This preprint is a preliminary version. The final version will be published at SQA4AI 2026.

An LLM-based Approach for Automatic ML Prototype Review

1st Selin Coban
Research Group Software Construction
RWTH Aachen University
Aachen, Germany
coban@swc.rwth-aachen.de

2nd Miguel Perez
Research Group Software Construction
RWTH Aachen University
Aachen, Germany
miguel.perez@rwth-aachen.de

3rd Çağatay Akpınar
Research Group Software Construction
RWTH Aachen University
Aachen, Germany
cagatay.akpinar@rwth-aachen.de

4th Baran Tanriverdi
Research Group Software Construction
RWTH Aachen University
Aachen, Germany
baran.tanriverdi@rwth-aachen.de

5th Horst Lichter
Research Group Software Construction
RWTH Aachen University
Aachen, Germany
lichter@swc.rwth-aachen.de

***Abstract*—When developing machine learning (ML) solutions, it is crucial to build prototypes that demonstrate the solution's technical feasibility and potential value. These ML prototypes are typically Jupyter notebooks. However, manually reviewing ML prototypes is time-consuming and can lead to relevant qualities being overlooked from diverse stakeholders' perspectives.**

This paper introduces an innovative approach that uses LLMs to automate the ML prototype review process, thereby improving quality and stakeholder awareness. Through a systematic literature review, we identified key quality characteristics and information needs. The result is an ML prototype review catalog containing a quality model, a list of information needs, and stakeholder personas.

We present PROTO-CHECK, a JupyterLab extension that implements our LLM-based review process. Evaluation results demonstrate high usefulness and usability, as well as heightened developer awareness of stakeholder qualities and needs.

***Index Terms*—Machine Learning, Jupyter Notebook, Software Quality, LLM**

I. INTRODUCTION

In the development of Machine Learning (ML) solutions, product-related development activities are often delayed until a prototype is built, demonstrating both the technical feasibility and the solution's potential value. These prototypes, which we refer to as *ML prototypes*, are typically developed rapidly through experiments with different ML models and are often implemented in Jupyter notebooks, enabling interactive experimentation and rapid iteration.

ML prototypes are developed in a highly interdisciplinary context, where go/no-go decisions on further product development involve input from both technical and non-technical stakeholders, including business and domain experts. Considering these diverse perspectives is crucial, as the quality of an ML prototype directly affects decision-making: poorly structured or undocumented prototypes may embed hidden assumptions, hinder reproducibility, and limit reusability [1]–[4].

When developing software, we have been using tools for many years to review and evaluate code and identify technical weaknesses (technical debt) early. However, additional qualities must be considered for ML prototypes, including transparency and the ability to communicate results effectively to various stakeholders.

Manual reviews of ML prototypes, however, can be time-consuming and may impede adoption among developers. Further, there is often a lack of awareness for relevant qualities and stakeholder concerns to consider [5]. To conduct such reviews in a targeted and efficient manner, they need to be automated. In this paper, we present a novel LLM-based approach to *automate the review of ML prototypes*, as LLMs are also employed in automatic code reviews.

II. RESEARCH GOALS AND DESIGN

To develop an LLM-based approach for ML prototype review, the following research questions must be answered:

- **RQ1:** What quality characteristics of and information about an ML prototype are relevant from the stakeholders' perspectives?
- **RQ2:** How can identified quality characteristics and stakeholders' information needs be used operationally to review ML prototypes?
- **RQ3:** How can reviews of ML prototypes be carried out automatically and effectively using LLMs?

To answer these questions, we organized our research and this paper according to the Design Science Research method [6], defining the following stages:

1) **Problem Identification & Motivation:** see Section I.
2) **Design & Development:** We identified relevant quality characteristics and stakeholders' information needs, applying an SLR and a code-based analysis. We compiled the results into an *ML prototype review catalog* (Section IV) and developed a *LLM-based review process* (Section V).

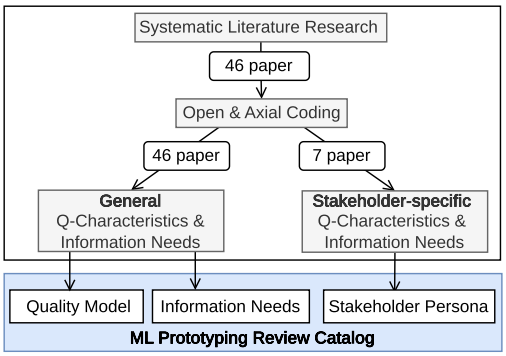

Fig. 1: Steps of the SLR and coding-based analysis and their relation to the review catalog

3) **Demonstration:** We developed the JupyterLab extension PROTO-CHECK implementing this review process (Section VI).
4) **Evaluation:** We conducted a case study to evaluate the usefulness and usability of PROTO-CHECK (Section VII).
5) **Communication:** This paper presents our research process, design decisions, and results. All associated artifacts, including the implementation, evaluation artifacts and a demonstration video, are made publicly available [7].

III. QUALITIES OF AND INFORMATION NEEDS

To answer research question RQ1, which asks for the relevant quality characteristics of ML prototypes and the stakeholders' information needs, we conducted a systematic literature review (SLR). Detailed information on the SLR, including the queries, databases, and exclusion criteria used, is available on Zenodo [7]. Finally, the SLR led to 46 papers, which we analyzed. By applying open and subsequently axial coding [8] on these papers, relevant concepts related to research question RQ1 were identified, extracted, and grouped into three categories:

1) General quality characteristics of ML prototypes.
2) General information about the development of the ML prototype that should be documented or made available.
3) Stakeholder-specific views on quality characteristics or information, which are shaped by their roles and responsibilities.

Our approach to SLR and coding-based analysis is shown in the upper part of Figure 1. The findings achieved served as the basis for developing usable artifacts to support the review of ML prototypes. As a result, we answer research question RQ2 by proposing an ML prototype review catalog consisting of three elements:

1) An ML prototype quality model organizing the identified quality characteristics (MLP-QM).
2) A list of information needs, i.e., information required about an ML prototype.
3) Stakeholder personas. Each stakeholder persona captures stakeholder-specific quality characteristics and information needs.

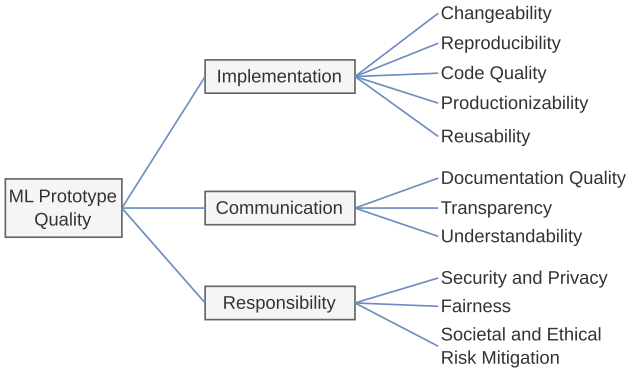

Fig. 2: Structure of the MLP-QM

The lower part of Figure 1 shows how the results of the coding-based analysis were incorporated into the elements of the review catalog.

IV. REVIEW CATALOG

In the following, we present the elements of the review catalog.

A. ML Prototype Quality Model

According to ISO/IEC 25000:2014 [9], a quality model defines a set of quality characteristics and their relationships, providing a framework for specifying quality requirements and evaluating quality. Like most quality models in software engineering, the MLP-QM is structured hierarchically (see Figure 2). The ISO/IEC catalogs 25010:2023 [10] and 25059:2023 [11] were used to select and define terms for identified quality characteristics. Deviations or distinctions from established terms are discussed in the following.

MLP-QM groups the 11 identified quality characteristics into the quality aspects *Implementation*, *Communication*, and *Responsibility* (all publications that mention the characteristic as relevant are listed in parentheses).

Implementation: Groups quality characteristics that affect the effectiveness and efficiency of implementation activities. *Changeability* [12]–[16] refers to how easily the ML prototype can be modified or extended, for example, to accommodate new features or experiments. We decided against using the term "modifiability" (ISO 25010) because it focuses on whether changes introduce new defects or degrade product quality. *Reproducibility* [2], [3], [17], [18] captures the degree to which results can be reliably re-executed, ensuring consistent outcomes under the same conditions. *Code Quality* [12]–[14], [17], [19] concerns the structure and clarity of the implementation, including practices such as modularization and coding guidelines. *Productionizability* [17] concerns how easily an ML prototype can be integrated into and deployed in a production environment, including dependency management and model persistence. We decided against using the terms "adaptability" or "installability" (ISO 25010) because they do not capture the effort required to transform or rebuild the ML prototype into a state suitable for integration into a production environment. *Reusability* [2], [14] denotes how

easily notebook artifacts, particularly code, can be applied in new contexts, which is influenced by the abstraction level and the use of clear name spaces.

Communication: Focuses on how effectively the ML prototype conveys its design, purpose, and functioning to different stakeholders. *Documentation Quality* [2], [4], [20] assesses the completeness and writing style of accompanying explanations. *Transparency* [4], [20]–[28] denotes how clearly the assumptions, design decisions and limitations are explained within the ML prototype. *Understandability* [12], [13], [16], [22], [23], [26], [29]–[32] reflects whether the ML prototype can be comprehended by diverse stakeholders such as data scientists, software engineers, domain experts, or project managers.

Responsibility: Addresses responsible and trustworthy development practices. *Security and Privacy* [16], [26], [27], [32], [33] evaluates the protection of sensitive information, including data handling and access control. *Fairness* [22], [24], [26], [32], [34] concerns potential biases in the data or model behavior that may disadvantage specific groups. *Legal and Ethical Compliance* [15], [16], [20], [23], [25]–[27], [34]–[37] refers to the alignment with relevant legal frameworks, organizational policies, and ethical guidelines to ensure that the solution can be deployed responsibly.

As with other quality models, the quality characteristics of the MLP-QM are not independent. For example, changeability and code quality influence each other, but are interrelated. However, each quality characteristic contributes to evaluating ML prototypes from different perspectives.

B. ML Prototype Information Needs

Table I lists essential information needs of stakeholders involved in prototyping. Respective information should be available so that ML prototypes can be understood, evaluated, and adapted. We have identified four categories of information needs: *Operation*, *Data*, *Model*, and *Limitations*.

TABLE I: Categories of Stakeholders' Information Needs

Category	Information on	Sources
Operation	Installation Reqs. Intended Use	[15], [20], [25], [27], [28], [38]
Data	Preprocessing Steps Dataset	[25]–[27], [33], [35], [38]–[40]
Model	Model Type Evaluation Method Evaluation Summary	[25], [28], [33], [35], [38]
Limit.	Out-of-Scope Use Drawbacks	[24], [27], [38]

Operation: Contains information necessary to operate an ML prototype. *Installation Requirements* specify, for example, the software environment, libraries, and dependencies. *Intended Use* explains the target application domain and expected inputs and outputs.

Data: Contains information on essential aspects of the data used. *Dataset* refers to the origin, composition, and characteristics of the data used to develop the ML model, including size, structure, and any relevant metadata. *Data Processing Steps* describe the transformations applied to the data before model training, such as cleaning, filtering, normalization, feature extraction, or augmentation. Providing clear, structured information about data ensures reproducibility and supports critical assessment of the ML model's behavior.

Model: Contains information on the developed ML model. *Model Type* refers to the underlying algorithm or architecture, including hyperparameters or relevant configurations. *Evaluation Method* details how the ML model performance is evaluated, including metrics, evaluation approach, and datasets used. *Evaluation Summary* presents the key results of this evaluation, enabling others to understand and interpret the ML model's performance.

Limitations: Contains information on the boundaries of the ML solution. *Out-of-Scope Use Cases* are scenarios in which the model should not be applied or where its reliability cannot be guaranteed. *Drawbacks* capture known weaknesses, such as performance issues or data constraints. Making limitations explicit supports responsible use, helps prevent misuse, and guides future development.

C. ML Prototype Stakeholder Persona

Our SLR analysis clearly revealed that not all quality characteristics or information needs are relevant to every stakeholder. Furthermore, specific quality characteristics, such as explainability, are interpreted differently across stakeholders [30]. Therefore, they are neither generalizable nor included in the MLP-QM. To address stakeholder-specific views on quality characteristics or information needs, we introduce the concept of *stakeholder personas*. The concept of a persona is not new and has been applied in a wide range of contexts. Shen et al. explored the use of *persona cards*, concise descriptions of stakeholder-specific interests and needs during ML system development [41]. Their findings align with those of Liu et al., who found that engaging with multiple AI-simulated expert personas increased the perceived creativity of research ideas and promoted critical thinking, without increasing perceived cognitive load [42].

One possible approach to involving stakeholder personas in the review of ML prototypes is to assign stakeholder roles to the corresponding quality characteristics and information needs based on the SLR findings. However, this approach poses two significant challenges. First, role definitions vary considerably across studies, and team composition in practice often differs from the roles synthesized in research. Second, the available literature on stakeholder-specific quality characteristics and information needs remains limited and incomplete.

As we aim for an LLM-based review approach, we use an LLM to derive a stakeholder persona from a *stakeholder profile*, which includes the *role name* and a brief description of the associated *tasks* and *needs*. The role name enables the LLM to estimate technical expertise; the tasks help understand how stakeholders interact with the ML prototype; and the needs section specifies further requirements.

V. The LLM-based Review Process

Figure 4 depicts the LLM-based review process, including two tools: the reviewer and the LLM. The *reviewer* takes a

Prompt for review of implementation qualities:

You are an expert in evaluating the *implementation* of Jupyter Noteboooks.
You will be given a Jupyter Notebook which has been pre-processed and depicts a Machine Learning prototype. [...] You need to evaluate the implementation quality of the notebook based on the following criteria:
1. *Changeability*: Can the notebook be easily modified for different datasets or analyses?
2. *Reproducibility*: Can the results be easily reproduced by others?
3. *Code Quality*: Does the notebook follow best practices?
4. *Productionizability*: Can the notebook be deployed into production environments with minimal effort?
5. *Reusability*: Can the notebook and its code be easily reused?
In each of the explanations, state what was good and what was bad about the notebook regarding the specific criterion. Do not offer to fix the issues by yourself. Just point them out.

Prompt for review of information needs regarding the model:

You are an expert in evaluating Jupyter Notebooks regarding documentation of the given information needs. You will be given a Jupyter Notebook which depicts a Machine Learning Prototype. You need to evaluate if the given information needs are documented in the notebook. Documentation can be in code comments or markdown cells.
1. *Model Type:* Is the type of model used clearly specified?
2. *Evaluation Method:* Is the evaluation method for the model clearly described?
3. *Evaluation Summary*: Are the results of the model clearly explained?
In each of the explanations, state what was good about the description, if it exists, or what documentation was missing.

Prompt template for review of stakeholder persona:

You are a {role} that evaluates a Jupyter Notebook. You will be given a Jupyter Notebook which has been pre-processed and depicts a Machine Learning prototype. [...]
You need to evaluate the notebook from your perspective, according to your role as a {role}. You are not allowed to deviate from your role.
In each of the explanations, state what was good and what was bad about the notebook according to your role. Focus only on aspects that are relevant for your particular role.
Your tasks in the project are: {task}. Your additional needs are: {needs}.

Fig. 3: Prompt excerpts for the elements of the review catalog

notebook and stakeholder profiles as input. Each stakeholder profile is incorporated into a predefined *stakeholder persona prompt*. For each information needs category, a respective *information need prompt* is predefined, and for each MLP-QM quality aspect, a corresponding *quality model prompt* is predefined to guide the review. Before querying the LLM, the reviewer preprocesses the notebook to minimize token usage. The LLM then receives the set of prompts and the preprocessed notebook and produces a *review report* that summarizes the review results across all targeted aspects. Finally, the reviewer presents the report through an interactive, user-friendly dashboard.

The quality of the prompts entered into LLMs is crucial. Therefore, we iteratively developed prompts for the three elements of the reviews catalog in experiments, with a particular focus on improving the reproducibility and consistency of the review report. Excerpts from the prompts are provided in Figure 3. For stakeholder personas, we developed a template prompt; the placeholders will be replaced with the information from a stakeholder profile.

Each prompt further specifies the expected structure and format of the LLM's output (details omitted in the shown excerpts). For each review aspect, the LLM must provide a rating on the commonly used Likert scale $\{--,-,o,+,++\}$ and an explanation, along with references to specific notebook cells, if applicable. The meaning of each rating is explained in Table II.

VI. Proto-Check

To demonstrate the application of the LLM-based review approach on ML prototypes, we developed the JupyterLab extension Proto-Check with access to OpenAI GPT-4.1-mini. During notebook development, users can trigger a review via a button or add stakeholder profiles, which can be saved and reused in other notebooks, in a dedicated input dialog.

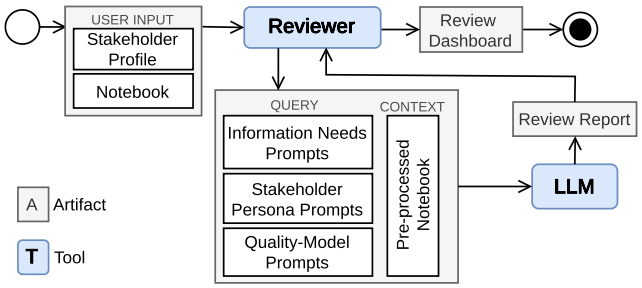

Fig. 4: LLM-based review process

The LLM-generated review report is presented in a review dashboard.

This dashboard, depicted in Figure 5, is structured as follows. The first row displays an evaluation summary. It includes a total score (the aggregation of the individual ratings), the number of positive, neutral, and negative ratings, and a legend. This gives users a general impression of the notebook's quality.

Below, the review results are listed in the following order: personas, quality model, and information needs. For each item, a dashboard card displays the corresponding review results. The card includes the overall score for the reviewed item, enabling users to quickly identify areas for improvement. After opening the card, the review results are displayed in groups by strengths and weaknesses.

VII. Evaluation

We conducted user experiments to evaluate the LLM-based review process we developed and the Proto-Check tool. The evaluation was designed to answer the following questions:

- **EQ1** How do developers perceive the *usefulness* of Proto-Check?
- **EQ2** How do developers perceive the *trustworthiness* of the content produced by the LLM?
- **EQ3** How do developers rate the *usability* of Proto-Check?

In these user experiments, participants used Proto-Check in a controlled setting to perform a predefined task.

A. Participants

11 participants from our professional and personal networks took part in the experiments, all of whom had experience with Python programming in Jupyter notebooks. The participants included individuals with varying levels of expertise, ranging from students to experienced practitioners. All participants had no prior exposure to Proto-Check. Table III summarizes key demographic information.

B. Task and Procedure

The experiment task was designed to ensure participants could use Proto-Check as intended and to examine whether its use raises awareness of the review catalog's elements. The task was divided into five steps: (1) First, participants were given a sample notebook on sleep quality prediction and asked to give an initial impression of its quality. (2) Then, the participants were asked to create an arbitrary stakeholder

TABLE II: Meanings of the ratings

Rating	Quality	Information	Stakeholder Persona
−−	is not met or actively violated	is absent or misleading	is unsatisfied
−	is weak in practice	is mentioned, but unclear or insufficient	is poorly satisfied
o	is unevenly or inconsistently met	is partially implied or scattered	is neutral
+	is mostly met in practice	is stated but has minor gaps	is partially satisfied
++	is met strongly and consistently	is explicit, easy to find, unambiguous, and sufficiently detailed	is fully satisfied

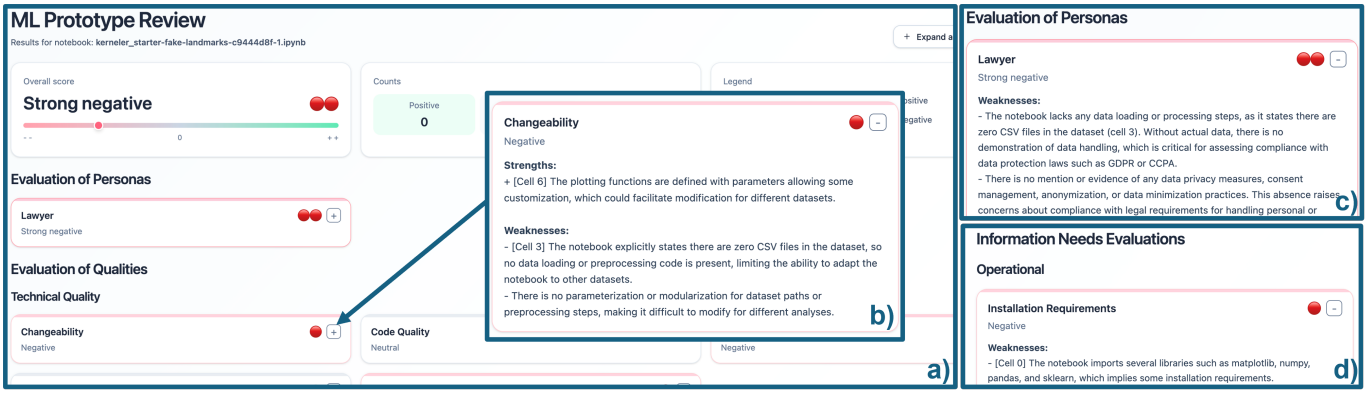

Fig. 5: The PROTO-CHECK Review Dashboard (a). Results for each review item are displayed as expandable cards (b) that list strengths and weaknesses in the corresponding area, e.g. stakeholder persona (c) and information needs (d).

TABLE III: Participant demographics (self-assessed). All have a background in Computer Science or a related field.

Px	Jupyter Notebook Exp.	ML Exp.	Role	Orga. Size
P1	Intermediate	Intermediate	CTO	< 10
P2	Intermediate	Intermediate	Student Researcher	< 10
P3	Beginner	Novice	Student Researcher	< 10
P4	Beginner	Beginner	Student	N/A
P5	Intermediate	Beginner	Student	N/A
P6	Intermediate	Beginner	Test Engineer	> 200
P7	Intermediate	Intermediate	Software Engineer	< 10
P8	Intermediate	Advanced	AI Engineer	> 200
P9	Beginner	Intermediate	CTO	< 10
P10	Intermediate	Intermediate	Data Scientist	$51 - 200$
P11	Beginner	Intermediate	Security Analyst	$51 - 200$

profile. (3) They then used PROTO-CHECK to create the review dashboard and had time to explore its contents. (4) Next, each participant selected two issues identified in the review and improved the corresponding sections of the notebook. (5) After making these changes, they regenerated the review dashboard and examined how the reported issues had changed. (6) Finally, participants were asked to reflect on their initial personal evaluation of the notebook's quality.

The experiments followed a three-phase procedure:

1) *Introduction:* Participants completed a consent and data-usage form and provided demographic information.
2) *Task:* Participants performed the assigned task while thinking aloud, enabling us to capture their reasoning processes and their interactions with PROTO-CHECK.
3) *Post-task questionnaire:* Participants assessed the usefulness and usability of the tool, as well as the trustworthiness of the LLM-generated content, by indicating their agreement with 27 statements on a Likert scale. Example statements are shown in Table IV.

The experimental sessions lasted on average 36 minutes, ranging from 25 to 45 minutes.

TABLE IV: Examples of statements to be evaluated using a Likert scale expressing the level of agreement.

Usefulness (11 stmts.)	
S3	I would use personas.
S6	The differentiation of strengths and weaknesses was useful.
LLM-Trustworthiness (6 stmts.)	
S1	The feedback provided by the LLM was easy to understand.
S2	The feedback provided by the LLM was accurate.
Usability (10 stmts., based on System Usability Scale [43])	
S2	I found the system unnecessarily complex.
S3	I thought the system was easy to use.

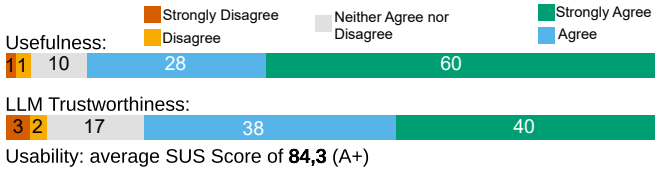

Fig. 6: Questionnaire results (values in %)

C. Data Collection & Analysis

For the *qualitative evaluation* of the experiments, audio from all sessions was recorded and transcribed. In addition, the written feedback provided on the post-task questionnaires was taken into account. To analyze these data, we conducted a thematic analysis following an open-coding approach: we highlighted relevant statements, assigned inductive codes reflecting the topic they addressed (e.g., specific PROTO-CHECK features), and then grouped and summarized all statements associated with each code to identify recurring themes.

We *quantitatively evaluated*, using descriptive statistics, the 27 statements included in the post-task questionnaire and the initial and final impressions of the notebook's quality, marked by the participants on a paper dashboard card.

D. Results

Figure 6 depicts a summary of the responses to the 27 statements on the Likert scale. We counted the frequency of responses (e.g., how many times participants answered with *totally agree*) and converted them into rounded percentages. The *usefulness* and *LLM trustworthiness* statements were constructed in a positive manner; thus, the level of agreement indicates a high level of usefulness or LLM trustworthiness. To get a high-level indication of usability, we compute the average SUS score (scale of 0-100 with a grading system, where A corresponds to excellent). These results indicate a high level of usefulness, LLM-trustworthiness, and usability.

We will discuss the results in more detail below, in combination with insights from the qualitative data, to answer the evaluation questions. Similar to PROTO-CHECK, we present identified strengths and weaknesses.

Usefulness (EQ1). The qualitative data indicate that participants considered the tool highly useful. *Strengths:* Participants particularly valued the inclusion of explicit cell references in the review explanations. P11, who created a "sleep expert" persona, noted that personas are highly practical because they reduce the need for repeated contact with the person represented by the persona. P11 also stated that they would reuse personas across projects. *Weaknesses:* P5 and P9 stated that the review criteria should be weighted, as they would not assign the same weight to each review item card.

LLM Trustworthiness (EQ2). The quantitative data suggest that participants generally trust the LLM. *Strengths:* 10 out of 11 participants verbally praised the plausibility and comprehensibility of the LLM-generated review results. Over multiple reviews, the review report remained largely consistent. *Weaknesses:* Regarding the questionnaire statements on the accuracy and specificity of the LLM's answers, three participants (strongly) disagreed, expressing concerns that the feedback can sometimes be inaccurate or overly generic. Four participants identified minor inconsistencies in the reproducibility of the review reports. Specifically, some ratings changed even though the corresponding issues had not been modified. Two additional participants noted redundancies in the review item cards, and two other participants noted variations in the wording of the explanations. Moreover, two participants expressed uncertainty about evaluating the plausibility of the persona-based review results without consulting the individual represented by the persona. P8 observed that, in their experience, LLMs tend to always identify potential improvements, potentially resulting in an endless feedback loop in which a "perfect notebook" can never be achieved.

Usability (EQ3). The calculated average SUS score corresponds to an excellent usability [44]. *Strengths:* Participants described PROTO-CHECK as clear, visually appealing, and easy to use. *Weaknesses:* Participants suggested several additional UI improvements, such as providing a textual summary of the review results at the top of the dashboard and offering alternative views, for example, a diff view to highlight differences between successive review reports.

Impact on Quality Awareness. Participants were first asked to estimate the quality of the notebook and then to reflect on their assessment after using PROTO-CHECK. Eight participants reported a more critical view of the notebook's quality, adjusting their initial estimates to align with the PROTO-CHECK ratings. They noted that they had previously overlooked specific criteria but acknowledged that these criteria do affect the notebook's overall quality. This effect could also stem from anchoring bias: participants adopt the criteria provided by PROTO-CHECK as an anchor and judge the notebook's quality relative to it. Three participants initially gave the same score as PROTO-CHECK.

VIII. DISCUSSION

Through user experiments, we obtained initial insights into the tool's usefulness and usability, as well as into developers' trust in LLM-generated review results. Furthermore, we identified the strengths and weaknesses of the approach and its realization, PROTO-CHECK. With our goal to raise awareness of relevant qualities, information needs, and stakeholder concerns to consider during prototyping, the evaluation results indicate that our approach significantly improves this aspect.

Given the possibility of endless feedback loops, PROTO-CHECK is best used alongside the development process to highlight potential improvement ideas, leaving it to the developer to decide which suggestions to implement. Furthermore, it has been shown to effectively teach users, such as students, which aspects to consider when developing notebooks.

IX. THREATS TO VALIDITY

Internal Validity: The researcher's presence during the experiments may introduce social desirability effects, causing participants to provide more positive feedback or adapt their behavior to perceived expectations. Additionally, participants recruited from the researchers' networks may feel implicitly encouraged to respond favorably. Furthermore, the usefulness of personas could not be fully evaluated due to the lack of participants with domain expertise.

External Validity: Since participants were drawn from personal and professional networks, the sample consists of people with similar educational backgrounds. This may not reflect the broader target population's diversity. The sample size (n=11) further reduces the generalizability of the findings. Finally, the short-term nature of the study does not capture long-term usage patterns and effects.

X. RELATED WORK

LLMs simulating Experts: A particularly promising application is in requirements engineering, where agents simulate different stakeholders to uncover latent needs [45]–[47]. Recent studies highlight the effectiveness of LLMs in simulating expert perspectives for research ideation, often using personas to represent intersecting identities and broaden the scope of ideation [42], [48]–[50]. However, in both fields, personas frequently lack sufficient diversity [51].

LLM-based Reviews: Research on automated code review has quickly shifted from simple defect detection to automatic correction. To enable LLMs to understand code differences, produce natural language feedback, and independently resolve reviewer comments, significant research makes use of fine-tuning and extensive pre-training [52]–[54]. Recent hybrid techniques use static analysis or symbolic reasoning to link LLM outputs to coding standards and logical constraints, thereby improving the accuracy of the results [55], [56].

Reviewing Systems for ML Prototypes: Static analysis methods are used for automatic identification of code smells and anti-patterns [57], [58], as well as compliance with specific best practices in Jupyter Notebooks [17], [59]. However, rule-based strategies struggle to capture the semantic details of ML workflows [13]. Recently, the use of LLMs for automated ML code review has received more attention [60], [61]. However, these techniques primarily focus on fixing bugs and general code issues. They do not address the broader aspects that affect collaboration and communication among stakeholders.

XI. Conclusion and Future Work

In this paper, we present an LLM-based approach for automatically reviewing ML prototypes. To enable this, we developed a quality model for ML prototypes and derived a set of information needs based on a literature analysis. We further incorporated the concept of stakeholder personas into the review process. Our approach is implemented as the JupyterLab extension Proto-Check. User experiments yielded favorable results regarding both usefulness and usability. Minor issues were identified, such as variation in wording across newly generated review reports. The results also indicate that developers perceive the stakeholder personas as valuable. However, to assess whether the persona-based review feedback is realistic, further studies involving domain experts are required.

References

[1] J. Almahmoud, R. DeLine, and S. M. Drucker, "How Teams Communicate About the Quality of ML Models: A Case Study at an International Technology Company," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, no. GROUP, Jul. 2021.

[2] R. L. Huang, S. Ravi, M. He, B. Tian, S. Lerner, and M. Coblenz, "How Scientists Use Jupyter Notebooks: Goals, Quality Attributes, and Opportunities," in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 2025, pp. 1243–1255.

[3] J. F. Pimentel, L. Murta, V. Braganholo, and J. Freire, "A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks," in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. IEEE, 2019, pp. 507–517.

[4] J. Wang, L. Li, and A. Zeller, "Better Code, Better Sharing: On the Need of Analyzing Jupyter Notebooks," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*, ser. ICSE-NIER '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 53–56.

[5] K. M. Habibullah and J. Horkoff, "Non-functional Requirements for Machine Learning: Understanding Current Use and Challenges in Industry," *Computing Research Repository (CoRR)*, vol. 2109, 2021.

[6] J. Vom Brocke, A. Hevner, and A. Maedche, "Introduction to Design Science Research," in *Design Science Research. Cases*. Springer, 2020, pp. 1–13.

[7] S. Coban and M. Perez, "Artifacts of the Paper "An LLM-Based Approach for Automatic ML Prototype Review" ," Dec. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.17952219

[8] M. Williams and T. Moser, "The Art of Coding and Thematic Exploration in Qualitative Research," *International Management Review*, vol. 15, no. 1, pp. 45–55, 2019.

[9] "Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE," International Organization for Standardization, Switzerland, Standard, Mar. 2023.

[10] "Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Product quality model," International Organization for Standardization, Switzerland, Standard, Nov. 2023.

[11] "Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Quality model for AI systems," International Organization for Standardization, Switzerland, Standard, Jun. 2023.

[12] C. Wong, G. Larsen, R. Huang, B. Sharif, and A. Peruma, "Method Names in Jupyter Notebooks: An Exploratory Study," in *2025 IEEE/ACM 33rd International Conference on Program Comprehension (ICPC)*. IEEE, 2025, pp. 355–366.

[13] B. van Oort, L. Cruz, M. Aniche, and A. van Deursen, "The Prevalence of Code Smells in Machine Learning Projects," in *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*. IEEE, 2021, pp. 1–8.

[14] Y. Jiang, C. Kastner, and S. Zhou, "Elevating Jupyter Notebook Maintenance Tooling by Identifying and Extracting Notebook Structures," in *2022 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2022, pp. 399–403.

[15] N. Tang, M. Li, A. Winecoff, M. Madaio, H. Heidari, and H. Shen, "Navigating Uncertainties: Understanding How GenAI Developers Document Their Models on Open-Source Platforms," *Computing Research Repository (CoRR)*, vol. 2503, 2025.

[16] A. Jabborov, A. Kharlamova, Z. Kholmatova, A. Kruglov, V. Kruglov, and G. Succi, "Taxonomy of Quality Assessment for Intelligent Software Systems: A Systematic Literature Review," *IEEE Access*, vol. 11, pp. 130 491–130 507, 2023.

[17] L. Quaranta, F. Calefato, and F. Lanubile, "Pynblint," in *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, I. Crnkovic, Ed. New York, NY, USA: ACM, 2022, pp. 48–49.

[18] C. Casseau, J.-R. Falleri, X. Blanc, and T. Degueule, "Immediate Feedback for Students to Solve Notebook Reproducibility Problems in the Classroom," in *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2021, pp. 1–5.

[19] M. S. Siddik and C.-P. Bezemer, "Do Code Quality and Style Issues Differ Across (Non-)Machine Learning Notebooks? Yes!" in *2023 IEEE 23rd International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE, 2023, pp. 72–83.

[20] A. Winecoff and M. Bogen, "Improving Governance Outcomes Through AI Documentation: Bridging Theory and Practice," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, ser. CHI '25. New York, NY, USA: Association for Computing Machinery, 2025.

[21] R. K. Mothilal, F. M. Lalani, S. I. Ahmed, S. Guha, and S. Sultana, "Talking About the Assumption in the Room," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, N. Yamashita, V. Evers, K. Yatani, X. Ding, B. Lee, M. Chetty, and P. Toups-Dugas, Eds. New York, NY, USA: ACM, 2025, pp. 1–16.

[22] T. A. Brereton, M. M. Malik, M. Lifson, J. D. Greenwood, K. J. Peterson, and S. M. Overgaard, "The Role of Artificial Intelligence Model Documentation in Translational Science: Scoping Review," *Interactive Journal of Medical Research*, vol. 12, p. e45903, 2023.

[23] A. Crisan, M. Drouhard, J. Vig, and N. Rajani, "Interactive model cards: A human-centered approach to model documentation," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 427–439.

[24] A. Balayn, L. Corti, F. Rancourt, F. Casati, and U. Gadiraju, "Understanding Stakeholders' Perceptions and Needs Across the LLM Supply Chain," *Computing Research Repository (CoRR)*, vol. 2405, 2024.

[25] S. Arnold, D. Yesilbas, R. Gröbner, D. Riedelbauch, M. Horn, and S. Weinzierl, "Documentation Practices of Artificial Intelligence," *Computing Research Repository (CoRR)*, vol. 2406, 2024.

[26] J. Chang and C. Custis, "Understanding Implementation Challenges in Machine Learning Documentation," in *Equity and Access in Algorithms,*

Mechanisms, and Optimization. New York, NY, USA: ACM, 2022, pp. 1–8.

[27] H. Gao, M. Zahedi, C. Treude, S. Rosenstock, and M. Cheong, "Documenting Ethical Considerations in Open Source AI Models," in *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, X. Franch, M. Daneva, S. Martínez-Fernández, and L. Quaranta, Eds. New York, NY, USA: ACM, 2024, pp. 177–188.

[28] T. R. Toma, B. Grewal, and C.-P. Bezemer, "Answering User Questions About Machine Learning Models Through Standardized Model Cards," in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 2025, pp. 1488–1500.

[29] S. Titov, Y. Golubev, and T. Bryksin, "ReSplit: Improving the Structure of Jupyter Notebooks by Re-Splitting Their Cells," in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2022, pp. 492–496.

[30] C. Meske, E. Bunde, J. Schneider, and M. Gersch, "Explainable Artificial Intelligence: Objectives, Stakeholders, and Future Research Opportunities," *Information Systems Management*, vol. 39, no. 1, pp. 53–63, 2022.

[31] V. Bracamonte, S. Pape, S. Löbner, and F. Tronnier, "Effectiveness and Information Quality Perception of an AI Model Card: A Study Among Non-Experts," in *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*. IEEE, 2023, pp. 1–7.

[32] F. Heymans and R. Heyman, "Identifying Stakeholder Motivations in Normative AI Governance: A Systematic Literature Review for Research Guidance," *Data & Policy*, vol. 6, 2024.

[33] C. E. Appel, " Expanding ML-Documentation Standards For Better Security ," in *2025 IEEE 33rd International Requirements Engineering Conference Workshops (REW)*. Los Alamitos, CA, USA: IEEE Computer Society, Sep. 2025, pp. 275–282.

[34] J. Siebert, L. Joeckel, J. Heidrich, A. Trendowicz, K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, and M. Aoyama, "Construction of a Quality Model for Machine Learning Systems," *Software Quality Journal*, vol. 30, no. 2, pp. 307–335, 2022.

[35] A. Bhat, A. Coursey, G. Hu, S. Li, N. Nahar, S. Zhou, C. Kästner, and J. L. Guo, "Aspirations and Practice of ML Model Documentation: Moving the Needle with Nudging and Traceability," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23. New York, NY, USA: Association for Computing Machinery, 2023.

[36] J. Brodny and M. Tutak, "Stakeholder Interactions and Ethical Imperatives in Big Data and AI Development," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 11, no. 1, p. 100491, 2025.

[37] J. L. Nunes, G. D. J. Barbosa, C. S. de Souza, and S. D. J. Barbosa, "Using Model Cards for Ethical Reflection on Machine Learning Models: An Interview-Based Study," *Journal on Interactive Systems*, vol. 15, no. 1, pp. 1–19, 2024.

[38] W. Liang, N. Rajani, X. Yang, E. Ozoani, E. Wu, Y. Chen, D. S. Smith, and J. Zou, "What's documented in AI? Systematic Analysis of 32K AI Model Cards," *Computing Research Repository (CoRR)*, vol. 2402, 2024.

[39] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. III, and K. Crawford, "Datasheets for Datasets," *Commun. ACM*, vol. 64, no. 12, p. 86–92, Nov. 2021.

[40] M. Miceli, T. Yang, L. Naudts, M. Schuessler, D. Serbanescu, and A. Hanna, "Documenting Computer Vision Datasets," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: ACM, 2021, pp. 161–172.

[41] H. Shen, W. H. Deng, A. Chattopadhyay, Z. S. Wu, X. Wang, and H. Zhu, "Value Cards: An Educational Toolkit for Teaching Social Impacts of Machine Learning Through Deliberation," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 850–861.

[42] Y. Liu, P. Sharma, M. Oswal, H. Xia, and Y. Huang, "PersonaFlow: Designing LLM-Simulated Expert Perspectives for Enhanced Research Ideation," in *Proceedings of the 2025 ACM Designing Interactive Systems Conference*, ser. DIS '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 506–534.

[43] J. Brooke *et al.*, "SUS - A Quick and Dirty Usability Scale," *Usability Evaluation in Industry*, vol. 189, no. 194, pp. 4–7, 1996.

[44] J. Lewis and J. Sauro, "Item Benchmarks for the System Usability Scale," *Journal of User Experience*, vol. 13, pp. 158–167, 05 2018.

[45] M. Ataei, H. Cheong, D. Grandi, Y. Wang, N. Morris, and A. Tessier, "Elicitron: A Large Language Model Agent-Based Simulation Framework for Design Requirements Elicitation," *Journal of Computing and Information Science in Engineering*, vol. 25, no. 2, p. 021012, 2025.

[46] M. A. Sami, M. Waseem, Z. Zhang, Z. Rasheed, K. Systä, and P. Abrahamsson, "Early Results of an AI Multiagent System for Requirements Elicitation and Analysis," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2024, pp. 307–316.

[47] Y. Li, J. Keung, Z. Yang, X. Ma, J. Zhang, and S. Liu, "SimAC: Simulating Agile Collaboration to Generate Acceptance Criteria in User Story Elaboration," *Automated Software Engineering*, vol. 31, no. 2, p. 55, 2024.

[48] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong *et al.*, "ChatDev: Communicative Agents for Software Development," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 15 174–15 186.

[49] J. Baek, S. K. Jauhar, S. Cucerzan, and S. J. Hwang, "Researchagent: Iterative Research Idea Generation Over Scientific Literature With Large Language Models," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025, pp. 6709–6738.

[50] H. Su, R. Chen, S. Tang, Z. Yin, X. Zheng, J. Li, B. Qi, Q. Wu, H. Li, W. Ouyang *et al.*, "Many Heads Are Better Than One: Improved Scientific Idea Generation by an LLM-Based Multi-Agent System," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 28 201–28 240.

[51] C. Lazik, C. Kauter, I. Nunes, A. Ziglowski, A. Pryma, C. Katins, L. Grunske, and T. Kosch, "The Good, the Bad, and the Uncanny: Investigating Diversity Aspects of LLM-Generated Personas for Requirements Engineering," in *2025 IEEE 33rd International Requirements Engineering Conference (RE)*, 2025, pp. 244–256.

[52] R. Tufano, S. Masiero, A. Mastropaolo, L. Pascarella, D. Poshyvanyk, and G. Bavota, "Using Pre-Trained Models to Boost Code Review Automation," in *Proceedings of the 44th International Conference on Software Engineering*, 2022, pp. 2291–2302.

[53] J. Lu, L. Yu, X. Li, L. Yang, and C. Zuo, "Llama-Reviewer: Advancing Code Review Automation With Large Language Models Through Parameter-Efficient Fine-Tuning," in *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2023, pp. 647–658.

[54] A. Frömmgen, J. Austin, P. Choy, N. Ghelani, L. Kharatyan, G. Surita, E. Khrapko, P. Lamblin, P.-A. Manzagol, M. Revaj *et al.*, "Resolving Code Review Comments With Machine Learning," in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice*, 2024, pp. 204–215.

[55] B. İçöz and G. Biricik, "Automated Code Review Using Large Language Models with Symbolic Reasoning," in *2025 9th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*. IEEE, 2025, pp. 1–5.

[56] S. Ramesh, J. Bose, H. Singh, A. K. Raghavan, S. R. Chowdhury, G. Sridhara, N. Saini, and R. Britto, "Automated Code Review Using Large Language Models at Ericsson: An Experience Report," *2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 602–607, 2025.

[57] K. Shivashankar and A. Martini, "MLScent: A Tool for Anti-Pattern Detection in ML Projects," in *2025 IEEE/ACM 4th International Conference on AI Engineering–Software Engineering for AI (CAIN)*. IEEE, 2025, pp. 150–160.

[58] P. Hamfelt, R. Britto, L. Rocha, and C. Almendra, "Automatic Identification of Machine Learning-Specific Code Smells," *Computing Research Repository (CoRR)*, vol. 2508, 2025.

[59] P. Subotić, L. Milikić, and M. Stojić, "A Static Analysis Framework for Data Science Notebooks," in *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice*, 2022, pp. 13–22.

[60] B. Jin, J. Wang, and P. Nie, "Suggesting Code Edits in Interactive Machine Learning Notebooks Using Large Language Models," *Computing Research Repository (CoRR)*, vol. 2501, 2025.

[61] H. Elhashemy, Y. Lotfy, and Y. Tang, "Bridging the Prototype-Production Gap: A Multi-Agent System for Notebooks Transformation," *Computing Research Repository (CoRR)*, vol. 2511, 2025.