The Verification Gate: A Work Method for Generative AI in Consequential Engineering Practice

Working paper, version 0.9. Not peer-reviewed.

Mati Melchior

Independent researcher, Physical AI Safety · Licensed mechanical engineer, MBA · ORCID 0009-0003-9647-1498 · mati@physical-ai-safety.com

September 2026 · DOI: 10.5281/zenodo.22271340 · Method page: physical-ai-safety.com/method/verification-gate

Abstract

Generative AI tools draft, translate, summarise and organise well. They fail at the points where a practising engineer must sign. Five such points recur: a table read from an image, an analogue gauge, a standard clause, a manufacturer fault code, and direct arithmetic. Peer-reviewed benchmarks in building engineering document the first and the last. Manufacturer documentation, a preprint and practitioner reports document the rest. In each case the output is fluent and unmarked. Nothing in the text signals the error. We present the Verification Gate, a work method for using these tools where an error costs money or safety. The method has one axiom and three steps in fixed order. It places one mandatory hold point between tool output and any commitment. It adds four fixed-structure instruments, a five-level task-tagging scale, a tool-free competence test, and a five-step installation procedure. It names seven failure classes (F1–F7) and maps each to the control that catches it. It does not teach tools. It fixes the order in which the engineer and the tool work. This paper fixes the method's vocabulary, states its boundaries, and defines a pre-registered evaluation protocol. No results are reported here, and no data has been collected. The protocol names its fields, its closed lists, and the sample size it needs. Any group that meets the stated criteria and submits verification logs under those fields is a valid application. Version 1.0 will report measured results if and when such logs exist. We release all instruments under CC BY 4.0.

Keywords: generative AI · verification · engineering practice · HVAC · mechanical engineering · work method · hallucination · human oversight

1. Introduction

A split air-conditioning unit shows fault code E3. A technician asks a generative AI tool what E3 means. The manufacturer tables give three answers. Daikin lists E3 as actuation of the high-pressure switch (Daikin n.d.). GREE lists E3 as low pressure (GREE n.d.). Mitsubishi Electric lists E3 as a remote-controller transmission error (Mitsubishi Electric n.d.). The first two are opposite faults. A technician who acts on the wrong one adds refrigerant to an overcharged system, or recovers charge from a system that is already short. A tool that answers from memory, without the unit's own manual in front of it, cannot show which table it used. Its answer carries no marker of origin.

This is the characteristic shape of harm from generative AI in engineering practice. The tool is not weak. GPT-4o passed two South Korean national engineer exams for building mechanical equipment, including air-conditioning and refrigerating machinery, on all five attempts (Choi, Lee, and Kim 2025). Its mean scores were 80.6 and 81.25 out of 100. The same study identifies three limitations: weak advanced reasoning, difficulty with legal questions, and poor interpretation of scientific figures. Those are three item types a technician signs for: a calculation, a regulatory requirement, a drawing.

So the gap is not capability. No step in the workflow stops the output before signature. Engineering already has such steps. A checker's signature precedes drawing release. A pressure test precedes commissioning. Generative AI has no equivalent step, so output can enter a signed report without any check.

This paper supplies the missing hold point as a complete, fixed method. We call it the Verification Gate. The method fits one training session. Its log has 13 fixed fields, so an auditor can count compliance.

Contributions.

1. A vocabulary of eleven defined terms, so that "verified" means one thing (Section 3).

2. A taxonomy of seven failure classes, F1–F7, specific to engineering deliverables, each tied to published evidence (Section 4).
3. The method itself: one axiom, seven principles, three steps, one gate, and a gate decision rule (Section 5).
4. Four instruments with fixed structure, including a 13-field verification log with closed lists, released as templates (Section 6).
5. A five-level task-tagging scale and a five-step organisational installation procedure (Sections 7 and 8).
6. A pre-registered evaluation protocol, with research questions fixed before data collection (Section 10).

We do not claim that the seven classes are exhaustive, or that the method makes output correct, and we say so plainly. The claim we defend is narrower, and it is falsifiable. Under the method, every item in an output is either traced to an authoritative source or blocked, and the trace is countable.

2. Background

2.1 Where the tools fail in engineering tasks

Sources: two peer-reviewed benchmarks, one national licensing-exam study, one ACM poster, one preprint, and one construction-law practitioner. Figure 2 shows the key numbers.

Tables inside codes and standards. AECBench tests nine large language models on 4,800 questions written by engineers in architecture, engineering and construction (Liang et al. 2026). Two tasks require reading a table inside a building code. Baseline accuracy was 45.27 and 40.65 percent. When the same table was supplied as expert-written running text, accuracy rose to 98.94 and 87.27 percent. The models knew the content. They failed to read it out of a table. The study concludes that the models show deficits "in interpreting knowledge from tables in building codes, executing complex reasoning and calculation, and generating domain-specific documents."

Licensing-exam errors by type. Choi, Lee, and Kim (2025) ran GPT-4o five times on two Korean national engineer exams. They classify 172 errors into three types. Knowledge errors (104) occur on non-calculation questions. Reasoning errors (48) occur on calculation questions. Context errors (20) occur on questions that require reading a figure or schematic. The model's mean response consistency was 97 percent. So a wrong answer is wrong on the next attempt too. That consistency is itself a finding. Repeating the query does not surface the error.

Schematics. An HVAC piping-and-instrumentation study concludes that "directly applying LLMs to P&ID digitization is highly challenging" (Lu et al. 2025). Usable results required an engineered pipeline, not an uploaded image and a question.

Analogue instruments. A benchmark of visual measurement reading reports that frontier vision-language models "struggle with measurement reading in general" (Lin et al. 2026, preprint). The authors attribute the difficulty to fine-grained spatial grounding: models read printed digits and misplace the pointer. We cite the pattern and not the numbers, because the work is under review.

Fabricated references. Hallucination in language models is a documented class with its own taxonomy (Huang et al. 2025). In construction documents it takes a specific form. A construction-law practitioner describes "phantom codes": standard designations that follow the real numbering convention and do not exist (Thomas 2026). Thomas reports that such codes pass document review and surface later, when a testing laboratory cannot locate the method. The cost by then is a change order or a claim.

2.2 Why order of use matters

A field experiment with nearly one thousand secondary students tested two tool designs (Bastani et al. 2025). Open access to a standard chat tool raised practice grades by 48 percent. After the researchers removed access, those students scored 17 percent below students who never had the tool. A version with safeguards that withheld direct answers raised grades by 127 percent, and the authors report that the later drop was largely mitigated. They attribute the difference to how the tool enters the task. The study is in secondary mathematics, not engineering practice. We take it as motivation for Principle P1, and Section 10 defines a protocol to test the principle in our own setting.

2.3 Professional bodies

ASHRAE adopted a policy on artificial intelligence in February 2025 and applies it to its standards process (ASHRAE 2025). ASTM International has a comparable policy (ASTM 2025). Both, as we read them, treat AI

output as material that a human must review and take responsibility for. Yet neither specifies a method for the practitioner. This paper supplies one.

2.4 What the method is not

The method is a work procedure, in the tradition of the inspection hold point, applied to a new source of unverified numbers. It does not detect hallucinations, tune prompts, or rank tools.

3. Definitions

The method uses the terms in Table 1 in the fixed sense given there and in no other sense.

Table 1. Definitions.

Term	Definition
Tool	Any generative AI system, text, vision or combined, that receives a prompt and returns output. The method names no products.
Output	Anything the tool returns, in any format, before it passes the gate.
Authoritative source	A document with an owner and standing: manufacturer manual, standard in its official text, approved organisational procedure, signed drawing, direct measurement, manufacturer data table. A tool is never an authoritative source. The output of a second tool is not one either.
Item	A unit of verifiable information: a number, an identifier, a clause, a factual claim, a reported action.
Verification	Comparing one item against one authoritative source and recording the result.
Gap	Any difference between output and source, of any size.
Deliverable	Output that has passed the gate in full and has a complete verification log. Only a deliverable may be signed, submitted, or used for a decision.
Commitment	Any act that moves a deliverable from draft to reality: signature, submission, purchase order, work instruction, client report, entry into an organisational system.
The gate	The mandatory hold point between output and commitment. Step C of the process.
Verification log	The instrument that records verification. Specified in Section 6.1.
Level	The permitted degree of tool use for a given task, on the five-level scale of Section 7.

4. Failure Taxonomy

We name seven failure classes. Each is a recurring pattern in tool output on engineering deliverables. Table 2 gives the class, the pattern, the engineering consequence, and the evidence status. We describe patterns, not accuracy percentages, except where a peer-reviewed source reports the figure. Percentages depend on the tool generation and age within months. Whether the patterns are more stable is a question for version 1.0 (Section 11).

Table 2. Seven failure classes.

#	Class	Pattern	Why it matters in engineering	Evidence
F1	Table from image	A query on a photographed table is far less accurate than the same query on the table as text	Unit tables, code tables, data sheets all arrive as images	Peer-reviewed (Liang et al. 2026)
F2	Analogue gauge	Digits and labels are read well. Pointer position is misread	Pressure, temperature, current. The reading enters a signed report	Preprint (Lin et al. 2026)
F3	Standard clause	Clause numbers and content are invented in a plausible structure	A legal inspection interval or a safety requirement that does not exist	Professional (Thomas 2026); mechanism (Huang et al. 2025)
F4	Fault code	Interpreted from memory. The same code means different, even opposite, things across manufacturers	Wrong diagnosis, wrong part, unsafe action	Manufacturer tables show the conflict (Section 1). Tool error inferred, not yet measured
F5	Direct arithmetic			Peer-reviewed (Choi, Lee, and Kim 2025; Liang et al. 2026)

#	Class	Pattern	Why it matters in engineering	Evidence
		Errors in calculation performed inside the model, on formula selection and on unit conversion	Load, flow, stress. The number enters a design decision	
F6	Silent completion	A part number, date, model or action that was never supplied is filled in with full confidence	A report states an action that was not performed	Mechanism (Huang et al. 2025)
F7	Right answer - wrong reasoning	Correct result with an incorrect justification	The practitioner absorbs a wrong justification that looks sound	Field observation. Not yet measured

The seven classes share one property. In every case the output carries no external marker of error. So verification cannot depend on suspicion. It runs on every item and leaves a record.

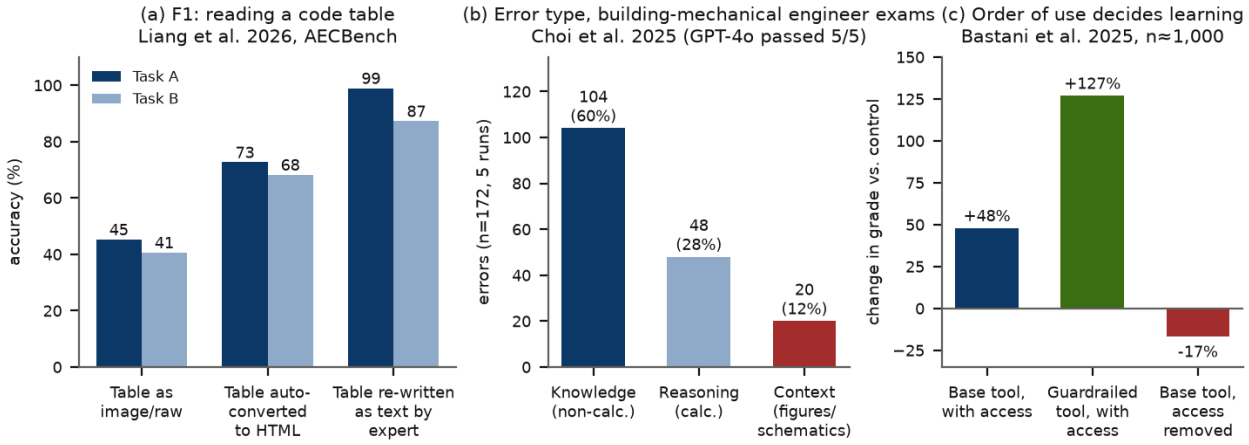

Figure 2. Published evidence behind three classes. (a) F1: accuracy on two table-reading tasks inside building codes, by table format, nine models (Liang et al. 2026). (b) Error type on two national engineer exams for building mechanical equipment that the model passed 5 of 5 times (Choi, Lee, and Kim 2025). Each error type maps to a question type. (c) Effect of tool design and of tool removal on learner grades (Bastani et al. 2025). Values are as reported in the sources' abstracts.

5. The Method

5.1 Axiom

> **A number that has not been verified does not exist.**

Operational meaning: a value that originates in tool output has no standing as data until someone checks it against an authoritative source. Until checked, the value is a draft. It may not enter a signed or submitted document, or a decision. The item types covered are those of Table 4, field 6.

Every rule in Sections 5 to 8 derives from this axiom.

5.2 Principles

P1 Fixed order. The three steps run in the order given, with no skipping. An estimate made after seeing the answer anchors to it. Step A therefore precedes Step B, without exception (motivation: Bastani et al. 2025).

P2 The tool drafts. It does not compute. For any calculation, the tool may propose a formula and identify variables. The arithmetic runs in a spreadsheet or a deterministic calculation tool, not in the model.

P3 Source before memory. For every normative item, the engineer attaches the authoritative source to the prompt or checks it afterwards. Interpretation from the tool's memory is not admissible, even when it is correct.

P4 Declare what is missing. Every prompt includes the instruction: "Do not invent. Write MISSING." An item marked missing is a correct result. An item completed without a source is a failure.

P5 The record is part of the deliverable. A deliverable without a verification log is not a deliverable. No exceptions.

P6 Measure competence without tools. At least one assessment component in any training runs with no tool access. Section 2.2 gives the reason.

P7 Tool independence. The method speaks of tasks, never of products. No product name appears in the method. Product names appear only in training material, marked as examples.

5.3 Three steps and one gate

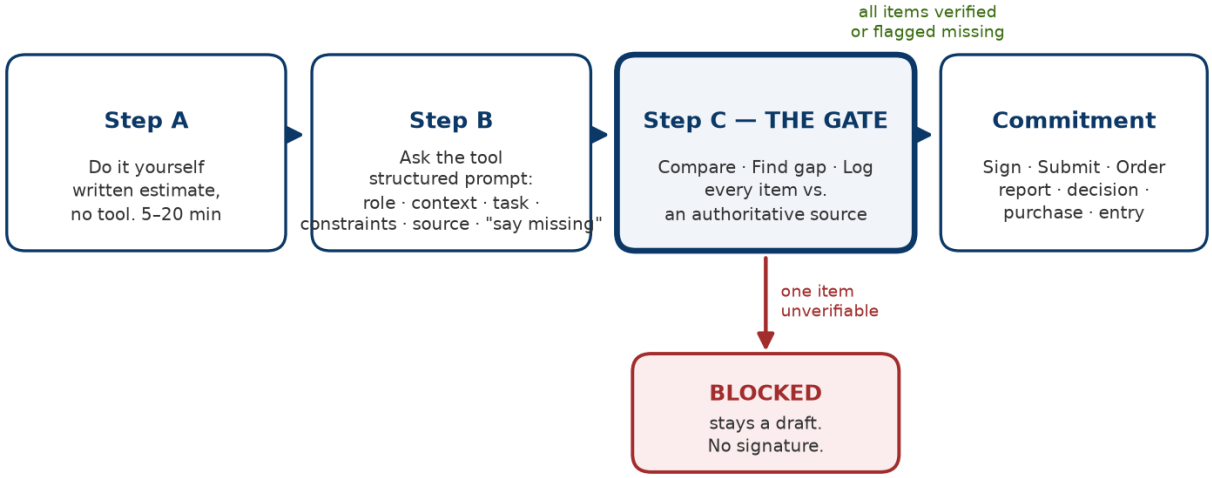

Figure 1. The process. Three steps in fixed order. The gate sits between tool output and any commitment. One unverifiable item blocks the whole output.

Step A. Do it yourself. A first pass at the task with no tool. An estimate, a manual reading, a written hypothesis, a rough draft. Step A takes five to twenty minutes. Its purpose is a recorded independent judgment, not an accurate one. The engineer writes the estimate down before opening the tool. An unwritten estimate does not count as Step A.

Step B. Ask the tool. A structured prompt with six mandatory parts:

1. Role: who the tool simulates (for example, a senior maintenance technician).
2. Context: equipment type, site, environment, which values come from measurement and which do not.
3. Task: what to produce, in a defined format.
4. Constraints: units, governing standard, length.
5. Source: the authoritative document, attached when relevant.
6. The missing instruction: "Do not invent. Write MISSING. If you lack a datum, ask before you write."

A prompt without parts 4 and 6 is not a prompt under this method.

Step C. The gate: compare, find the gap, log. The engineer checks every item in the output against an authoritative source, records the gap, and writes the result in the verification log. Table 3 gives the decision rule.

Table 3. Gate decision rule.

Item state	Action
Verified, matches source	Passes. Record source and location.
Verified, gap found	Corrected to the source. Record original output, source, correction, gap class.
Not verifiable, no source available	**Blocked.** Item marked "not verified". It does not enter the deliverable. If the deliverable depends on it, the deliverable stays a draft.
Tool wrote MISSING	Correct behaviour. Recorded as such.

Gate rule. An output with one blocked item is not a deliverable. Partial verification is not a state in Table 3. One blocked item holds the whole output at draft.

5.4 Past the gate: commitment

Only a deliverable may proceed to commitment. A deliverable is output whose every item passed verification or carries the MISSING mark, with a complete verification log.

5.5 Which control catches which failure

Figure 3 maps the seven classes to the controls in the method. Every class has at least one primary control, and the gate is primary for six of seven. F7 is the exception. A right answer with wrong reasoning passes a source check. It is caught by Step A, where the engineer's own reasoning exists to compare against, and by the error-hunt quiz.

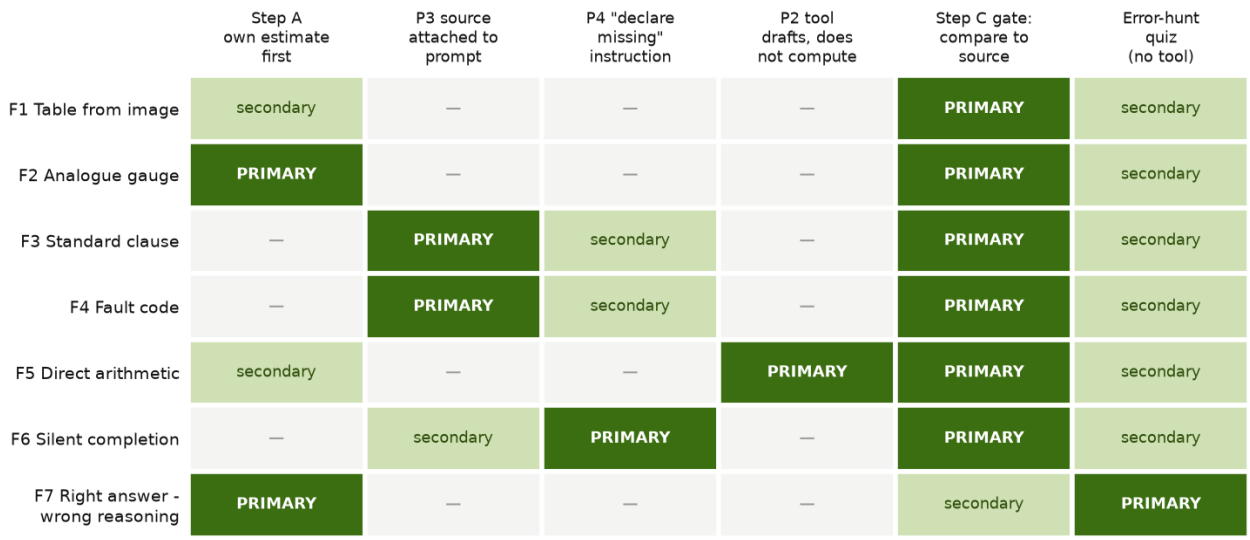

Figure 3. Coverage matrix. Rows are failure classes. Columns are controls in the method. Primary: the control is designed to catch that class. Secondary: it catches the class as a side effect. This is a design mapping, not a measurement. We plan to test it against logged gaps in version 1.0.

6. Instruments

Four documents with fixed structure. The structure is fixed so that anyone can count the results (Section 10).

6.1 The verification log (Appendix A)

One log accompanies every deliverable. Table 4 gives the 13 mandatory fields in order. Fields 3, 4, 6, 10 and 11 are closed lists. Closed lists make the log countable (Section 10).

Table 4. Verification log fields.

#	Field	Type	Values
1	Deliverable ID	text	task or session code
2	Date	date	
3	Discipline	list	Mechanical · HVAC · Other
4	Deliverable type	list	Report · Procedure · Calculation · Specification · Table · Translation · Other
5	Item checked	text	short description
6	Item type	list	Number · Part no./Model · Standard clause · Fault code · Frequency · Reported action · Term translation · Other
7	What the tool said	text	verbatim output
8	Verification source	text	name of the authoritative document
9	Location in source	text	page / clause / table
10	Result	list	**Match · Gap · Not verifiable · Missing (flagged by tool)**
11	Gap class	list	blank if match. Otherwise **F1–F7** or "other"
12	Correction	text	final value

#	Field	Type	Values
13	Verification time (min)	number	

6.2 AI use declaration (Appendix B)

Attached to every submission and every organisational deliverable. Seven fields: tools used. Parts of the deliverable they touched. Parts produced by hand. Items verified, and the source for each. Items found wrong and corrected. Level on the scale (Section 7). Signature and date.

6.3 Error-hunt quiz (Appendix C)

Measures competence without tools. Six tool outputs from the working domain, each containing one error of class F1–F7. The candidate identifies the error, classifies it, and names the verification that would have caught it. It runs with no tool access, so it is the one assessment component that no tool can bypass.

6.4 Task-tagging table (Appendix D)

Applies the five-level scale to the tasks of an organisation or a course. Fields: task · deliverable · level 0–4 · reason · approver · date.

7. The Five-Level Scale

The organisation tags every task before work begins. The tag is a written entry in the task-tagging table (Appendix D). Figure 4 and Table 5 give the scale.

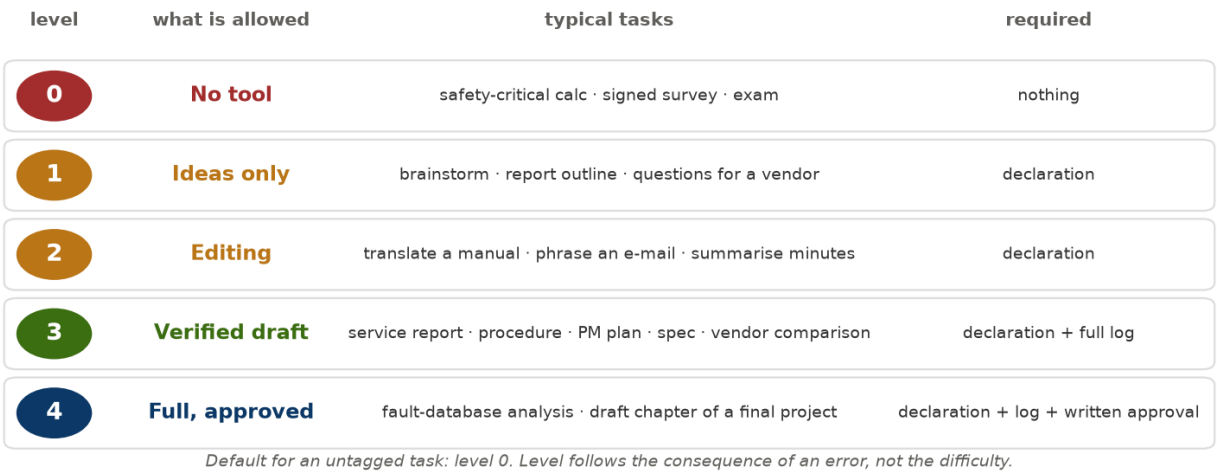

Figure 4. The five-level scale. Level follows the consequence of an error, not the difficulty of the task.

Table 5. Levels and requirements.

Level	Name	Permitted	Required
0	No tool	No tool use	nothing
1	Ideas only	The tool proposes directions, structure, questions. All content is written by hand	declaration
2	Editing	The tool improves text written by hand: phrasing, translation, summary, language	declaration
3	Verified draft	The tool drafts the deliverable. Every item passes the gate (Section 5.3)	declaration + full log
4	Full, approved	Extensive use including analysis and organisation. Only with the supervisor's explicit approval	declaration + full log + written approval

Three tagging rules apply. The default for an untagged task is level 0, until someone tags it otherwise. Level follows the consequence of an error, not the difficulty. A simple calculation that enters a signed survey is level 0. An organisation may tighten a level. It may not loosen a level beyond what the method permits.

8. Installation in an Organisation

Installation takes five steps, in fixed order (Figure 5, Table 6). Training is step 4.

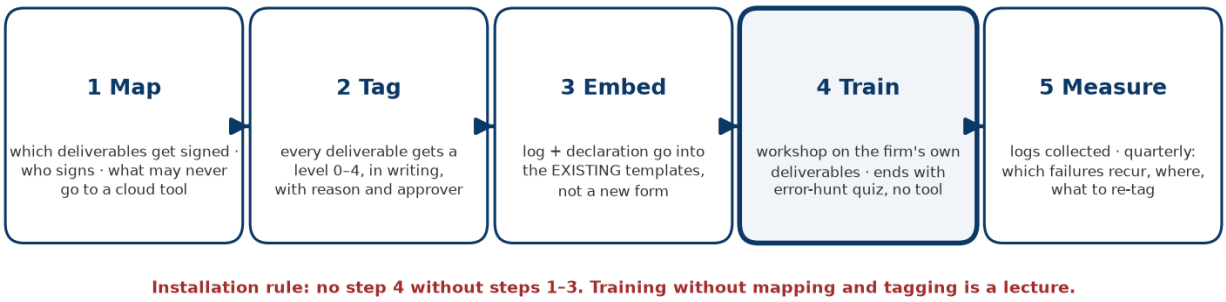

Figure 5. Five-step installation. Training is step 4, not step 1.

Table 6. Installation steps.

Step	Name	What is done	Product
1	Map	Which deliverables get signed or lead to commitment. Who signs. Which documents may never be uploaded to a cloud tool (confidentiality, classification, IP)	list of deliverables + "never to cloud" list
2	Tag	Every deliverable on the list receives a level, in writing, with reason and approver	task-tagging table (Appendix D)
3	Embed	The log and the declaration go into the organisation's existing templates. Not a new form	updated templates
4	Train	A workshop on the organisation's own deliverables, using the three-step process. It ends with the error-hunt quiz, no tool	trained staff · quiz results
5	Measure	Collect the logs. Quarterly: which failure classes recur, in which deliverables, and what changes in tagging	quarterly report · updated tags

Installation rule. Step 4 requires the outputs of steps 1 to 3: the deliverable list and the tagging table. The workshop format is active, with participants working on the organisation's own deliverables. Active formats outperform lectures in undergraduate STEM (Freeman et al. 2014). We assume, and plan to check, that the result transfers.

9. Boundaries

The method applies under three conditions. Outside them it gives no assurance.

9.1 The method does not apply when:

- There is no authoritative source. If no document with an owner exists, there is nothing to verify against. The gate blocks the item. That is the specified outcome, not a failure of the method.
- The task is creative or open-ended and leads to no commitment. Examples: brainstorming and personal study. These are level 1 tasks, and the gate does not apply.
- The "tool" is a validated deterministic system. Certified engineering calculation software is not a tool in the sense of this method.

9.2 The method does not promise:

- That the deliverable is correct. It promises that someone checked every item against a source and recorded the result. If the source is wrong, the deliverable is wrong, and the log shows exactly where.
- Speed. Step C adds time per deliverable. Field 13 records it, and we plan to report it (Section 10). Whether it is less than the cost of a signed error is an empirical question.

9.3 The method does not replace:

- Professional competence. Verification requires the competence to read the source. The method assumes that competence and does not supply it.
- Professional responsibility. Responsibility stays with the signer. ASHRAE (2025) and ASTM (2025) take the same position.

10. Evaluation Protocol for Version 1.0

10.1 Who may submit a log

A group applies the protocol when it meets three inclusion criteria and submits verification logs under the fields of Appendix A. First, every participant in the group signs off on mechanical or HVAC engineering work. Second, the group uses the method as its spine and the instruments of Section 6 in its own work. Third, every participant submits logs at a fixed rate over a fixed period. The group is therefore also the measurement instrument. Figure 6 gives the timeline.

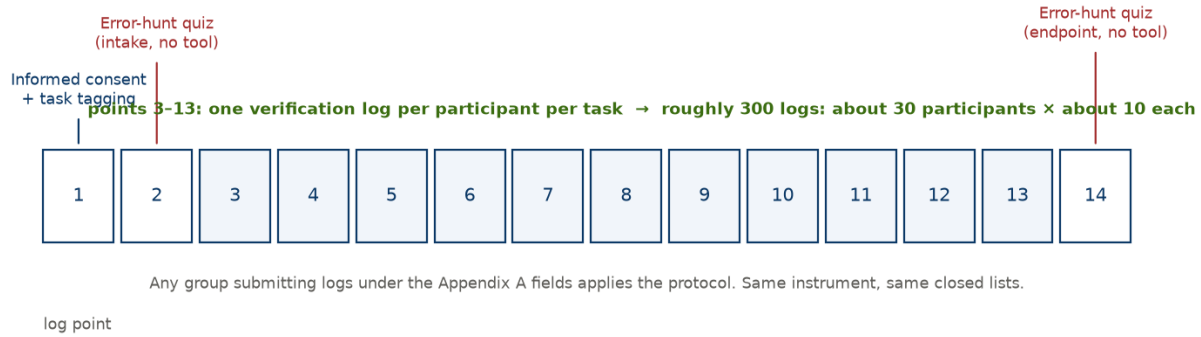

Figure 6. Evaluation timeline. Consent and tagging at log point 1. Pre-test without tools at intake. One verification log per participant per task at log points 3–13. Post-test without tools at the endpoint.

10.2 Data

Each participant submits about ten verification logs (Appendix A). Fields 3, 4, 6, 10 and 11 are closed lists. The required volume is roughly 300 logs: about 30 participants submitting about 10 each. Each log row is one verified item. The error-hunt quiz runs twice, at intake and at endpoint, with matched difficulty and no tool access.

10.3 Research questions, fixed before collection

1. Distribution of gap classes F1–F7 by discipline and by deliverable type.
2. Share of items blocked (not verifiable) by item type.
3. Mean verification time by deliverable type.
4. Change along the log sequence: does the share of gaps caught rise with practice?
5. Comparison between the two disciplines.
6. Change in error-hunt score from intake to endpoint.

10.4 Consent and privacy

Written informed consent at intake (Appendix F). We analyse data in aggregate and anonymised form only. No participant or employer is identifiable in any publication. Participation in the research is separable from the work itself. A participant may decline the research with no penalty.

10.5 Analysis

Descriptive statistics for questions 1 to 3 and 5. For question 4, gap rate per log over the sequence, with confidence intervals. For question 6, paired comparison of pre and post scores. All counts derive from closed-list fields. We plan to release the analysis script with the data.

10.6 What the data will not say

The protocol collects verification logs from participants. Its data says nothing about how an organisation installs the method. We will document organisational cases (Section 8, step 5) separately.

11. Limitations

Evidence base. We have collected no data. We have recruited no group. Version 0.9 rests on external published evidence, and we state this plainly. None of it measures this method. It measures the failures the method targets. The mapping in Figure 3 is a design claim, not a result.

Single author. The method's author wrote the instruments, the coding rules and the taxonomy. We expect agreement with the method's intent to run higher than an outside coder would produce. A taxonomy's author codes it generously for the same reason. We release the instruments and the coding rules so that others can run the protocol independently.

Self-report in the log. Field 13 (verification time) and field 7 (verbatim output) depend on the participant. We plan to spot-check a random sample of logs against the participants' saved tool sessions.

Generation drift. Tool behaviour changes between versions. A class that is frequent in 2026 may be rare in 2028. The method speaks of task types so that it survives this drift. We plan to revise the taxonomy against the logged data in version 1.0.

F7 is not yet measured. Right answer with wrong reasoning enters the taxonomy from field observation. It is the class most likely to change in version 1.0.

12. Conclusion

Generative AI tools reach engineering practice with high general scores and specific, repeatable failures at the points of signature. We defined a hold point for that situation. We stated one axiom and derived seven principles from it. We fixed three steps and placed one gate between output and commitment, with a four-way decision rule and no partial pass. We released four instruments with closed-list fields and a five-level scale tied to the consequence of an error. We fixed the sample the protocol needs and the fields it collects. A group that meets the three inclusion criteria and submits logs under those fields produces measured data on the method itself. We release every instrument, so that anyone can test the method, contest it case by case, or improve it.

Reproducibility and Ethics Statement

We release six instruments under CC BY 4.0. They are the verification log template (bilingual, closed lists, summary sheet), the AI use declaration, and the error-hunt quiz structure and rubric. They also include the task-tagging template, the prompt template, and the informed-consent text. Appendix G is this paper's own verification log. It has 23 items: 14 matches, 6 gaps corrected before release, 2 items removed as unverifiable, and 1 planned figure marked as such. We plan to add the anonymised log dataset, the coding rules, and the analysis code in version 1.0. Participants join the research by written consent and may withdraw at any time. We will publish no personal or employer data.

Licence and Attribution

References

ASHRAE. 2025. *ASHRAE Policy on Artificial Intelligence.* February 2025. Atlanta, GA: ASHRAE. https://www.ashrae.org/file%20library/about/position%20documents/ashrae-ai-policy-feb-2025.pdf

ASTM International. 2025. *ASTM Policy on the Use of Artificial Intelligence.* West Conshohocken, PA. https://store.astm.org/media/wysiwyg/ASTM_AI_Policy.pdf

Bastani, H.; Bastani, O.; Sungu, A.; Ge, H.; Kabakcı, Ö.; and Mariman, R. 2025. Generative AI Without Guardrails Can Harm Learning: Evidence from High School Mathematics. *Proceedings of the National Academy of Sciences*, 122(26): e2422633122. https://doi.org/10.1073/pnas.2422633122

Choi, H.; Lee, J.; and Kim, J. 2025. Performance Evaluation of GPT-4o on South Korean National Exams for Building Mechanical Equipment Maintenance. *Scientific Reports*, 15: 30436. https://doi.org/10.1038/s41598-025-16118-x

Daikin. n.d. *Error Code Reference (Service Manual Extract).* Technical document. https://www.daikin.com/-/media/Project/Daikin/daikin_com/products/ac/services/error_codes/pdf/sm-ts3_p1-6_errorcode-pdf.pdf

Freeman, S.; Eddy, S. L.; McDonough, M.; Smith, M. K.; Okoroafor, N.; Jordt, H.; and Wenderoth, M. P. 2014. Active Learning Increases Student Performance in Science, Engineering, and Mathematics. *Proceedings of the National Academy of Sciences*, 111(23): 8410–8415. https://doi.org/10.1073/pnas.1319030111

GREE. n.d. *Understanding GREE Mini-Split Error Codes.* Technical note. https://www.greecomfort.com/news-and-events/understanding-gree-mini-split-error-codes/

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; and Liu, T. 2025. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, 43(2). https://doi.org/10.1145/3703155

Liang, C.; Huang, Z.; Wang, H.; Chai, F.; Yu, C.; Wei, H.; Liu, Z.; Li, Y.; Wang, H.; Luo, R.; and Zhao, X. 2026. AECBench: A Hierarchical Benchmark for Knowledge Evaluation of Large Language Models in the AEC Field. *Advanced Engineering Informatics*, 71: 104314. https://doi.org/10.1016/j.aei.2026.104314

Lin, F.; Liu, Y.; Xu, H.; Yue, C.; He, Z.; Zhao, M.; Chen, M. H.; Liu, J.; Yao, J. G.; and Yang, X. 2026. Do Vision-Language Models Measure Up? Benchmarking Visual Measurement Reading with MeasureBench. arXiv:2510.26865, v2 (24 March 2026). Preprint, not peer-reviewed. https://arxiv.org/abs/2510.26865

Lu, C.; Walker, S.; Struck, C.; Itard, L.; and Saelens, D. 2025. Poster Abstract: P&ID-to-Graph: LLM-Assisted Digitalization of HVAC Diagrams. In *Proceedings of the 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '25)*, 296–297. https://doi.org/10.1145/3736425.3772104

Mitsubishi Electric. n.d. *Mr. Slim Error Code List.* Technical document. https://www.mitsubishielectric.com.sg/getmedia/1a8fe41a-9b17-49f2-a8b5-3e4842aa99ba/MrSlimerrorCode.pdf

Thomas, B. 2026. When "Smart" Tools Make Dumb Mistakes: AI's Hidden Risks in Construction Documents. Gausnell, O'Keefe & Thomas LLC, January 2026. https://gotlawstl.com/when-smart-tools-make-dumb-mistakes-ais-hidden-risks-in-construction-documents/

Appendices (released separately, CC BY 4.0)

A. Verification log template (xlsx, bilingual, closed lists, summary sheet). Structure per Section 6.1. **B.** AI use declaration (fixed text). **C.** Error-hunt quiz: structure, rubric, and item generator. **D.** Task-tagging table template (xlsx). **E.** Prompt template per Section 5.3, Step B. **F.** Informed-consent text per Section 10.4. **G.** This paper's own verification log: 23 items, each number traced to its source (xlsx).

Version History

Version	Date	Change
0.9	09/2026	Foundation document. Method fully specified. Evidence from external sources only.
1.0	—	+ measured results, only when logged data exists
2.0	—	+ certification track · only when a certifying body exists

The DOI 10.5281/zenodo.22271340 identifies version 0.9, and the Concept DOI covering all versions is on the Zenodo record page.

Every number in this document traces to a cited source. Numbers we could not trace were left out. The document's own verification log is released with the appendices.