Towards Trustworthy AI Assistants in Safety-Critical Engineering: A Co-Designed Trust Calibration Checklist

Matteo Pallomo, Eva Ledovskaia, Andrea Barbantani, Alessio Quaranta, Marco Farano, Elena Ruberto, Fiamma Pia Paternò, Michele Barale, Federico Ciociola, Flora D'Angelo.
Politecnico di Torino
Turin, Italy - 2026

1 Abstract

Artificial Intelligence (AI)-assisted systems are increasingly being integrated into engineering workflows to support activities such as documentation analysis, verification, and requirement generation. However, within safety-critical environments, the operational adoption of such systems strongly depends on the ability of practitioners to assess whether the AI assistant can be considered sufficiently trustworthy for the delegated task.

This work presents the development of an operational trust-calibration checklist intended to support both the design-time and validation-time assessment of AI-assisted engineering systems. The proposed framework was iteratively refined through multiple co-design and review phases involving both non-expert participants and industrial practitioners operating in the ADAS domain.

The final checklist organizes trust evaluation into five operational sections addressing task criticality assessment, objective system evidence, human-system transparency, operational robustness, and human oversight and control. Additionally, the framework introduces explicit *Go/No-Go* evaluation logic driven by task criticality in order to support operational adoption decisions.

To demonstrate the applicability of the proposed approach, the checklist was implemented into example engineering requirements using the EARS formalism and applied to a representative ADAS requirement-generation use case involving an AI-assisted engineering tool.

The results suggest that trustworthiness in AI-assisted engineering systems emerges from the interaction between objective evidence, transparency, robustness, and effective human supervision rather than from isolated technical performance metrics alone.

2 Purpose of the Work

The objective of this work is to develop a generic checklist that can be applied both before and after the development process of an AI-based system. More specifically, the proposed framework aims to provide an operational assessment document that may support either the design phase or the validation phase in evaluating the trustworthiness of a given system.

Although the checklist is primarily intended for AI-assisted engineering tools and AI-based agents, a major design objective of this work is generality. Ideally, the resulting framework should remain applicable across different domains and application contexts with minimal or no structural modifications.

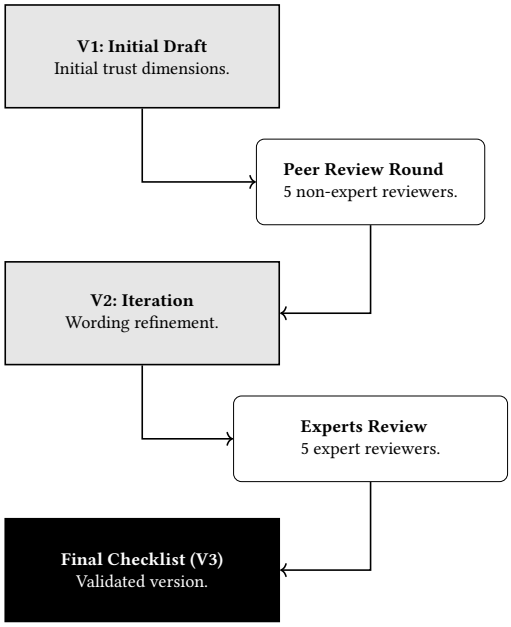

Figure 1: Overview of the iterative refinement workflow.

3 Methodology

The checklist refinement process followed an iterative co-design methodology. Rather than treating the checklist as a static artifact, each version was considered an evolving prototype progressively refined through structured feedback sessions involving both non-expert participants and industrial practitioners.

The co-design process evolved across four main iterations summarized in Figure 1. Each step consisted of two main phases: data collection and data analysis. A detailed explanation of each phase will be provided in the following sections.

4 Data Collection Procedure

The data collection procedure was tailored according to the type of participants from whom feedback was collected. In particular, separate procedures were defined for *non-experts* and *experts*. Both procedures are detailed in the following sections.

Table 1 summarizes the demographics of the participants involved in each review phase.

4.1 Data Collection with Non-Experts

The first iteration of the refinement process involved non-expert participants, namely peers from our academic environment. The

ID	Gender	Role
		V1
NE1	Male	Engineering Student
NE2	Female	Engineering Student
		V2
NE3	Female	Engineering Student
NE4	Male	Engineering Student
NE5	Male	Engineering Student
		V3
E1	Male	Agile Product Owner
E2	Male	Agile Scrum Master
E3	Male	SW Architect - ADAS
E4	Female	SW Engineer - ADAS
E5	Male	System Engineer - ADAS

Table 1: Participant demographics.

objective of these early review sessions was not to validate domain-specific engineering constraints, but rather to identify issues related to wording clarity, ambiguity, readability, and overall questionnaire structure before involving industrial practitioners.

The data collection process was divided into three main phases:

- *Framework Summary (3 mins):* during the initial phase, participants were introduced to the overall trust framework derived from our previous scoping review. This brief overview aimed to establish a common baseline among reviewers, ensuring that each participant could contextualize the rationale behind the checklist and understand the relationship between the proposed questions and the broader trust dimensions previously identified in the literature.
- *Section-by-Section Review (15 mins):* participants were then guided progressively through the checklist, reviewing one section at a time. For each section, reviewers were encouraged to identify wording ambiguities, unclear formulations, structurally inconsistent questions, overlapping concepts, and potentially missing trust-related aspects. During this phase, the checklist was printed on A3 paper sheets, and the collected feedback was initially captured through physical Post-it notes and handwritten annotations attached to the printed versions of the checklist.
- *Final Free Feedback (5 mins):* the final phase of the session was intentionally left less structured in order to encourage broader reflections regarding the checklist and its overall approach. Participants were allowed to provide open-ended comments not necessarily tied to specific sections or questions. This phase aimed to promote the emergence of more spontaneous observations, alternative perspectives, and higher-level considerations regarding the usability and perceived purpose of the framework.

4.2 Data Collection with Experts

The final stage of the data collection process involved interviews with industrial practitioners, particularly engineers working at *Jaguar Land Rover Italia*. Most participants operated within the ADAS domain, although they covered different organizational and technical roles (please refer to Table 1).

Compared to the earlier review sessions, the expert interviews followed a less structured and more discussion-oriented approach. Rather than strictly guiding participants through the checklist, the process was intentionally adapted to allow experts to lead the discussion more freely according to their professional experience and priorities. The feedback was collected via a *Microsoft Teams* online meeting whose minutes are reported in Appendix B.

- *Framework Summary (5 mins):* unlike the non-expert participants, the industrial practitioners were not previously familiar with either the trust framework derived from our previous work or the scope of the current study. Therefore, additional time was dedicated to presenting the overall objective of the work, the intended outcomes, and the main findings of the previous studies.
- *Expert Walkthrough and Live Feedback (25 mins):* during the main portion of the review, the checklist was shared on screen and discussed collaboratively section by section. However, unlike the earlier sessions, experts were encouraged to move more freely across the document, revisit previous sections, ask clarification questions, and focus on aspects they considered particularly relevant from an industrial perspective. The objective of this approach was to maximize the value of the practitioners' expertise and avoid constraining the discussion within an excessively rigid interview structure.
  During this phase, the feedback extended beyond wording or readability considerations and increasingly focused on industrial realism, workflow integration, explainability requirements, accountability concerns, trust calibration, and standards compliance. The review therefore acted both as a validation of the checklist content and as a broader reflection on the practical integration of AI-based assistants within safety-critical engineering workflows.

4.3 Version Summary

The checklist evolved through three major iterations, each introducing progressively more mature assumptions regarding the operational trust assessment of AI-assisted engineering systems.

Version 1 — Initial Draft. Version 1 represented an exploratory attempt to identify relevant trust dimensions for AI-assisted engineering workflows. The initial structure primarily focused on collecting a broad set of potentially relevant evaluation criteria without a sufficiently clear separation between technical robustness, transparency, operational governance, and human oversight aspects.

During the first internal review phase, it became evident that the overall approach suffered from several major limitations:

(1) The usability of the checklist was limited and primarily intended for the design phase. It did not adequately support validation activities.
(2) The checklist structure lacked sufficient conceptual organization and clarity.
(3) The proposed evaluation criteria were overly domain-specific and therefore insufficiently generic for broader engineering applicability.

(4) Several questions and sections exhibited overlap and lacked conceptual orthogonality.

As a consequence, following the first iteration, the checklist was substantially reworked by introducing the concept of operational *sections* and by generalizing multiple trust-related concepts.

Version 2 — Structural Redesign. Version 2 introduced a major redesign of the checklist architecture. The framework was reorganized into multiple operational sections, each targeting a distinct assurance dimension: objective system evidence, human-system transparency, operational robustness and human oversight.

This redesign aimed to:

- Improve conceptual orthogonality between evaluation dimensions.
- Reduce overlap between checklist items.
- Increase applicability across heterogeneous domains.
- Allow the usage of the checklist during both design and validation phases.

Although Version 2 represented a significant improvement over the initial draft, several redundancies and ambiguities were still present, particularly regarding traceability, uncertainty communication, and robustness-related concepts.

Version 3 — Expert-Driven Refinement. Version 3 was developed following feedback collected from industrial domain experts involved in Advanced Driver Assistance Systems (ADAS) development activities.

Based on this review phase, the checklist underwent further rationalization and simplification. Several overlapping items were merged or removed, wording was refined to improve operational clarity, and the distinction between objective evidence, transparency, robustness, and supervisory control was strengthened.

The final version additionally introduced:

- Explicit task criticality assessment through Section 0.
- A ruleset allowing users to derive a binary operational outcome from the checklist (i.e., *Trust can be granted* versus *Trust cannot be granted*).
- Clearer operational interpretations of checklist fulfillment.
- Improved alignment between the checklist structure and real-world engineering validation workflows.

The resulting framework represents the final operational trust calibration checklist adopted in this work.

One additional refinement introduced between Version 2 and Version 3 concerned the transition from a purely *checkbox-based* structure to a predominantly *yes/no-based* questionnaire format. Although this modification did not alter the underlying content or the resulting assessment logic of the checklist, it significantly improved its readability and operational usability. More specifically, the new structure simplified the enumeration of individual questions, allowing each checklist item to be uniquely referenced through a compact identifier format (e.g., *S1-Q3*: Section 1, Question 3). This improvement facilitates traceability, discussion during review activities, and the referencing of specific checklist items within reports or validation documents.

The checkbox-based structure was intentionally retained only for the *Task Criticality Assessment* gate, since that section requires the selection of a single mutually exclusive classification level.

The granular changes made from Version 2 to Version 3 are available in Appendix D.

5 Checklist Usage

The checklist may be used in two distinct scenarios: (i) to support and guide the system design process, or (ii) to validate a completed system design prior to operational deployment.

In both cases, the checklist shall be applied according to the following procedure.

First, the evaluator shall assess the task criticality through *Section 0*. In performing this assessment, the evaluator should consider the following question: *what consequences could arise from erroneously trusting this system?* The identified criticality level determines the strictness of the subsequent trust assessment and the minimum acceptable checklist fulfillment threshold. The criticality levels are intentionally defined in a generic manner in order to avoid binding the framework to a specific industrial sector or application domain.

Indeed, the interpretation of terms such as *Catastrophic* may vary significantly depending on the operational context. For example, within the medical domain, a catastrophic failure may involve the loss of one or more patients, whereas in a financial context it may correspond to the loss of a substantial amount of capital. As a consequence, preliminary activity may be considered before Section 0, namely the contextual definition of what each criticality level represents within the specific operational domain.

After the task criticality has been established, all remaining sections (*Section 1* through *Section 4*) shall be completed. There is no predefined limit on the number of satisfied or unsatisfied checklist items. Instead, the overall trustworthiness assessment shall be determined by comparing the achieved checklist fulfillment level against the minimum threshold associated with the assessed task criticality, as defined in Table 2.

Based on this comparison, operational trust in the system may either be granted, conditionally granted, or denied.

6 Checklist Operationalization

In this section, a subset of example requirements derived from the proposed checklist is presented. In order to avoid excessive redundancy, not every checklist item is mapped to a corresponding requirement example. However, the objective of this section is to demonstrate that the checklist can be translated into actionable design artifacts capable of supporting both system designers during the early development stages and validators during validation and test-case definition activities.

In this context, it is important to emphasize that the generated requirements intentionally use the *AI system* as the subject of the statements. This choice was made primarily for practical and illustrative purposes, namely to provide the reader with a concrete and relatable reference scenario.

A more general formulation of the requirements may be obtained by replacing the subject with the more generic term *system*.

Task Criticality	Minimum Checklist Fulfillment	Operational Interpretation
Catastrophic	100% of applicable items	All applicable checklist items across all sections must be satisfied. No unresolved deficiencies are acceptable prior to operational deployment.
Major	80% of applicable items	Nearly all checklist items must be satisfied. Residual deficiencies must be formally documented, justified, and assessed as non-safety-critical. **S4-Q2 (system override) must be satisfied.**
Moderate	60% of applicable items	Most checklist items should be satisfied. Limited deficiencies may be tolerated provided compensating controls or additional human oversight mechanisms are in place.
Minor	40% of applicable items	Partial compliance is acceptable when outputs remain subject to standard engineering review and operational consequences are limited.
Negligible	20% of applicable items	Checklist execution is primarily advisory. Limited assurance evidence may be acceptable due to minimal operational impact.

Table 2: Suggested minimum checklist fulfillment thresholds as a function of task criticality.

Conversely, more specialized requirement formulations may be derived by further contextualizing the subject according to the target application domain or operational environment.

6.1 Example Requirements Derived from the Checklist

6.1.1 Section 1 — Objective System Evidence.

Checklist Item: *S1-Q1*

WHILE defining the AI system validation strategy,
THEN a representative set of ground-truth scenarios shall be identified and documented.

Checklist Item: *S1-Q8*

WHEN the AI agent has generated engineering artifacts or recommendations,
THEN it shall be possible to trace back the artifacts to any provided input document/information.

6.1.2 Section 2 — Human-System Transparency.

Checklist Item: *S2-Q2*

WHEN the AI system generates recommendations or engineering outputs,
THEN the associated reasoning process shall be presented in a logically coherent and reviewable manner.

Checklist Item: *S2-Q5*

WHEN the AI system generates recommendations or engineering outputs,
THEN the system shall explicitly communicate associated uncertainty and confidence intervals to the user.

6.1.3 Section 3 — Operational Robustness.

Checklist Item: *S3-Q1*

WHEN incomplete, degraded, or ambiguous inputs are provided to the AI system,
THEN operational performance shall remain within predefined acceptance thresholds.

Checklist Item: *S3-Q8*

WHEN the AI system is modified or updated,
THEN regression validation activities shall be executed before operational redeployment.

6.1.4 Section 4 — Human Oversight and Control.

Checklist Item: *S4-Q1*

WHEN the AI agent has generated engineering artifacts or recommendations,
THEN it shall not commit any work before being accepted by a human-reviewer.

Checklist Item:*S4-Q2*

WHILE the AI system is operating,
THEN it shall be possible, at any moment, to override the system stopping its execution.

7 Applied Use Case - ADAS System Engineer

In this section, the proposed checklist is applied to a representative industrial use case within the automotive domain, specifically in the context of ADAS (Advanced Driver Assistance Systems).

More specifically, we considered the following task: an AI-augmenting tool that enables the automatic generation of system requirements from heterogeneous engineering inputs and project deliverables.

The objective of the engineer is to evaluate the trustworthiness of the proposed AI system in order to determine the appropriate operational trust level associated with its adoption. In particular, the assessment aims to determine whether:

- the system can be considered sufficiently trustworthy for largely autonomous usage;
- the system may be adopted only under continuous human oversight;
- or the system should be rejected due to insufficient trustworthiness guarantees.

The following subsections illustrate how the checklist can be used to support this evaluation process and how the resulting assessment may influence operational adoption decisions.

Section 0 — Task Criticality Assessment. The first step consists in evaluating the criticality of the delegated task. Within the considered scenario, the AI system is responsible for generating system requirements that may eventually influence safety-critical ADAS functionalities such as emergency braking, lane-keeping assistance, or driver monitoring behavior.

Although the AI system does not directly control the vehicle, incorrect or ambiguous requirements may propagate downstream into architectural decisions, implementation defects, or incomplete safety validation activities.

For this reason, the engineer classifies the task criticality as *Major*. The rationale behind this decision is that incorrect outputs may not immediately produce catastrophic consequences; however, they may significantly contribute to engineering defects, regulatory non-compliance, or safety-related integration issues. Furthermore, such errors may also have a substantial business impact, since correcting erroneous requirements during later development stages is typically far more costly than addressing them early in the process.

As a consequence, the checklist requires a high fulfillment level before operational trust may be granted.

Section 1 — Objective System Evidence. The engineer then evaluates the availability of objective evidence supporting the reliability of the AI system.

Particular attention is dedicated to the existence of benchmark validation activities performed against representative engineering datasets and previously validated requirement baselines. The engineer verifies whether the generated outputs have been compared against ground-truth reference artifacts and whether measurable performance indicators are available.

The availability of traceable validation artifacts also plays a major role in the assessment. Within the automotive domain, engineering activities are strongly audit-oriented and require high traceability between inputs, outputs, verification activities, and final decisions.

The engineer additionally evaluates whether the AI system provides version control mechanisms and whether configuration changes can be tracked over time. This aspect is considered particularly important because the behavior of AI systems may evolve following retraining activities, prompt modifications, or software updates.

Finally, the engineer verifies the availability of documentation regarding system assumptions, operational limitations, and uncertainty estimation mechanisms. The absence of clear operational boundaries would significantly reduce the possibility of safely integrating the system into industrial workflows.

Section 2 — Human-System Transparency. The second operational assessment phase focuses on transparency and reviewability.

The engineer evaluates whether the generated requirements are understandable, logically coherent, and aligned with established engineering conventions. In particular, the engineer analyzes whether the AI-generated requirements can be reviewed without excessive additional effort and whether the associated reasoning process remains sufficiently interpretable.

The communication of uncertainty is also considered essential. The engineer expects the AI system to explicitly highlight ambiguous situations, incomplete information, or low-confidence outputs rather than presenting all generated requirements with the same apparent certainty.

Another important aspect concerns behavioral predictability. The engineer repeatedly executes similar requirement-generation tasks in order to evaluate whether the AI system behaves consistently across comparable interactions.

If similar prompts generate highly unstable outputs or significantly different engineering assumptions, operational trust in the system would decrease substantially.

Section 3 — Operational Robustness. The robustness assessment evaluates how the AI system behaves under imperfect operational conditions.

Within realistic engineering environments, input documentation is often incomplete, partially outdated, inconsistent, or ambiguously written. For this reason, the engineer intentionally provides degraded and partially conflicting engineering inputs in order to observe the system behavior.

The engineer verifies whether the AI tool remains capable of generating coherent outputs without producing logically inconsistent or clearly unverifiable requirements. Particular attention is dedicated to anomalous conditions where the system may hallucinate unsupported engineering assumptions.

The engineer additionally evaluates the existence of regression validation activities performed after system updates. This aspect is considered critical because even small changes in AI models or prompting strategies may unintentionally modify output behavior.

Operational monitoring and logging mechanisms are also analyzed. The ability to reconstruct previous interactions and generated outputs is considered fundamental for post-analysis activities, debugging, and accountability purposes.

Section 4 — Human Oversight and Control. The final section evaluates whether sufficient human authority remains present within the operational workflow.

The engineer explicitly rejects the possibility of fully autonomous requirement generation for safety-relevant engineering activities. Instead, the AI system is considered a support tool whose outputs must remain subject to expert review and approval.

The engineer verifies that generated requirements remain clearly distinguishable from manually authored engineering artifacts. This separation is considered important to avoid ambiguity regarding authorship, accountability, and validation responsibilities.

The existence of override and escalation mechanisms is also evaluated. The engineer expects to retain the ability to reject generated outputs, interrupt automated activities, and escalate uncertain situations to senior domain experts whenever necessary.

Finally, the checklist confirms that responsibility for final engineering decisions remains entirely under human authority. The AI

system may accelerate documentation activities and reduce repetitive work, but accountability for safety-critical decisions cannot be delegated to the AI agent itself.

8 Conclusions

This work presented the development of an operational trust calibration checklist intended for AI-assisted systems operating within engineering environments. The proposed framework was specifically designed to support both the design and validation phases of AI-based systems by providing a structured method for evaluating their trustworthiness.

The final checklist evolved through multiple iterative refinement phases involving both internal restructuring activities and feedback collected from industrial practitioners operating in the ADAS domain. Throughout this process, the framework progressively shifted from an exploratory trust-oriented questionnaire toward a more operational and engineering-oriented assessment instrument.

The resulting checklist organizes trust evaluation into five complementary operational sections: task criticality assessment, objective system evidence, human-system transparency, operational robustness, and human oversight and control. This structure aims to reduce conceptual overlap while improving the applicability of the framework within realistic engineering workflows.

The proposed approach additionally introduced explicit *Go/No-Go* evaluation logic driven by task criticality. This aspect allows the checklist to move beyond a purely descriptive artifact and become a practical decision-support instrument capable of guiding operational adoption choices.

The work also demonstrated that checklist items may be translated into actionable engineering requirements through the use of structured requirement specification approaches such as EARS. This operationalization activity showed how abstract trust-related principles may be concretely transformed into verifiable functional and non-functional system requirements.

The presented ADAS use case further highlighted how the checklist may support engineers in evaluating whether an AI-assisted system can be trusted autonomously, adopted under human supervision, or rejected due to insufficient assurance evidence.

9 Limitations and Future Work

Despite the iterative refinement process and the involvement of industrial practitioners, this work still presents several limitations.

First, the participant pool involved in the refinement and review activities remained relatively limited. Although the collected feedback proved useful for improving the operational structure and clarity of the checklist, a broader and more heterogeneous set of reviewers would improve the robustness and generalizability of the resulting framework.

Second, the industrial experts involved in the review process belonged to a relatively homogeneous engineering environment within the automotive and ADAS domain. As a consequence, some assumptions embedded within the checklist may still partially reflect domain-specific engineering practices, validation expectations, and organizational workflows.

Another limitation concerns the qualitative nature of the current trust calibration process. Although the checklist introduces operational *Go/No-Go* thresholds based on task criticality, the evaluation itself still relies primarily on expert judgment and manual assessment activities. Future work may therefore investigate more quantitative scoring methodologies capable of supporting partially automated trust assessment procedures.

Additionally, the operationalization activity presented in this work focused only on a limited subset of example requirements. A larger-scale validation effort involving complete requirement generation workflows and downstream verification activities would provide stronger evidence regarding the practical applicability of the proposed framework.

Future work may therefore focus on:

- validating the checklist across multiple industrial sectors;
- evaluating inter-reviewer consistency during checklist execution;
- investigating quantitative trust calibration metrics;
- analyzing the relationship between task criticality and acceptable automation levels;
- and studying how trust calibration evolves through prolonged interaction with AI-assisted engineering systems.

10 Appendix A - Checklist Versions

10.1 Version 3 — Final

10.1.1 Section 0 — Task Criticality Assessment.
Target: Assess criticality of the delegated task.
Rule: Select one single classification level.

- ☐ **Catastrophic:** Failure may cause severe legal liabilities, critical financial losses, or permanent reputational damage.
- ☐ **Major:** Failure may cause major operational disruptions, regulatory non-compliance, or critical errors in final output.
- ☐ **Moderate:** Failure may introduce localized issues with the workflow, requiring substantial correction effort.
- ☐ **Minor:** Failure has limited operational impact and remains recoverable through standard review activities.
- ☐ **Negligible:** Failure produces minimal operational or downstream consequences.

10.1.2 Section 1 — Objective System Evidence.
Target: Evaluate objective evidence supporting reliability and clear usage boundaries.
Rule: Check Yes or No for all applicable items.

1. Benchmark validation against representative ground truth is available. ☐ Yes ☐ No
2. Output consistency across repeated executions and system versions has been verified. ☐ Yes ☐ No
3. It is clear what types of personal or company data can be safely entered into the system to avoid privacy or copyright violations. ☐ Yes ☐ No
4. Appropriate usage contexts and system operational limitations are clearly documented. ☐ Yes ☐ No
5. User instructions, warnings, and system operational limitations are clearly documented. ☐ Yes ☐ No
6. Confidence estimates or uncertainty indicators are available. ☐ Yes ☐ No
7. Version control and configuration management mechanisms are implemented. ☐ Yes ☐ No
8. Validation and verification documents are traceable and auditable. ☐ Yes ☐ No

10.1.3 Section 2 — Human-System Transparency.
Target: Evaluate interpretability and reviewability of generated outputs.
Rule: Check Yes or No for all applicable items.

1. Generated outputs are understandable without excessive additional analysis. ☐ Yes ☐ No
2. Supporting reasoning or justification can be followed coherently. ☐ Yes ☐ No
3. System behavior remains predictable across comparable interactions. ☐ Yes ☐ No
4. Generated outputs align with user expectations and goals. ☐ Yes ☐ No
5. Uncertainty, ambiguity, and confidence limitations are communicated clearly. ☐ Yes ☐ No
6. Standard users can review and verify the generated outputs within an acceptable timeframe and effort. ☐ Yes ☐ No

10.1.4 Section 3 — Operational Robustness.
Target: Evaluate the system's ability to handle imperfect, realistic, or unexpected conditions.
Rule: Check Yes or No for all applicable items.

1. System performance remains acceptable under incomplete, degraded, or ambiguous inputs. ☐ Yes ☐ No
2. The system avoids generating clearly false, invented answers, or logically contradictory outputs. ☐ Yes ☐ No
3. The system avoids generation of logically inconsistent or unverifiable outputs. ☐ Yes ☐ No
4. When the AI application is updated, users briefly check that their standard, saved prompts still work as expected. ☐ Yes ☐ No
5. Resource utilization and response latency remain within operational constraints. ☐ Yes ☐ No
6. The system detects and flags anomalous or unsupported operating conditions. ☐ Yes ☐ No
7. Operational monitoring and logging mechanisms are available. ☐ Yes ☐ No
8. Operational monitoring and logging mechanisms are available. ☐ Yes ☐ No

10.1.5 Section 4 — Human Oversight and Control.
Target: Evaluate adequacy of human authority and supervisory control.
Rule: Check Yes or No for all applicable items.

1. Human review is required before operational acceptance of generated outputs. ☐ Yes ☐ No
2. Human override mechanisms remain available for safety-relevant decisions. ☐ Yes ☐ No
3. Responsibility boundaries between operators and the AI system are explicitly defined. ☐ Yes ☐ No
4. AI-generated content remains clearly distinguishable from manually created content. ☐ Yes ☐ No
5. Reviewer actions, modifications, and approvals are logged and traceable. ☐ Yes ☐ No
6. Clear procedures exist when the AI provides uncertain or conflicting outputs. ☐ Yes ☐ No
7. Human supervision level scales with task criticality. ☐ Yes ☐ No
8. Users possess enough baseline knowledge to identify and catch potential system errors. ☐ Yes ☐ No

10.2 Version 2

10.2.1 Section 1 — Objective System Metrics.
Target: Evaluate objective evidence supporting reliability, traceability, and compliance.
Rule: Check all applicable items.

- ☐ Availability of benchmark test results against representative ground truth.
- ☐ Availability of evidence demonstrating consistent outputs across repeated executions and system updates.
- ☐ Availability of evidence that training and validation data originate from legitimate and documented sources.
- ☐ Availability of evidence that operational and user data are protected with appropriate privacy and cybersecurity measures.
- ☐ Availability of documentation regarding system architecture, assumptions, and operational principles.
- ☐ Availability of confidence levels or uncertainty estimates associated with generated outputs.
- ☐ Availability of evidence that the system complies with applicable standards, guidelines, and development processes.
- ☐ Availability of evidence that generated outputs comply with applicable standards and regulations.
- ☐ Availability of version control and configuration management mechanisms.
- ☐ Availability of traceable validation and verification artifacts.

10.2.2 Section 2 — Human-System Transparency.
Target: Evaluate interpretability, predictability, and traceability of generated outputs.
Rule: Check all applicable items.

- ☐ System outputs are understandable without excessive additional analysis.
- ☐ The reasoning or justification supporting generated outputs can be followed without logical inconsistencies.
- ☐ System behavior remains predictable across similar interactions.
- ☐ Generated outputs align with operational expectations and engineering intent.
- ☐ Confidence estimates and limitations are communicated clearly.
- ☐ The system provides sufficient traceability between inputs and generated outputs.
- ☐ The system communicates uncertainty, ambiguity, or low-confidence conditions explicitly.
- ☐ Generated artifacts can be reviewed and validated by domain experts within acceptable effort.

10.2.3 Section 3 — Operational Robustness.
Target: Evaluate operational stability and resilience under realistic conditions.
Rule: Check all applicable items.

- ☐ Output repeatability is maintained for equivalent inputs and operational conditions.
- ☐ System performance remains acceptable under incomplete, degraded, or ambiguous inputs.
- ☐ Failure modes and known operational limitations are documented.
- ☐ The system avoids generation of logically inconsistent or unverifiable outputs.
- ☐ Adversarial, malformed, or unexpected inputs have been evaluated.
- ☐ System updates trigger regression validation procedures.
- ☐ Resource utilization and response latency remain within operational constraints.
- ☐ The system detects and flags low-confidence or anomalous outputs.
- ☐ The system fails safely under invalid or unsupported operating conditions.
- ☐ Operational monitoring and logging mechanisms are available for post-operation analysis.

10.2.4 Section 4 — Human Oversight and Control.
Target: Evaluate adequacy of human supervision and authority.
Rule: Check all applicable items.

- ☐ Human override mechanisms are always available for safety-relevant decisions.
- ☐ Human review is required before generated outputs are operationally accepted.
- ☐ Responsibility boundaries between the operator and the AI system are explicitly defined.
- ☐ AI-generated artifacts remain distinguishable from manually authored artifacts.
- ☐ Reviewer modifications and approval actions are logged and traceable.
- ☐ Escalation procedures exist for uncertain, conflicting, or low-confidence outputs.
- ☐ The required level of human supervision scales with task criticality.
- ☐ Operators possess sufficient domain expertise to critically evaluate generated outputs.
- ☐ System usage policies define acceptable operational boundaries and prohibited usage conditions.

10.3 Version 1

10.3.1 Scenario. You are a System Engineer that works on ADAS (Assisted Driving Assistance System). You are using an AI-agent to auto-generate requirements from your system of choice (e.g. Adaptive Cruise Control). You are given the following checklist to evaluate how much you trust the system you are using.

10.3.2 Objective System Metrics.

- ☐ Are test results provided against a baseline/ground truth?
- ☐ Is there evidence of outputs stability over time?
- ☐ Are data used for training coming from legitimate datasets?
- ☐ Is there evidence that data input in the system are safely kept?
- ☐ Is the algorithm behind the agent well documented and available?

10.3.3 *Context.*

- ☐ Is the output obtained consistent and potentially useful with respect to the question asked?
- ☐ Would you use the system for safety and mission critical applications?
- ☐ Would you feel more comfortable using the system with a human in the room?
- ☐ Are you afraid about being replaced by AI?

10.3.4 *Human traits.*

- ☐ Have you ever had first hand (design) experience with AI agents?
- ☐ Have you ever had first hand (usage) experience with AI agents?
- ☐ Are you generally open minded with respect to technology?
- ☐ When you use automatic tools, are you skeptical about their outputs?

10.3.5 *Subjective perception.*

- ☐ Is the reasoning/thought process of the system clearly human understandable?
- ☐ Is the reasoning/thought process free of logical gaps?
- ☐ Is the tone and behavior of the system perceived as benevolent/trustworthy?
- ☐ Is the output of the system free of social biases?

10.3.6 *Adaptive trust dynamics.*

- ☐ For each input the user receive the same output
- ☐ Does the design account for the fact that the user's trust is not static, but will change over time through ongoing interaction?
- ☐ Does evolving trust influence the user's intent and actual usage?

11 Appendix B : Industrial Review Meeting Minutes (V2 to V3) - Meeting Minute

Date: May 8, 2026
Mode: Online Microsoft Teams Meeting
Objective: Evaluation of Version 2 (V2) and identification of structural improvements required for the development of Version 3 (V3).

Participants

- Author/Researcher (Moderator)
- E1: Agile Product Owner
- E2: Agile Scrum Master
- E3: SW Architect – ADAS
- E4: SW Engineer – ADAS
- E5: System Engineer – ADAS

Introduce Task Criticality Assessment. All participants at the meeting agreed that the checklist, as it was, presented the major problem of not being "assessable": there was no clear criteria to understand how to use or assess the final result from the checklist.

Decision: Introduce a dedicated section for task criticality assessment and define explicit *Go/No-Go* evaluation rules as a function of the assessed criticality level.

Checklist Structure and Generalization. The expert team observed that Version 2 still contained several overlapping concepts distributed across multiple sections of the checklist. In particular, E3 and E5 noted that traceability, uncertainty management, and robustness-related concepts appeared repeatedly under different formulations, potentially increasing ambiguity during practical checklist execution.

Additionally, the participants highlighted that some checklist items were too tightly connected to specific organizational or governance practices, reducing the general applicability of the framework across heterogeneous engineering domains.

Decision: Version 3 (V3) will introduce a clearer separation between objective evidence, transparency, robustness, and human oversight dimensions. Redundant items will be merged or removed in order to improve conceptual orthogonality and checklist usability.

Objective Evidence and Auditability. During the discussion of the Objective System Metrics section, E3 emphasized the importance of maintaining measurable and auditable evaluation criteria. The participants suggested that several checklist items should be reformulated to focus more explicitly on validation evidence, configuration management, and traceable verification artifacts rather than generic documentation availability.

The discussion also highlighted the importance of distinguishing between objective system properties and human-perceived trust factors.

Decision: Version 3 will refine the wording of multiple Section 1 items to emphasize measurable validation evidence, auditability, and operational deployment aspects.

Transparency and Human Reviewability. E1 and E2 observed that some items related to transparency and interpretability partially overlapped with robustness and traceability concepts. The participants emphasized that transparency-related requirements should primarily focus on the ability of practitioners to understand, review, and validate generated outputs within realistic engineering workflows.

The discussion additionally highlighted the need to simplify the communication of uncertainty-related concepts.

Decision: Version 3 will simplify the transparency section by consolidating uncertainty-related statements and narrowing the focus toward interpretability and human reviewability.

Operational Robustness and Human Oversight. E4 and E5 noted that certain robustness-related items duplicated concepts already addressed within objective consistency verification. The participants suggested focusing the robustness section more explicitly on operational resilience under degraded, ambiguous, or unexpected conditions.

The discussion also reinforced the importance of maintaining clear human supervisory authority during safety-relevant engineering activities.

Decision: Version 3 will streamline the robustness section by removing duplicated repeatability and documentation-related concepts, while strengthening the distinction between operational robustness and human oversight responsibilities.

Conclusion and Next Steps. The feedback collected during the meeting will be consolidated and used to guide the structural rationalization of the checklist. The primary objective of Version 3 will be to improve conceptual clarity, reduce overlap between checklist items, and increase the operational usability of the framework across different engineering domains.

12 Appendix C: AI Usage Disclosure

In the writing of this document, the following LLMs were used to support the team. Table 3 summarizes the adopted tools and their usages.

LLM Name	Usages
ChatGPT	• Checklist drafting. • LaTeX formatting help. • Document writing (i.e. text enhancement and typo correction).
Gemini	• Checklist drafting. • LaTeX formatting help. • Document writing (i.e. text enhancement and typo correction).

Table 3: Overview of LLMs and their usages.

13 Appendix D: Team Members & Contacts & Useful Links

Name Surname	Student ID	Email
Matteo Pallomo	**s337855**	**s337855@studenti.polito.it**
Eva Ledovskaia	s310269	s310269@studenti.polito.it
Andrea Barbantani	s342878	s342878@studenti.polito.it
Alessio Quaranta	s351986	s351986@studenti.polito.it
Marco Farano	s337643	s337643@studenti.polito.it
Elena Ruberto	s336356	s336356@studenti.polito.it
Fiamma Pia Paternò	s342654	s342654@studenti.polito.it
Michele Barale	s341813	s341813@studenti.polito.it
Federico Ciociola	s341238	s341238@studenti.polito.it
Flora D'Angelo	s336616	s336616@studenti.polito.it

Table 4: Team Members - In bold the name of the team leader.

14 Appendix E - V2 to V3 granular changes.

Section	V2 Element	Change	V3 Modification / Rationale
Global	No explicit task-risk classification layer	Added	Introduced Section 0 to explicitly classify operational risk before evaluating trustworthiness requirements.
Global	Single-checkbox itemized lists	Modified	Introduced numbered lists and right-aligned "Yes/No" checkboxes to enhance visual alignment and evaluation usability.
Section 1	Privacy/cybersec protection statement	Removed	Removed because privacy/cybersecurity governance was considered outside the narrowed operational trust-calibration scope.
Section 1	Separate compliance items	Merged	Compliance/process/output compliance merged into a single requirement to eliminate duplicated evaluation logic.
Section 1	Benchmark test result wording	Modified	Rewritten to emphasize validation activity rather than mere existence of results.
Section 1	Consistency evidence wording	Modified	Rewritten to improve measurability and operational precision.
Section 1	Training/validation data wording	Modified	Simplified into a provenance-oriented statement to reduce wording redundancy.
Section 1	Architecture documentation wording	Modified	Expanded to explicitly include operational limitations.
Section 1	Confidence estimate wording	Modified	Generalized uncertainty representation mechanisms.
Section 1	Version control wording	Modified	Changed from passive availability to active implementation requirement.
Section 1	Validation artifact wording	Modified	Expanded to explicitly include auditability.

Table 5: Granular changelog between Version 2 and Version 3 (Part 1).

Section	V2 Element	Change	V3 Modification / Rationale
Section 2	Transparency section target	Modified	Target narrowed to interpretability and reviewability to reduce overlap with robustness and auditability.
Section 2	Reasoning consistency wording	Modified	Rewritten to reduce overlap with logical consistency validation in Section 3.
Section 2	Predictability wording	Modified	Operational wording refined for consistency.
Section 2	Duplicate uncertainty communication items	Merged	Multiple uncertainty-related statements consolidated into a single item.
Section 2	Input-output traceability statement	Removed	Removed because technical traceability responsibility was consolidated into Section 1 auditability.
Section 2	Expert reviewability wording	Modified	Sentence simplified without semantic change.
Section 3	Robustness section target	Modified	Refocused explicitly on operational robustness.
Section 3	Output repeatability statement	Removed	Removed because repeatability verification was already covered in Section 1 consistency validation.
Section 3	Failure mode documentation statement	Removed	Removed because operational limitations documentation was consolidated into Section 1.
Section 3	Regression validation wording	Modified	Procedural clarity improved.

Table 6: Granular changelog between Version 2 and Version 3 (Part 2).

Section	V2 Element	Change	V3 Modification / Rationale
Section 3	Low-confidence/anomaly wording	Modified	Refocused from uncertainty communication toward anomaly detection.
Section 3	Safe-failure statement	Removed	Removed because it partially overlapped with anomaly detection and human oversight mechanisms.
Section 3	Operational logging wording	Modified	Simplified because post-operation analysis is implicitly enabled by logging availability.
Section 4	Oversight section target	Modified	Expanded to emphasize governance and decision ownership.
Section 4	Human review wording	Modified	Grammatical refinement only.
Section 4	Override availability wording	Modified	Adjusted to avoid implying absolute system availability guarantees.
Section 4	Responsibility boundary wording	Modified	Generalized responsibility assignment beyond a single operator.
Section 4	Reviewer logging wording	Modified	Expanded accountability coverage.
Section 4	Escalation wording	Modified	Low-confidence wording removed because uncertainty handling was consolidated elsewhere.
Section 4	Human supervision wording	Modified	Simplified wording only.
Section 4	Domain expertise wording	Modified	Shortened for brevity.
Section 4	Operational boundary policy statement	Removed	Removed because organizational governance policy was considered outside the operational trust-calibration scope.

Table 7: Granular changelog between Version 2 and Version 3 (Part 3).