

How a CAI score is constructed

A CAI score is built from codebase evidence under published rules. This article follows that evidence through the scoring method: what enters the number, what stays outside it, and why weaker measured areas receive more influence.

Lasse Overgaard · Canine Development · 17 September 2026

CAI, the Codebase Assurance Index, constructs a score by applying published rules to evidence produced from a codebase.[10] The reference scorer works from that evidence; it does not inspect the repository itself. This division determines what a score can establish. Someone can recompute the number from the supplied evidence without thereby confirming that the measurements accurately describe the code.

Two decimal places can obscure the amount of work that precedes a result. Test coverage is 82.4 percent. Complexity is 14.2. A repository scores 69.35. That precision says nothing on its own about the scope or quality of the examination behind the figure. A codebase may contain thousands of files, several architectural styles, uneven test practices, dead code, and code whose purpose is no longer recorded. The measurement is recorded as structured evidence; CAI folds the eligible part of that evidence into one number. The score is a compression of what was recorded.

To understand that compression, a reader needs answers to six questions:

- What was allowed to influence it?
- What was deliberately kept out?
- What happens when evidence is partial or absent?
- How much can a strong area compensate for a weak one?
- What does the final word, such as **Adequate** or **Strong**, actually add?
- And can another person reconstruct the same result from the same evidence?

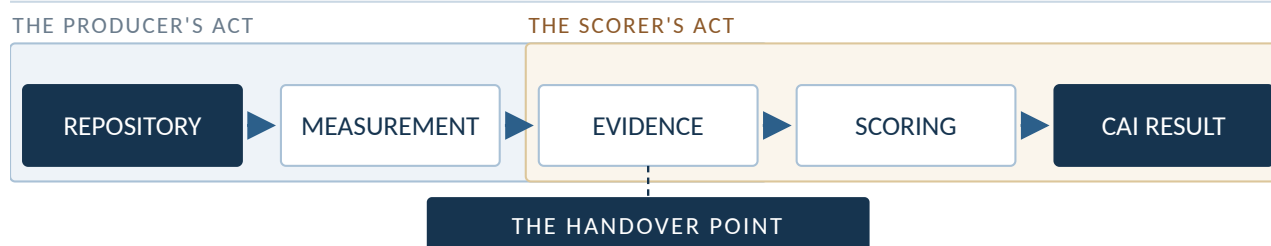
The scoring path answers these questions at different stages. Some concern the upstream evidence, while others concern the rules that reduce it.

1. The score begins before the scorer

CAI calls the party that measures a codebase the producer. The producer inspects the repository, decides what to examine, records its findings, and answers for the measurement if it is disputed. It supplies an evidence bundle, a structured record of those measurements rather than a prose report. The reference scorer reads the score-bearing parts of that bundle and applies the rules of a named rubric, the versioned document that defines how those parts combine; it does not open the repository or decide what to examine.[1][5][6]

The result therefore depends on two successive acts:

TWO ACTS, ONE RESULT



Judgement: what to examine, and what the findings are worth.

Arithmetic: the rules of a named rubric applied to the evidence supplied.

Figure 1. A score is the product of two acts. The producer judges what to examine and records what it found; the scorer applies the rules of a named rubric to that record. The evidence bundle is where one hands over to the other.

The first transformation includes decisions about what to examine and how to record the findings. The second applies arithmetic to the resulting evidence under the named rubric. Both occur before the reader sees a number; their boundary determines which part of the result can be checked by recomputing it.

The arithmetic half is open. Hand a recipient the evidence bundle and the rubric the result names, and they can run the fold themselves, meaning the calculation that reduces the evidence to the score, and check whether those inputs really do produce the published figure.[1][5][14]

What that leaves untouched is the producer's inspection of the repository. A calculator can confirm that $7 \times 8 = 56$. It cannot confirm that somebody counted seven of the right thing. Agreement on the arithmetic establishes that the number follows from the evidence supplied with it, and establishes nothing about whether that evidence describes the code.

Keeping the arithmetic open makes that boundary inspectable. The evidence bundle is the handover point. The producer claims:

This is what we measured.

The scored result makes a second claim:

Given that evidence, this is the score the CAI rules produce.

The companion explainer on recomputation examines the trust boundary in depth, alongside the paper it draws on.[14] The scoring path examined here begins with the evidence bundle at that boundary.

2. The first challenge is deciding what gets to survive compression

A score leaves out most of the material in an analysis. Preserving every finding, file, detector result and intermediate observation would reproduce the analysis instead of reducing it. The scoring rules must therefore make clear which information participates in the reduction, and a reader must be able to inspect what remains outside the number.

CAI's raw material is a set of deterministic dimensions, each one an individual check whose numeric result follows from fixed rules rather than from a judgement call. Beside them sits a smaller set of direct meta-dimensions. All of it is

organised under a versioned rubric, whose catalogue lists those checks and records which category each one belongs to.

Each dimension carries context about how its result was produced. Its raw score is only one input to the next stage; two accompanying fields, **coverage** and **confidence**, affect how the scorer uses it. They answer separate questions and operate at different points in the fold.

Coverage asks: how much of the relevant surface did we actually measure?

Imagine a dimension receives a raw score of 8 out of 10. If the producer covered the full relevant surface, the scorer uses the number as it stands:

$$8 \times 1.0 = 8$$

If only half the relevant surface was covered, the number is scaled to match what was actually examined:

$$8 \times 0.5 = 4$$

The rule is simply:

$$\text{effective} = \text{score} \times \text{coverage}$$

Coverage therefore changes the effective value entering the next stage; it is not merely a caveat beside an unchanged score.[5][6] An 8 based on half the target is not treated as though the other half politely agreed.

WHAT COVERAGE DOES TO A SCORE

Full relevant surface measured



Half the relevant surface measured



Figure 2. Coverage changes the value that enters the next stage. A dimension measured across half its relevant surface contributes half its raw score, rather than carrying a full score with a caveat attached.

Confidence weights a dimension within its category

Confidence does not touch the value. It affects how strongly a measured dimension contributes when related dimensions are combined inside a category, meaning the group of related checks it belongs to. Categories in turn feed a lens, which is one of the broad assurance perspectives a CAI result is reported under.[1][5][6][7]

They act at separate points:

- coverage changes the effective value;
- confidence changes the weight of that value inside its category.

Both reduce a dimension's influence. One says we saw less of it than we meant to; the other says we are less sure of what we saw. A reading that treats them as one field supports conclusions the evidence does not carry.

3. The catalogue does not get to vote by row count

Two areas of concern sit inside the same lens. One needs ten dimensions to describe properly. The other is adequately covered by two.

If every dimension went straight into the lens as an equal independent input, the first subject would get five times as many chances to influence the result. Its extra influence would come from needing more rows, not from a judgement that it matters five times as much. That is an odd way to assign importance, and the catalogue is where it enters.

CAI therefore groups deterministic dimensions into **categories** before those categories enter a lens.[1][3][5] Related dimensions are combined first, and the category then participates in the lens fold as one structured input instead of as ten separate votes. Catalogue detail should not quietly become voting power.

Category grouping gives the catalogue structure a direct effect on the score. Move a dimension into another category and you change which other dimensions it combines with, potentially changing the result.[3][5][9] The category map is score-moving method data, not just a way to organise the catalogue. Indicator structure sits alongside weighting, aggregation and missing-data treatment in the general composite-indicator literature as a consequential design choice. [13]

Where the published catalogue fixes the category map, a reclassification capable of moving a score belongs in a new rubric version rather than an editorial tidy-up.[9] The catalogue both describes what is measured and shapes how those measurements are reduced. Changes to its score-moving structure therefore require the same scrutiny as changes to a formula.

Meta-dimensions take a shorter route

Not every score-bearing input takes that path. CAI also supports measured meta-dimensions, which feed a lens directly without passing through a category.[5][6] A meta-dimension scored at 7.2 enters its lens as 72 on the 0 to 100 lens scale.

The direct route matters: category structure affects aggregation, while a meta-dimension bypasses it. The rubric must preserve that distinction for a recipient to replay the calculation.[3]

4. The worst-first weighting rule

CAI combines the category scores and direct meta-dimensions within each lens using a weighted rule. It does **not** use an ordinary mean.

The current scorer uses a worst-first ordered weighted average, or OWA.[5] The rule underneath it is short. Take the numeric inputs to a lens, order them from weakest to strongest, give the weakest the largest weight, and hand progressively smaller weights to the stronger ones. Strong areas still count. They do not get to drown out a weak one.

OWA itself is an established aggregation family, introduced by Ronald Yager and developed through a substantial later literature.[11][12] The family is external and well studied. The parameter choices CAI makes within it are CAI's own, and the literature validates the operator family, not those choices.

An ordinary average is thoroughly democratic: every number gets an equal vote, and a serious weakness can be outvoted by a quorum of healthy areas that have nothing to do with it. CAI is deliberately less democratic, because the method is built to preserve the signal of weakness where an average would let it disappear beneath a pile of respectable scores.[3][5][8]

This rule is a methodological choice among possible ways to aggregate the inputs.[3] The current within-lens OWA uses a default decay parameter of:

$$q = 0.75$$

For the across-lens headline fold, the default is:

$$q = 0.55$$

Both are versioned scoring parameters resolved by the rubric version, not settled by whichever scorer build happened to run.[5][8][9]

5. How three lens scores become 69.35

Not every lens appears in every result. The catalogue distinguishes core lenses from conditional ones, which apply only where the architecture calls for them: a repository with no web front end produces no accessibility result at all instead of a bad one.[7] Even a lens designated as core may fail to resolve to a number, and only lenses with numeric results enter the fold.[5]

Suppose a simplified CAI result contains three numeric lens scores:

LENS	SCORE
Code Health	60
Readiness	75
Accessibility	90

A normal arithmetic mean would give:

$$(60 + 75 + 90) / 3 = 75$$

The CAI headline fold lands elsewhere, because with the across-lens OWA parameter $q = 0.55$ the weakest score receives the greatest weight:

ORDERED SCORE	OWA WEIGHT
60	0.5398
75	0.2969
90	0.1633

These weights produce the headline:

$$\begin{aligned}
 &60 \times 0.5398 \\
 &+ 75 \times 0.2969 \\
 &+ 90 \times 0.1633 \\
 &= 69.35
 \end{aligned}$$

The ordinary mean gives **75**; the current CAI fold gives **69.35** from the same three scores.[5] Each number is a constructed summary. The difference is in what the rules retain: equal influence for every lens in the mean, and greater influence for the weakest measured lens in the CAI fold.

THE SAME SCORES UNDER TWO RULES

Ordinary mean · every lens an equal vote



Worst-first fold · $q = 0.55$ · the weakest carries most



Figure 3. The same three lens scores under two aggregation rules. The mean gives every lens an equal vote. The worst-first fold gives the weakest measured lens more than half the weight, and lands four and a half points lower.

The headline is another compression layer

Each lens has already compressed its categories and direct meta-dimensions once. The headline then compresses the measured lenses again, so the full score is not one average taken over one giant bag of findings. It is a staged reduction:

HOW THE FOLD REDUCES IN STAGES

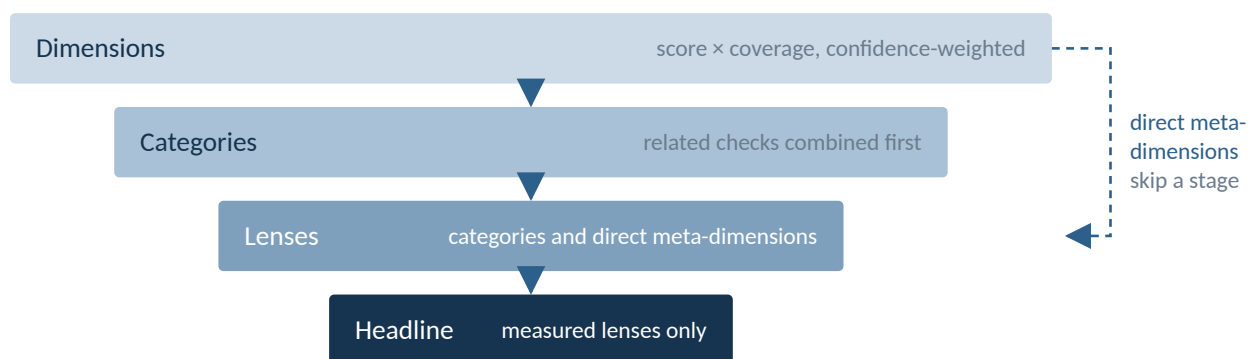


Figure 4. Each stage compresses what the stage below it produced. Direct meta-dimensions take the shorter route, entering a lens without passing through a category.

Because the reduction happens in stages, two repositories can arrive at the same headline by different routes. The number cannot tell you which route it took. The decomposition can: the lens scores the headline was folded from, and the categories beneath them.

6. Missing evidence is where simple arithmetic becomes dangerous

Now take the Accessibility lens out of the example. Not reduced to zero: removed, with no numeric lens result at all. The remaining measured set is:

60, 75

The scorer does not insert a synthetic zero where Accessibility used to be.[2][5] It recalculates the OWA weights over the survivors, so the new weights are:

ORDERED SCORE	OWA WEIGHT
60	0.6452
75	0.3548

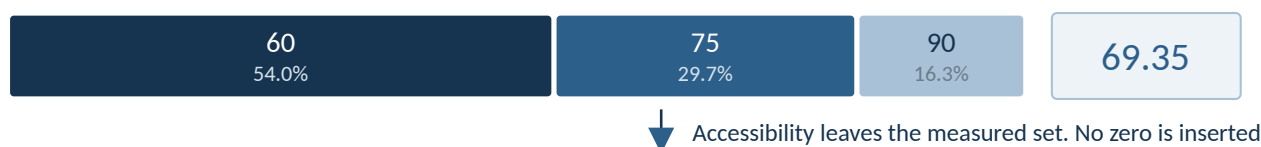
The new headline is:

$$\begin{aligned} &60 \times 0.6452 \\ &+ 75 \times 0.3548 \\ &= 65.32 \end{aligned}$$

Nothing was scored as zero, and the headline still moved by four points. What changed was the set, not the values: the missing 90 no longer participates, so the two survivors absorb the full weight between them.

WHAT HAPPENS WHEN A LENS IS ABSENT

Three measured lenses



Two measured lenses - the weights renormalise over the survivors



Figure 5. Removing the strongest lens from the measured set moves the headline without any value being scored as zero. The weights renormalise over the two survivors, which then absorb the full weight between them.

“Missing does not count as zero” and “missing does not affect the result” are **not** the same statement.[2][5] The first describes the scorer's treatment of absent evidence. The second is contradicted by the change from 69.35 to 65.32; removing an input changes the set over which the weights are calculated.

Zero and absence tell different stories

A zero means we measured this and the result was zero. Absence means there is no numeric score-bearing result here at all. Those are different claims about the world, and collapsing them into one value discards the difference. The scorer therefore works with the measured set instead of manufacturing failure values for the items it never saw.[2][5][6]

What the numeric fold cannot do is tell you why something is absent. The lens may never have applied, the producer may not have resolved it, or the measurement scope may have been thin. Those are upstream evidence questions and the fold holds no opinion about which one occurred.[2]

Partial coverage is a third case

A partially covered dimension is **not absent**. It is measured evidence with a reduced effective value, and it participates all the way up: it can lower a category, influence a lens, and under the current rules reach the critical gate. That gate is the rule under which a measured contributor whose effective value falls below the rubric's threshold constrains how its lens is labelled, while leaving the lens's numeric score alone.[2][5][6][8]

Absence, a measured zero and partial coverage therefore require separate descriptions. An absent item contributes no numeric input; a measured zero contributes a numeric failure; and a partially covered result participates with a reduced effective value. Treating them as interchangeable would make the score harder to interpret and could change the arithmetic itself.

THREE STATES THAT ARE NOT THE SAME

Absent There is no numeric score-bearing result at all. Contributes no input. The weights renormalise over what remains.	Partial Measured, with a reduced effective value. Participates all the way up, and can reach the critical gate.	A measured zero We looked, and the result was zero. Contributes a numeric failure to the fold.
---	--	---

Figure 6. Three states that look alike in a report and behave differently in the fold. Absence removes an input, partial coverage reduces one, and a measured zero contributes a failure.

7. A few things are deliberately kept out of the number

CAI keeps advisory material outside the deterministic score. Under this **advisory firewall**, a language-model judgement or another advisory observation may travel with a result to explain or contextualise it, but it cannot move the number.[5][6]

So an advisory observation is free to say:

this looks suspicious.

It cannot go on to add:

therefore minus four points.

The separation lets a reader challenge an advisory interpretation on its merits. The reader can inspect the numeric fold independently, without having to determine whether the disputed judgement changed the headline by an undisclosed amount.

The same principle covers the other descriptive fields carried in the evidence model. One of them is producer-reported survey completeness, carried as `surveyFit`, which records how many applicable survey lenses the producer identified against how many actually resolved. A thin reading and a broad reading can land on the same headline while resting on very unequal evidence, so the field travels with the result and stays outside the numeric fold.[2][6] Staying outside the fold is not a demotion. Some information is too important to disguise as a score.

Architecture gets one extra sanity rule

The Architecture lens also has a safeguard for thin measurement surfaces. A repository with almost no analyzable architecture can otherwise look impressive on architecture measures that were given little opportunity to fail. The scorer limits what such a result can contribute.

The current parameters therefore allow Architecture to be dropped altogether when there is no analyzable project surface, and to be capped when the analyzable surface is very thin.[5][6][8] The current defaults are 2 analyzable projects, 1,500 production lines of code, and a low-surface cap of 69.[8]

These thresholds are versioned scoring choices, not empirically settled constants about architecture. The words printed beside the number also depend on versioned choices, through the band rules.

8. The number is not the word

Our synthetic score of **69.35** is a number. Under the current baseline band cutlines it is also **Adequate**. A band is the word CAI prints beside the number, and a cutline is the boundary between one band and the next:[4][8]

BAND	BASELINE RANGE
Critical	below 25
Weak	25 to below 50
Adequate	50 to below 70
Strong	70 to below 90
Exemplary	90 to 100

THE BASELINE BAND CUTLINES

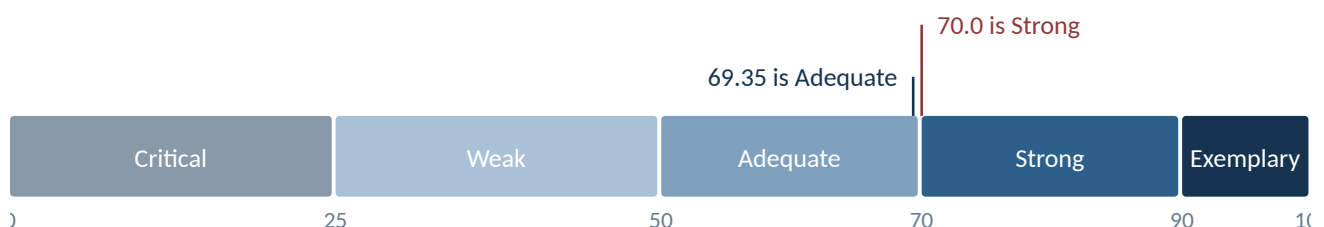


Figure 7. The baseline band cutlines. The number and the word answer different questions: the first says where the arithmetic landed, the second says how this rubric classifies it.

Those are the reader-facing names. Anyone checking a result against the scorer or a rubric catalogue will meet the implementation's own tokens for the middle three bands, `poor`, `fair` and `healthy`, which map to Weak, Adequate and Strong at the same cutlines.[8]

Move the score from 69.9 to 70.0. The numeric movement is a tenth of a point and the label changes from Adequate to Strong. Software did not cross a geological fault line at 70. The rubric crossed a cutline.

The arithmetic fold produces the number; versioned cutlines determine the band assigned to it.[3][8] The scorer carries other interpretive rules with the same separation. A sufficiently weak measured contributor can cap a lens band without changing the lens's numeric score at all. Quality-bar rules can shift all four of a lens's cutlines together, according to the quality bar declared in the evidence bundle, so the same score reads differently for a prototype and for a system in production. When the bundle declares nothing, the production baseline applies. The headline band can be coherence-capped, so that it never sits more than one band above the weakest measured category. All three adjust the label and leave the arithmetic alone.[5][8]

So when a report prints:

69.35
Adequate

Those two lines answer different questions. The first says where the arithmetic landed. The second says how this rubric classifies that result.[3]

9. A score without its rubric version has lost part of its identity

A result presented as a single sentence leaves out its scoring rules:

This repository scored 72.

To interpret or recompute that 72, the reader needs to ask:

Under which rubric?

A rubric catalogue can carry a `scoring` block holding the score-moving parameters: the OWA decay values, the critical gate, the architecture-surface thresholds, the band cutlines and the quality-bar rules. A catalogue published without that block resolves to the scorer's frozen defaults, and those defaults are the values this article quotes. Either route is fixed by the version, because any change capable of moving a score for unchanged evidence mints a new rubric version, and older versions stay available so a historical result can still be replayed under the rule that actually produced it.[8][9]

So `rubricVersion` selects the score-moving semantics whether they were written into the catalogue or inherited from the frozen default, and it does so for the score-moving catalogue structure as well as for the parameters.[1][5][9] Change any of that and unchanged evidence can produce another result, or the same result wearing a different word. The version is therefore part of the score's identity and not a filing detail attached to it.

The resulting contract is testable. Here, *eligible evidence* means the measured, non-advisory material the fold is allowed to use:

same eligible evidence + same rubric version = same score

It does not establish the different claim that:

this repository has one eternal true number.

That would be a rather ambitious thing to promise about software. A scorer build can still be worth recording as provenance, but it is not a second authority competing to decide the number. The named rubric is the scoring-semantics identity used to replay the fold.[1][3][8][9]

10. What recomputation gives you, and what it does not

A constructed result can be checked against the inputs from which it was calculated. Given the evidence and the named rubric, a recipient can ask whether that evidence produces the reported score. If it does not, there is an arithmetic, implementation or artifact problem to investigate. If it does, the fold has been recomputed successfully; that agreement still does not verify the producer's measurement of the repository.[1][5][14]

Recomputation exposes arithmetic errors; the evidence bundle lets readers inspect which observations entered the fold; and the named rubric identifies the rules applied to them. Whether those observations accurately describe the repository remains the producer's responsibility.

11. The complete scoring path

The stages described above can be read together in one diagram:

THE SCORING PATH FROM END TO END

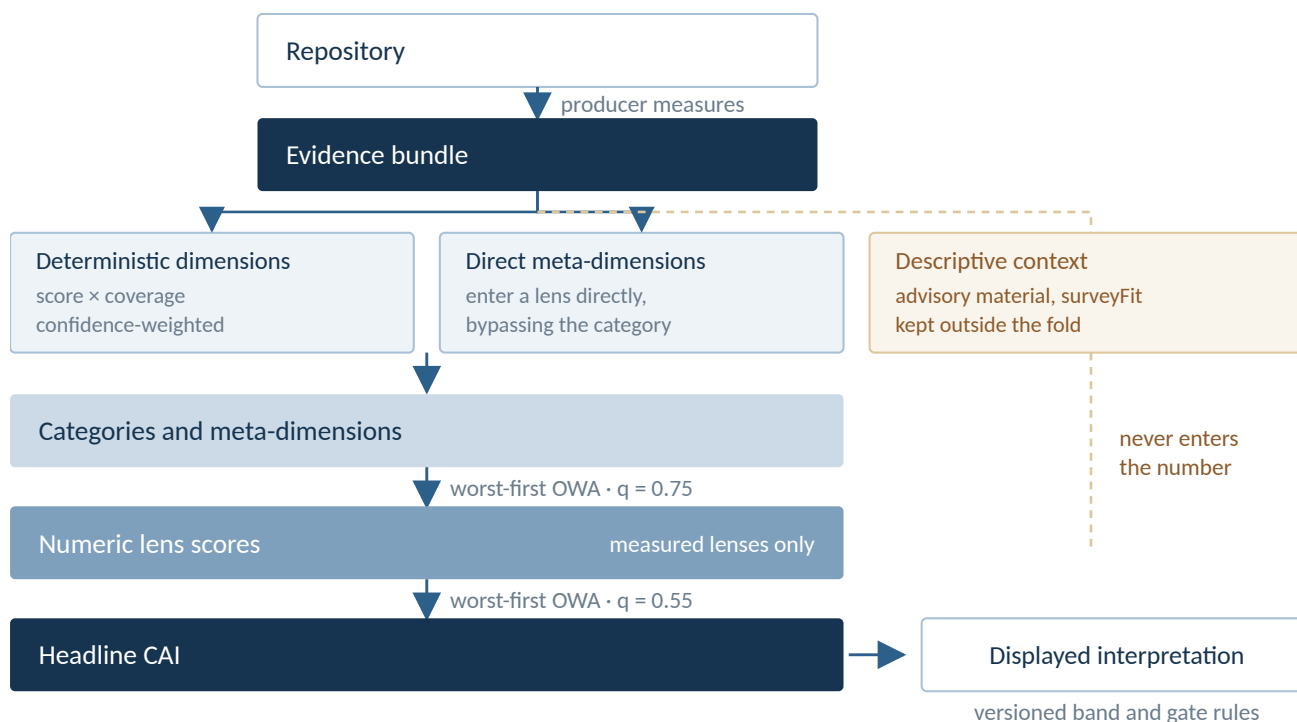


Figure 8. The whole path in one view. Descriptive and advisory material travels with the result and stays outside the numeric fold at every stage.

The path begins with the producer's measurement and ends with a displayed interpretation under a named rubric. The headline needs to travel with that rubric version and its decomposition: the measured lenses and the categories and dimensions beneath them. Without those accompanying parts, a reader cannot trace the number back through the scoring stages, regardless of how many decimal places it carries.

12. The construction in brief

The method can be stated in one paragraph:

A producer measures a codebase and emits structured evidence under a named CAI rubric. CAI reduces eligible deterministic dimensions through categories, combines those categories with direct meta-dimensions into lens scores, gives weaker measured inputs more influence through a worst-first ordered weighted average, and then folds the measured lenses into one headline. Missing numeric items are not replaced with synthetic zeroes, advisory material cannot move the deterministic number, and the rubric version identifies the score-moving rules needed to replay the result. The final band is an interpretation of the number under that same versioned method.[1][2][3][5][6][8][9]

Its shortest formulation is:

A CAI score is a versioned compression of codebase evidence, not a number hiding in the source code waiting to be found.

The compression is inspectable when the evidence and decomposition accompany the result: a recipient can check what participated, what received more influence and what stayed outside the number. The rubric version identifies the rule used for the calculation; the producer remains responsible for the measurement. A good score does not eliminate the need to ask questions. It makes the questions better.

Related reading

- *Recomputable scoring is not the same as reproduced measurement*, and the paper it draws on, *From Score to Evidence: Recomputation, Provenance, and Measurement Boundaries*, for the trust boundary between upstream measurement and deterministic scoring.[14]
 - *Applicability is part of measurement*, and the paper *Applicability, Coverage, and Missing Evidence in Codebase Assurance Scoring*, for why an item is absent and what completeness context can and cannot tell you.[2]
 - *A CAI score without its rubric version is incomplete*, for version identity across time and across comparisons.
 - *Normative Choices in a Technical Scoring Standard*, for weighting, cutlines and the governance of score-moving design decisions.[3]
 - *How the Codebase Assurance Index is constructed*, for the full technical account this explainer compresses.[1]
-

References

CAI implementation sources are cited at commit [1d557c36faa6f7650cfe6746db01665ad2ac9bfc](#). Nothing under [src/Cai.Scoring](#) has changed between that commit and [a24b47b2](#) of 16 September 2026, so those links still show the current implementation. External sources establish general methodology only and do not validate CAI's specific scoring choices.

1. **Jimmy Borch / Canine Development**, *How the Codebase Assurance Index is constructed*. Founding technical account of the current evidence-to-score path, category arithmetic, OWA, omission semantics, banding and rubric-version identity.
2. **Jimmy Borch / Canine Development**, *Applicability, Coverage, and Missing Evidence in Codebase Assurance Scoring*. Founding technical treatment of omission, confidence, partial coverage, measured-lens participation and non-scored survey completeness. CAI-specific implementation claims in this article are checked against the reference scorer and evidence model directly.
3. **Jimmy Borch / Canine Development**, *Normative Choices in a Technical Scoring Standard*. Founding paper on weighting, aggregation, cutlines, versioned scoring parameters and the distinction between implementation facts and stronger empirical claims.
4. **Codebase Assurance Index**, *How the CAI is computed*. Current public reader-facing specification surface, re-read 14 September 2026. <https://codeassuranceindex.info/spec/>
5. **CanineCC/CAI**, [src/Cai.Scoring/CaiScorer.cs](#). Current reference implementation of category aggregation, meta-dimension participation, lens OWA, headline OWA, missing-lens behaviour, critical gating, band coherence and verification. <https://github.com/CanineCC/CAI/blob/1d557c36faa6f7650cfe6746db01665ad2ac9bfc/src/Cai.Scoring/CaiScorer.cs>
6. **CanineCC/CAI**, [src/Cai.Scoring/EvidenceBundle.cs](#). Current evidence fields, coverage, confidence, advisory status, thin-lens fallback and descriptive non-scored `surveyFit`. <https://github.com/CanineCC/CAI/blob/1d557c36faa6f7650cfe6746db01665ad2ac9bfc/src/Cai.Scoring/EvidenceBundle.cs>
7. **CanineCC/CAI**, [src/Cai.Scoring/Lenses.cs](#). Current ten-lens catalogue and core/model-aware classification. <https://github.com/CanineCC/CAI/blob/1d557c36faa6f7650cfe6746db01665ad2ac9bfc/src/Cai.Scoring/Lenses.cs>
8. **CanineCC/CAI**, [src/Cai.Scoring/ScoringParameters.cs](#). Current rubric-carried within/across OWA parameters, critical gate, architecture-surface parameters, baseline band cutlines, band labels and quality-bar rules. <https://github.com/CanineCC/CAI/blob/1d557c36faa6f7650cfe6746db01665ad2ac9bfc/src/Cai.Scoring/ScoringParameters.cs>
9. **CanineCC/CAI**, *ADR-0004: Versioned, frozen rubric catalogs*. Current versioning rule: score-moving changes mint a new rubric version; a catalog carries a `scoring` block and one published without it resolves to `ScoringParameters.Default`; score-moving catalogue structure and fold parameters are pinned to the versioned rubric. <https://github.com/CanineCC/CAI/blob/1d557c36faa6f7650cfe6746db01665ad2ac9bfc/docs/adr/0004-versioned-frozen-rubrics.md>
10. **CanineCC/CAI**, *README*. Public repository overview of the CAI standard, rubric versioning, reference scorer and verification surfaces. <https://github.com/CanineCC/CAI/blob/1d557c36faa6f7650cfe6746db01665ad2ac9bfc/README.md>
11. **Ronald R. Yager**, “On ordered weighted averaging aggregation operators in multicriteria decisionmaking”, *IEEE Transactions on Systems, Man, and Cybernetics* 18(1), 1988, pp. 183-190. DOI [10.1109/21.87068](#). Used only to establish the OWA operator family.
12. **Ali Emrouznejad and Marianna Marra**, “Ordered Weighted Averaging Operators 1988-2014: A Citation-Based Literature Survey”, *International Journal of Intelligent Systems* 29(11), 2014, pp. 994-1014. DOI [10.1002/int.2167](#)

3 . Used to place OWA in its broader research family, not to validate CAI's parameterisation.

13. **OECD and European Commission Joint Research Centre**, *Handbook on Constructing Composite Indicators: Methodology and User Guide*, OECD Publishing, 2008. DOI [10.1787/9789264043466-en](https://doi.org/10.1787/9789264043466-en) . Used for the general methodological proposition that weighting, aggregation, indicator structure and missing-data treatment are consequential composite-indicator design choices. It does not validate CAI's specific choices.
 14. **Jimmy Borch / Canine Development**, *From Score to Evidence: Recomputation, Provenance, and Measurement Boundaries*. Founding technical treatment of the boundary between upstream measurement, deterministic aggregation, artifact provenance and verification.
-
-

STATE OF THIS ARTICLE

This article describes the CAI implementation and specification as they stood on 17 September 2026. Those sources move. The two decay parameters, the architecture thresholds and the band cutlines quoted here are the values in force at that date, not constants of the method, and a later rubric version can change any of them.

That is why a score carries its rubric version. This article explains how the construction works. The rubric version printed beside a particular result is what fixes the rules that produced it, and where the two ever differ, the rubric that scored the result is the one that counts.

Informative and non-normative. Where this article and the published CAI specification or the applicable rubric version disagree, the specification and the rubric control.