UNIVERSITY OF OSLO

EE146

A structured AI code-review framework for validating R-based cost-effectiveness models

Benjamin P. Geisler, M.D., M.P.H. · Handout, ISPOR Europe 2026

validateHE

Department of Health Management and Health Economics, University of Oslo, Norway · b.p.geisler@medisin.uio.no

Pages: 1 the poster · 2 methods and checklists · 3 results, limitations, and a synthetic report · 4 taking part, FAQ, and references

Background

- Errors in submitted models are common: **39 of 41** NICE single technology appraisals had at least one reported error [2].
- **Validation** judges a model's accuracy in making relevant predictions. The ISPOR-SMDM task force distinguishes face validity, verification, cross-validation, external, and predictive validation; transparency lets others judge all of them [1].
- Validation is rarely reported: fewer than **4%** of recent early breast cancer model evaluations report validating the computerised model [3]; **30%** of recent cancer evaluations report any validation [4].
- Expert review is scarce and slow; AI agents can read, run, and test model code.
- PubMed (October 2026): no peer-reviewed study of AI for validating health-economic models; LLM studies so far build or replicate models [5, 6]. Other AI model-QC studies at this meeting test against seeded errors, an AI-built model, or human QC, mostly in Excel [7–9].

Aim: to describe validateHE, an AI agent skill run in Claude and OpenAI Codex, and its first use on four "production" R models.

What was evaluated

What	Details
Reports	Four, generated in Claude: one on a pre-release version (July 2026), three on v0.14.0 (August–October 2026)
Models	Four "production" R models: three partitioned survival, one decision tree
Modules run	Verification on all four; transparency and face validity on two (one model had no documentation; one run was a verification-only rerun)
Repeat runs	Two models re-run in OpenAI Codex for comparison
Used on production models	Transparency, face validity, and verification of partitioned survival and decision-tree models
Tutorial models only (so far)	Verification of Markov, DES, and microsimulation models; cross-validation; external validation
Next: checking the findings	Owners confirm or reject each finding; independent human review of a subset; seeded-error models; council with vs. without deliberation

Table 1. What the first results rest on, and how the findings will be checked.

Results

4	3 + 1	287	20
"production" R models: ASCERTAIN + METIMMOX	partitioned survival / decision tree	actionable findings	critical findings

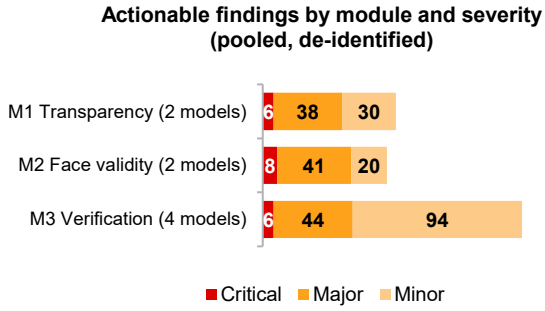


Figure 1. Findings from the Claude runs, pooled across the four models.

- All four reports were generated in Claude (Opus 5; one model with Opus 5.5). No comparator models or external data were supplied, so Modules 4–5 did not run.
- Per model reviewed, each module raised about 35 actionable findings (M1 37, M2 35, M3 36); 7% were critical, 43% major, and 50% minor.
- Every report had at least one major or critical finding. Overall assessments: 1 *fit only after revisions*, 2 *not yet fit*, and 1 *not assessable on the reviewed evidence*.
- Repeat runs in OpenAI Codex, major/critical findings vs. Claude: model A, 49 (GPT-5.6 Terra) vs. 50 (Claude Opus 5); model B, verification only, 15 (GPT-6.1 Sol) vs. 2 (Claude Opus 5.5).

Conclusions

1. Agents can run a structured, checklist-based validation of "production" R models and return evidence-cited findings with recommended fixes.
2. Human and AI validation might complement each other: people bring scoping, subject-matter knowledge, judgement, and experience; agents may be stronger on code and detail.
3. Whether the agents flagged the right issues still has to be confirmed by model owners and human reviewers.

Future directions

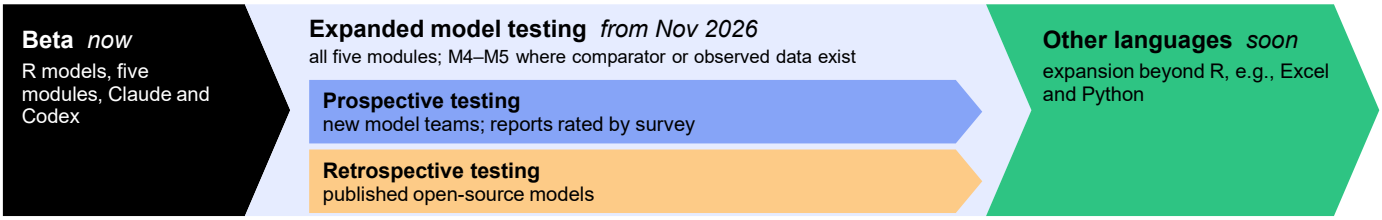


Figure 2. Roadmap: two parallel testing streams that converge on other languages (dates are targets).

validateHE is an AI agent skill: instructions, checklists, and base-R scripts that an AI agent (Claude or OpenAI Codex) follows. It works on a read-only copy of an R model repository and its documentation, runs one module per ISPOR-SMDM validation type [1], and writes an HTA-oriented report (HTML and PDF). Supported model types: cohort Markov, decision tree, partitioned survival, discrete-event simulation, and microsimulation.

Five modules, one per validation type

Module	What it asks	How validateHE does it	Tested on
1 Transparency	Is the model documented clearly enough to review?	Documentation reviewers check completeness and internal consistency of the README, dossier, and code comments; CHEERS 2022 when a manuscript is supplied.	production models
2 Face validity	Do the structure, inputs, and outcomes look plausible?	Simulated “evidence review group”: six subagents with separate roles each write an independent critique, blind to the others; an optional deliberation round follows.	production models
3 Verification	Does the code do what the modellers intended, correctly?	Black-box tests (including unit tests of each function), white-box review with root-cause tracing, and a coding-standards review.	production (PSM, decision tree); tutorials only (Markov, DES, microsim.)
4 Cross-validation	Do other models of the same problem reach compatible answers?	Compares results with a supplied comparator model or published results: corroboration, not proof.	tutorial models only
5 External validation	Does the model reproduce what actually happened?	Compares predictions with supplied observed data (life tables, registries, trials).	tutorial models only

Table 2. Modules, and where each has been tested: on the four “production” models of the first results, or so far only on public tutorial models (dry runs).

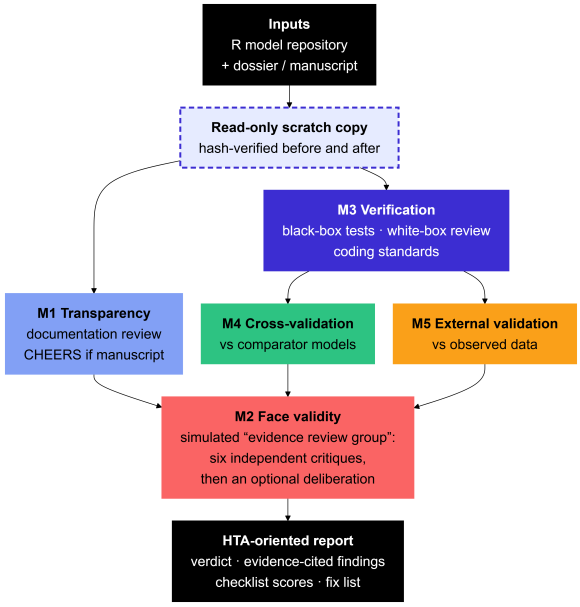


Figure 3. Modules and run order; M4 and M5 run in parallel.

Run order

M1 runs alongside everything; M4 and M5 run in parallel once M3 has a base case; the review group runs last and reads all technical findings.

Black-box review

Runs the model and tests its outputs without reading the code: extreme values (e.g., all utilities = 1 ⇒ QALYs = life-years), cohort conservation, swapped strategies, repeatability; plus unit tests of each function on the path from inputs to results.

White-box review

Reads the critical calculations line by line, traces results to inputs and back, and finds the root cause of black-box failures.

Coding standards

Consistency first: hidden global variables, duplicated definitions that drift, and silently swallowed errors outrank style. Style is judged against a configurable profile (default: tidyverse, roxygen2, DARTH naming; or the team's own conventions).

Dependency map

Function inventory and call graph: decides which functions get unit tests and lets the white-box reviewer trace failures to their source.

Simulated “evidence review group”

Six subagents with disjoint roles (clinician, technical modeller, health economist, HTA assessor for a named jurisdiction, patient representative, payer) each write an independent critique, blind to the others; an optional deliberation round follows, and the minutes go into the report. Items that ask for human expert review stay capped unless one is documented.

Guardrails

Never edits the model (temporary copy, before/after content hash) · never applies fixes · every finding cites file and line · states what it could not test in that run, and why · never fills in comparator or observed values from general knowledge.

Checklist coverage

Checklist	What it is	Which modules score it
CDA-AMC (CADTH) Model Validation Tool	HTA model-validation checklist, 53 items	M1: decision problem · M2: model specification · M3: clinical-effectiveness modelling, transparency, black-box (incl. PSA when present) and white-box items · M4/M5: comparison with submitted or observed evidence. Excel-specific items are scored through their R equivalents.
TECH-VER	Verification checklist in five stages	M3: input, event/state, result, and uncertainty calculations, plus programming quality · M2: outcomes vs. expert knowledge · M4/M5: model-comparison and life-table rows.
AdViSHE	Validation-assessment tool, parts A–E	M2: conceptual model, input data, and outcome plausibility · M3: computerised-model checks · M4: comparison with other models · M5: comparison with empirical data · M1: where each item is reported.
CHAOS (draft guideline)	Checklist for open-source health-economic models	M1: metadata · M3 coding standards: coding quality, reproducibility, packages and dependencies.
CHEERS 2022	Reporting standard, 28 items	M1, only when the documentation includes a full-length manuscript or preprint.

Table 3. Which module scores which checklist (items paraphrased; items owned by a module that was not run are reported as Deferred).

Findings by review method

Method	Models	Critical	Major
Transparency (M1)	2	6	38
Face validity council (M2)	2	8	41
Black-box tests (M3)	4	2	12
White-box review (M3)	4	3	18
Coding standards (M3)	4	1	14
Total		20	123

Table 4. Critical and major findings by method, pooled across the Claude runs; 144 minor findings bring the total to 287. M1 and M2 ran on two models (one had no documentation; one was a verification-only rerun). Per model reviewed: M1 37, M2 35, M3 36.

Checklist results, pooled

Checklist (models)	Pass	Part.	Fail	NA	Not t.	Def.
CDA-AMC (4)	74	29	21	44	44	16
TECH-VER (3)	105	23	20	83	35	2
AdViSHE (4)	9	9	23	1	0	18
CHAOS (4)	1	19	12	0	0	8

Table 5. Item verdicts summed over the reports: Partial, Not-tested, and Deferred abbreviated. One report scored TECH-VER by stage, so it is left out of that row. CHEERS did not apply (no manuscript, or M1 not run).

Limitations

- **No comparison with human review yet.** The head-to-head with hackathon review promised in the abstract was not run, and owners or independent reviewers have not yet confirmed the findings. Planned: owner confirmation, independent human review of a subset, seeded-error models.
- **Small, convenience sample.** Four models from two projects linked to the author; three partitioned survival models and one decision tree; all in R.
- **Partial module coverage.** Modules 4–5 did not run (no comparator models or observed data supplied); M1 and M2 ran on two of the four models.
- **Not yet used on production models.** Verification of Markov, DES, and microsimulation models, and Modules 4–5, have so far been tested only on public tutorial models.
- **Counts vary between agents and runs.** Repeat runs differed widely (15 vs. 2 major/critical findings on one model). A count of findings is not a measure of accuracy.
- **Two skill versions.** One pre-release report (July 2026) and three on v0.14.0 (August–October 2026); the Claude model version of the July run was inferred.
- **Developer as evaluator.** The tool was built and evaluated by the same author; severity was assigned by the agents against fixed anchors.
- **AI output can be wrong.** Every finding cites its test and location so a person can check it; the report is a review aid, not a replacement for expert judgement.

What a report looks like: a synthetic example

SYNTHETIC EXAMPLE. Every name, number, and finding below is invented. Full abridged example at **validateHE.org/sample-report**

1 Executive summary (non-technical)

Model: example cohort state-transition (Markov) model in R, four health states, Strategy A vs. Strategy B, monthly cycles.

Overall assessment: Fit only after revisions

The model runs, and most basic consistency checks pass. Two do not: time in one health state is counted twice in the QALYs of Strategy B, and costs stay discounted when discounting is switched off. Both change the cost-effectiveness results, so the model should be corrected and checked again before it informs a decision.

Validation method	Assessment
1 Transparency	Minor concerns: run order of the scripts unclear
2 Face validity	Not requested
3 Verification	Major concerns: two core checks failed
4 Cross-validation	Not requested
5 External validation	Not assessable: no observed data supplied

Table 6. The five validation methods at a glance (synthetic).

Grading

Severity. Critical: decision-relevant results are wrong, or a decision-relevant input silently does nothing, or the structure breaks (e.g., the cohort leaks). **Major:** base case correct but a material latent risk (hidden global variables, duplicated definitions that drift, error-masking). **Minor:** bounded defects with small, demonstrated impact; conventions that hinder verification. **Info:** observations; not counted.

Overall: fit for the stated decision problem · fit with minor revisions · fit only after revisions · not yet fit · not assessable on the reviewed evidence.

Per method: no material, minor, major, or critical concerns · not assessable. **Checklist items:** Pass · Partial · Fail · NA · Not-tested · Deferred.

"For clarification": a finding that depends on context the review could not see is phrased as a question to the authors.

2.3 Verification results (one finding)

BB-001 QALYs exceed life-years when all utilities are 1

Major · Fail · Module 3, black-box test BB-U1

Test: with every health-state utility set to 1, QALYs must equal life-years, discounted and undiscounted.

Inputs: all utilities set to 1 for both strategies; other inputs at base case.

Expected: QALYs equal life-years in both strategies.

Observed: Strategy A, 7.92 QALYs and 7.92 life-years. Strategy B, 8.71 QALYs and 8.37 life-years: QALYs 4.1% higher. The excess equals the time Strategy B spends in the Complicated state, which is added twice.

Checklist: CDA-AMC 38; TECH-VER TV-3.2-07

Location: R/outcomes.R:48

Recommended fix (not applied): count the QALYs accrued in the Complicated state once in the Strategy B total, then re-run the utilities = 1 test.

Figure 4. A finding as it appears in the technical body (synthetic).

Appendix A Recommended changes (not applied)

ID	File:line	Severity	Recommended change
BB-001	R/outcomes.R:48	Major	Count Complicated-state QALYs once in the Strategy B total.
BB-002	R/costs.R:23	Major	Read the cost discount rate from the model inputs.
TR-001	README.md	Minor	For clarification: which script produces the base case?

Table 7. The fix list, sorted by severity (synthetic).

Validate your model with us: free while in beta

We run it for you.
Share a current project to find bugs before reviewers do, or an older model with its documentation just to try validateHE.

You run it yourself.
We share the skill with all its materials (e.g., checklists) via GitHub or e-mail, and help you run it at your site, in person or remotely. Needs R, Quarto, and Claude or OpenAI Codex.

validateHE.org
sign up · sample report

You get a full report with evidence-cited findings and recommended fixes; in return, we ask you to fill in a short evaluation survey. R models now; Excel and other languages soon. Expanded model testing starts November 2026; we schedule models in the order requests arrive.

How your material is handled during the beta

We are still developing our processes for the full range of projects, from open-source academic models to confidential models of commercial organizations. During the beta, it works like this.

If we run it for you

- You send your model and documentation by a secure transfer method we agree with you. Please do not send personal or patient-level data.
- We are happy to sign your confidentiality agreement before you share anything; Benjamin P. Geisler signs it personally.
- validateHE runs on our computer in an AI coding tool, Claude Code (Anthropic) or Codex (OpenAI). The provider processes your code and documentation to produce the review; we do not allow it to use your material for model training. Before you send anything, we confirm the provider's terms (training, retention and data region) in writing.
- validateHE works on a temporary copy and never changes your model.
- We delete your model and documentation 90 days after we deliver the report, or earlier if you ask. We keep the report unless you object; on request we delete everything, including the report.

If you run it yourself

- We share the skill, free during the beta, and help you run it.
- We take no copies of your model, documentation, or report.
- validateHE runs in your own Claude or OpenAI Codex, so your own agreement with that provider applies.

Research use of the beta

- Evaluation survey answers may be reported in aggregated, anonymised form, for example in conference abstracts and papers.
- Nothing from your model, documentation, or report is published or shown to others without your written permission.
- With your permission, findings may be rewritten into de-identified example reports.

Frequently asked questions

Does validateHE change my model?
No. It works on a temporary copy, checks by before/after content hash that the original is unchanged, and never applies fixes; they are listed in the report as recommendations.

Is the face-validity council a panel of human experts?
No. It is AI-simulated, and the report says so. Checklist items that ask for human expert review stay capped unless your materials document one.

What if I have no comparator model or observed data?
Then cross-validation and external validation are reported as not assessable. Values are never filled in from general knowledge.

What do I get back?
A report in HTML and PDF: a plain-language executive summary, a technical body with the evidence behind each finding, and appendices with the recommended fixes, a function inventory and dependency map, and the council minutes.

Which models can it validate?
Models written in R: cohort Markov, decision tree, partitioned survival, discrete-event simulation, and microsimulation. Excel and other languages are planned.

Does it replace human review?
No. It is a review aid: every finding cites its test and file location so a person can check it, and it states what it could not test in that run, and why.

What does it cost?
Nothing during the beta. In return, we ask for a short evaluation survey.

Can I see an example first?
Yes. A fully synthetic example report is on validateHE.org.

Questions? Write to b.p.geisler@medisin.uio.no or request a spot at validateHE.org.

1 Eddy DM, et al. Model transparency and validation (ISPOR-SMDM TF-7). Value Health 2012. PMID 22999134
2 Radeva D, et al. Errors in NICE single technology appraisals. Int J Technol Assess Health Care 2020. PMID 32618536
3 Abraham K, et al. Reporting of model validation. PharmacoEconomics 2025. PMID 40293688
4 Li P, et al. Validation of cancer models. PharmacoEconomics 2026. PMID 42745154
5 Imre A, et al. LLMs in HE modelling: review. Expert Rev Pharmacoecon Outcomes Res 2026. PMID 42605534
6 Reason T, et al. AI to automate HE modelling. PharmacoEconomics Open 2024. PMID 38340276
7 Chhatwal J, et al. AI-driven ISPOR-SMDM validation of an Excel CEM. ISPOR Europe 2026, MSR129
8 Srivastava T, et al. Validation hackathon: LLM, human or agent? ISPOR Europe 2026, MSR181
9 Pandey S, et al.; Verhoek A, et al. AI verification of HE models. ISPOR Europe 2026
10 Fleurence RL, et al. ELEVATE-GenAI. Value Health 2025. PMID 40653157

AI disclosure (ELEVATE-GenAI [10]): validateHE runs in Claude (Anthropic) and Codex (OpenAI); reports shown were generated with Claude Opus 5 and 5.5, repeat runs with GPT-5.6 Terra and GPT-6.1 Sol in Codex. Poster and handout drafted with AI assistance, checked by the author.

Funding: University of Oslo operating funds (driftsmidler). **Conflicts of interest:** none.

validateHE