software for the training and testing chain developed according to relevant/domain specific QA guidelines?</td>
<td>Model fitting</td>
<td>(Pre-trained) models connected to chains are broken down into “model-units” [21]; each model (-unit) is checked for over-/under-fitting effects; model-chains are integrated and step-wise checked; entire model algorithms developed according to relevant/domain specific QA guidelines?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.5 What kind of (inherent) bias does the data have?</td>
<td></td>
<td>2.3 Are the code and AI frameworks/libraries with their configuration under version control to reproduce outcomes?</td>
</tr>
<tr>
<td>Bias</td>
<td>Source of raw data before processing is reliable (no manipulation etc.); relevant bias aspects are identified (cultural-bias, locality-bias, social-bias etc.) [4].</td>
<td>1.6 Why is the dataset representative for the ML algorithm (learning and testing)?</td>
<td></td>
<td>2.4 How much of the model can be “white boxed” to validate it?</td>
</tr>
<tr>
<td>Completeness</td>
<td>Use cases define boundaries (check with misuse cases); External drivers about data completeness and boundaries are identified (example: security attack vectors are changing over time); adversarial examples, fitting to the (business) domain [14].</td>
<td>1.7 What boundaries or limitations does the data have?</td>
<td></td>
<td>2.5 What useful checks on a “white boxed” model? How does the model/system react to the white boxing outcome?</td>
</tr>
<tr>
<td>Process chain</td>
<td>Bias change/added via processing (labeling-criteria, filtering-rules, concatenation-rules etc.); training and test data are not mixed/enriched or under defined and proven aspect “enriched” (under/overfitting aspects are identified); process chain under configuration management (code, parameters etc. and their processing artefacts input &amp; output data to proof determinism) [15].</td>
<td>1.8 What is the criteria set for a complete over/under fitting aspect of the data inside the boundaries?</td>
<td></td>
<td>2.6 Which kind of “hot-spots” are acceptable?</td>
</tr>
<tr>
<td>Regulations/Compliance</td>
<td>Different aspects of regulation are listed (user-based, country-based, usage-based etc.); relevant data protection laws (GDPR etc.) for the aspects are listed; assurance that the training and serving phase is aligned with the current data privacy and protection laws (anonymization, masking, deleting etc.); confirmation that the product/service “is legal” [16].</td>
<td>1.9 Are completeness and boundaries constant over time or can external drivers change them?</td>
<td></td>
<td>2.7 Which hyper-parameters are available?</td>
</tr>
</tbody>
</table>

**Visual fields of machine learning**

- **Bias**
  - **Bias**
  - **Model transparency**
    - Visualization tools for learning-steps or layers etc. are used; models are checked for “hot-spots” (example: deactivation of high connected nodes and their impact to the output can impact the robustness of the model and its usage context); evaluation for over-/under-fitting [18].
    - Relevant hyper parameters are identified; value ranges of the relevant hyper parameters are evaluated and tuned to optimize the outcome for the product/service context; the ground truth is identified and evaluated for the product/service context [16].
    - Gap between demanded product/service specific aspects and model is identified; methods [8] and [19] for checking robustness are applied; versions are run in parallel (diffy mode).
    - Established methods of derivation of training- and test-data are used; separation of the data into training and test data is well defined and under configuration management; sufficient completeness of dataset is checked [20].

- **Regulations/Compliance**
  - **Regulations/Compliance**
  - **Model completeness**
    - Licenses check of the model and framework/library is compliant for productive serving; references to users are checked; in case of open source: the project is “active” on e.g. github and its license fit to product; bugs are fixed fast; transparency of test activities is given.
    - How reliable (about the usage domain quality aspects) are third party included AI and ML models and frameworks/libraries? 
    - How is the training-data derived?
    - 2.13 Is the model “white boxed” or “black boxed” model? How can the model react to the input data? Is the model-enriched or under defined?

- **Understanding the learning process**
  - **Understanding the learning process**
  - **Model fitting**
    - (Pre-trained) models connected to chains are broken down into “model-units” [21]; each model (-unit) is checked for over-/under-fitting effects; model-chains are integrated and step-wise checked; entire model-chains are tested against relevantQA guidelines.
    - What impact will over-/under-fitting have? Is this being monitored?
    - How can chains be broken down for model-“unit”-testing?
chain is checked end to end, critical model (unit) is cross-checked with other model implementations (example: Keras can use a TensorFlow and Mxnet implementation of a algorithm for cross-checking model behavior) or checked by more simple models to assure not to rely on a special implementation or a side-effect of an implementation bug.

<table>
<thead>
<tr>
<th>ISO Characteristic</th>
<th>evAiA Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Suitability</td>
<td>1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.13, 2.15</td>
</tr>
<tr>
<td>Performance Efficiency</td>
<td>3.2, 3.7, 3.8</td>
</tr>
<tr>
<td>Usability</td>
<td>Not explicitly addressed by evAiA</td>
</tr>
<tr>
<td>Compatibility</td>
<td>2.16, 2.17</td>
</tr>
</tbody>
</table>

E. QA recommendations

Table 6 correlates with the aspects of Tables 2, 3 and 4 in that it proposed related methods or approaches. Their adequateness depends on the business goals and the desired tradeoff between quality risk mitigation and effort to the benefit of the action. Rather than a rule based approach, [9] consider it more as a practice collection for QA inspiration. Furthermore, a business goal often is not only to mitigate quality risks – a business objective can be to push some specific ISO characteristics of Table 5. Step 3 of the evAiA sequence (figure 1) balances the selection of the safeguarding measures. Depending on the specific safeguarding strategy of the product or service, the measures of table 6 are selected by the business related quality risks and focused characteristics (table 5). For both the tables 2, 3 and 4 provide the most relevant topics and issues that have to be addressed in the safeguarding measures which are selected from table 6. To support the engineering of selected quality characteristics, the corresponding questions should be handled with priority. In any case, any QA measures have to include the verification of behavior in case of incorrect or unacceptable data.

F. Transparency report

Based on the identified risks of the risk evaluation and the selected QA recommendations, the outcome of the evAiA approach is a list of actions that make transparent what kind of quality improvements are possible and/or should be done to have a state of the art AI and ML based product or service. Wherever the state of the art is adequate to the needs, no further quality improvement actions are proposed.

| Table 6 correlates with the aspects of Tables 2, 3 and 4 in that it proposed related methods or approaches. Their adequateness depends on the business goals and the desired tradeoff between quality risk mitigation and effort to the benefit of the action. Rather than a rule based approach, [9] consider it more as a practice collection for QA inspiration. Furthermore, a business goal often is not only to mitigate quality risks – a business objective can be to push some specific ISO characteristics of Table 5. Step 3 of the evAiA sequence (figure 1) balances the selection of the safeguarding measures. Depending on the specific safeguarding strategy of the product or service, the measures of table 6 are selected by the business related quality risks and focused characteristics (table 5). For both the tables 2, 3 and 4 provide the most relevant topics and issues that have to be addressed in the safeguarding measures which are selected from table 6. To support the engineering of selected quality characteristics, the corresponding questions should be handled with priority. In any case, any QA measures have to include the verification of behavior in case of incorrect or unacceptable data. Based on the identified risks of the risk evaluation and the selected QA recommendations, the outcome of the evAiA approach is a list of actions that make transparent what kind of quality improvements are possible and/or should be done to have a state of the art AI and ML based product or service. Wherever the state of the art is adequate to the needs, no further quality improvement actions are proposed.

| Table 6 correlates with the aspects of Tables 2, 3 and 4 in that it proposed related methods or approaches. Their adequateness depends on the business goals and the desired tradeoff between quality risk mitigation and effort to the benefit of the action. Rather than a rule based approach, [9] consider it more as a practice collection for QA inspiration. Furthermore, a business goal often is not only to mitigate quality risks – a business objective can be to push some specific ISO characteristics of Table 5. Step 3 of the evAiA sequence (figure 1) balances the selection of the safeguarding measures. Depending on the specific safeguarding strategy of the product or service, the measures of table 6 are selected by the business related quality risks and focused characteristics (table 5). For both the tables 2, 3 and 4 provide the most relevant topics and issues that have to be addressed in the safeguarding measures which are selected from table 6. To support the engineering of selected quality characteristics, the corresponding questions should be handled with priority. In any case, any QA measures have to include the verification of behavior in case of incorrect or unacceptable data. Based on the identified risks of the risk evaluation and the selected QA recommendations, the outcome of the evAiA approach is a list of actions that make transparent what kind of quality improvements are possible and/or should be done to have a state of the art AI and ML based product or service. Wherever the state of the art is adequate to the needs, no further quality improvement actions are proposed.

| Table 6 correlates with the aspects of Tables 2, 3 and 4 in that it proposed related methods or approaches. Their adequateness depends on the business goals and the desired tradeoff between quality risk mitigation and effort to the benefit of the action. Rather than a rule based approach, [9] consider it more as a practice collection for QA inspiration. Furthermore, a business goal often is not only to mitigate quality risks – a business objective can be to push some specific ISO characteristics of Table 5. Step 3 of the evAiA sequence (figure 1) balances the selection of the safeguarding measures. Depending on the specific safeguarding strategy of the product or service, the measures of table 6 are selected by the business related quality risks and focused characteristics (table 5). For both the tables 2, 3 and 4 provide the most relevant topics and issues that have to be addressed in the safeguarding measures which are selected from table 6. To support the engineering of selected quality characteristics, the corresponding questions should be handled with priority. In any case, any QA measures have to include the verification of behavior in case of incorrect or unacceptable data. Based on the identified risks of the risk evaluation and the selected QA recommendations, the outcome of the evAiA approach is a list of actions that make transparent what kind of quality improvements are possible and/or should be done to have a state of the art AI and ML based product or service. Wherever the state of the art is adequate to the needs, no further quality improvement actions are proposed.
IV. EvAlA In Producti Development

A. Context of the evaluation project

This example applies evAlA to a cloud product/service of the Volkswagen Group IT cloud [24], which uses AI for anomaly detection. We use the questions listed in the introduction to demonstrate the benefit of the systematic PQR analysis and application of the questionnaires to the product team even in cases where the evaluation is done after development start.

1) How can we estimate the quality risks?

Based on the product vision and the product features, the PQR analysis is set up. The outcomes are technical (TPQR) and methodical (MPQR) product quality risks. The following is an extract of the main risks identified by our PQR-analysis:

TPQR 1 – Inadequate implementation: The implementation of machine learning based applications is challenging because it involves various successive transformation processes that have to fit together smoothly. The most risky part is the data-preprocessing step that involves the log categorization and the feature representation. An incorrectly implemented transformation leads to insufficient machine learning models and thus to poor predictive capability.

TPQR 2 – Inadequate deployment: Due to the application’s complexity, there is a high risk that the data representations in the training mode and in the predictive mode are not mapped equally. However, in order to deploy machine-learning models in production, it is crucial to preprocess the data in the same way as in the training mode. Otherwise, the model is not able to predict anything, because it does not receive the data in the required format.

MPQR 1 – Inadequate data representation: The usage of log files as an input for machine learning algorithms is challenging, since we have to deal with heterogeneous, unstructured data. Various preprocessing steps are required to transform the raw log data into a numerical representation. However, this complex data structure along with the transformations may bear the risk of being insufficient in terms of error prediction.

MPQR 2 – Inadequate model quality: The quality of machine learning models depends on various hyper-parameters and the algorithms themselves. Hence, it is challenging to select the most suitable model among all combinations. However, there is always a risk that machine-learning algorithms are insufficient to model a problem.

For easier and guided application of the PQR approach in a product setting, it is recommended to have a workshop self-service kit for the product workshop team. The Volkswagen AG uses a four-step design thinking based approach to identify service kit for the product workshop team. The Volkswagen AG cloud [24], which uses AI for anomaly detection, is a good example of this approach.

B. How do different application domains use evAlA in their daily work?

1) EvAlA applications in different domains

An observation of the teams during the application of the evAlA approach was conducted in a wide range of business areas of the enterprise to get generalization insights. To validate the relevance of evAlA to projects of the Volkswagen AG, we confronted evAlA users with the following questions:

A. What insights does the questionnaire-based evAlA approach deliver to the product teams?

B. How do different application domains use evAlA in their daily work?
C. What is missing to achieve a more effective QA?

We present insights from ten projects/product teams of three legal entities of the Volkswagen Group: The Group Research and engineering entity focus on embedded vehicle AI systems, while Volkswagen Group IT and Audi brand IT emphasizes on business digitalization as AI application domains. The projects have a wide range from focus on autonomous driving assistance systems, after sales use cases to IT internal technical use cases. Objectives of the ML models of our evaluation have a wide range from object recognition on pictures to text analysis in streams. The respective results of the evAla approach are:

A. All teams argued that they did not systematically address all of the evAla aspects. Especially teams with few AI senior experts needed assistance. This assistance gap is closed with evAla for QA aspects. The self-service kit get a high acceptance rate by the teams because they are independent in doing their work by their responsibility without external “supervisor” like from a QA department. The evAla self-service kit fits with the agile mindset about autonomy and mastery.

B. The company’s research teams do not deliver production-ready systems or services and can skip some of the “serving” aspects. However, they have to assure that later on their service design is extensible to fit all serving aspects. Based on the different outcomes the questionnaire is seen more as an inspiration to not forget something relevant or the questionnaire is seen as a part of the product or service documentation to confirm that the released version is safeguarded adequately aligned with state-of-the-art approaches.

C. Feedback about the evAla questionnaire has led to structural improvements, more precise questions with examples to avoid misunderstanding. This has rendered evAla useable without trained moderators to support the self-service mindset of autonomous teams. Furthermore evAla helps to close the gap between the established generic and software code driven QA approaches and the ML specific data driven QA aspects. However, an open point which evAla cannot address is the inherent lack of transparency of how ML algorithms have learned what they have learned. This is still an open research aspect which is important for some businesses cases which demand to demonstrate in a transparent way the decision finding of the AI bases system.

This project validation and feedback loop checks the feasibility of the application of the evAla approach and prepares the rollout of evAla for 2020. Feedbacks and lessons learned of the evaluation leads to some small enhancements and the setup of a self-service kit (a check-list and a how-to). The self-service kit is to ensure a scaling application without experienced moderators for evAla. The rollout quickly establishes the base for a broad empirical analysis. The important actualization of evAla by periodic investigation and subsequent integration of the rapidly progressing state-of-the-art will be assured by a dedicated working group. The working group has to reflect the progress in research approaches and methods for safeguarding ML products and services with the objective to transfer and integrate them into the applicable state-of-the-art in enterprise ML development and service delivery. Currently the frequency for the periodical update is annual.

The added value of the establishment of the evAla method is manifold:
- Teams using ML get guidance for their specific product or service safeguarding (developer view);
- Products and services are transparent safeguarded and the QA is documented (governance view);
- The organization establishes a practice to ensure common safeguarding understanding for ML products and services (QM view);
- The approach to enhancing evAla is open and transparent to ensure currency in the fast developing ML domain (management/organization development view).

V. CONCLUSION AND OUTLOOK

The presented evAla evaluation activities have proven that the approach provides added value in practice. The systematic questionnaire reveals aspects that need systematic tracking and mitigation. EvAla inspires actions and measures to improve AI and connectionist ML models and their training and serving environments. In particular, evAla leads to transparency about the current state of QA. This leads to active decisions about how much additional qualification of the service is useful. EvAla got fast acceptance for example in a centralized AI competence center which introduced evAla as a standard for their project QA. As the ISO 25010 mapping indicates, evAla mostly contributes to safeguarding on the reliability and functional suitability characteristic.

The presented approach is neither a generic assessment model nor a QA standard for AI based products and service. EvAla is rather an instrument helping to pave the way to a systematic QA for AI based products and services. EvAla extends the established QA approaches with AI domain specific aspects. With the self-service offer, the integration into the autonomous agile teams is possible as well as in other development approaches like V-model. Furthermore, there is an option to use the evAla method by central governance or QA instances to compare different business areas or specific service domains about their established ML safeguarding in the future for organizational wide improvements to reach some baselines in ML QA if there is a demand like for agile transitions it came [26].

The future research and development of evAla includes extending the questionnaire to better address more non-connectionist ML approaches. Furthermore, an investigation about typical patterns on ML safeguarding shall be derived by collecting results of a wider range of product evaluations based on the structured questionnaires of evAla.

REFERENCES


