VENIO

STRATEGIC GUIDE 2026

The AI Review Field Guide

Criteria, validation, and the record

How to write the instructions an AI classifier works from, how to test whether it applied them, and what to keep.

01 Start here

This guide covers three things: writing the criteria an AI classifier works from, testing whether it applied them correctly, and keeping a record that explains the review to someone who was not there.

One claim runs underneath all of it. When AI-assisted review disappoints, the instructions are almost always the reason, not the model.

That is worth stating plainly because it contradicts how most pilots are diagnosed. A team runs a first pass, reads inconsistent output, and concludes the technology is not ready. In most of those cases, the classifier applied a vague instruction consistently, which is what it is built to do.

Ask three experienced reviewers on the same matter to write down what makes a document responsive and you will get three answers that disagree at the edges. Linear review absorbs that. A reviewer who is unsure asks the person next to them, and the disagreement disperses into unrecorded judgments. A classifier cannot ask. It applies what it was given to every document the same way, which makes the ambiguity visible rather than creating it.

So the skill has moved. It is no longer in operating the platform, which is now genuinely easy. It is in specification: writing down what you are looking for precisely enough that someone who has never seen the matter could apply it, then checking whether they did.

Section 3 is that skill. Section 5 is the check. Everything else supports one or the other.

What this assumes

That a qualified attorney still makes every determination carrying legal weight, and the classifier produces a first pass. That you have access to an AI-assisted review capability or are evaluating one. And that at some point someone outside your team will ask how the review was conducted, and you would rather answer with a method than a reassurance.

02 Where AI review pays

Most teams associate technology-assisted review with one matter: the big one. But across a year, review hours are spread over a long tail of mid-sized matters nobody considered automating.

That tail exists because traditional technology-assisted review carried a setup cost. It needed a seed set, a subject matter expert to train it, and someone to run the workflow. Below some volume the setup outweighed the savings, so teams put reviewers on it and worked through the set. The threshold was a function of setup cost, not of whether the matter would have benefited.

Criteria-based classification removes that setup. In Venio Review, no additional indexing or model training is required for a new document set; the same written instructions are applied against it directly. There is no seed set to build and no expert to borrow.

The consequence is not that AI review improved on large matters. It is that it became available on matters that never qualified.

Matter profile	Previously	With criteria-based classification
Large litigation or investigation	TAR justified, setup amortized across volume	Unchanged, with a faster start
Mid-sized commercial dispute	Usually linear, setup rarely justified	Viable. Criteria written once, applied immediately
Internal investigation, narrow scope	Linear and usually urgent	Viable, and speed matters more here than savings
Public records or FOIA response	Linear, repetitive, deadline-driven	Viable, and criteria are reusable across requests
Portfolio of small recurring matters	Never considered, each too small alone	The strongest case. Criteria reusable across the portfolio

The last row is the one worth sitting with. A team running thirty small matters a year could never justify a per-matter setup cost, but the aggregate hours across those thirty frequently exceed the single large matter that gets all the attention.

03 Writing criteria that survive

You are not writing a prompt. You are writing the instruction sheet you would hand to a competent reviewer joining the matter tomorrow, who has attended none of the calls and cannot ask you a question.

Seven rules, each with a weak and a strong version.

1. Define the tag, not the goal

First drafts usually describe what the legal team is trying to establish. The classifier sees one document at a time and has no access to the theory of the case.

WEAK

Documents relevant to the company's knowledge of the defect.

STRONG

Any document in which an employee describes, reports, forwards, or responds to a report of cracking, separation, or failure in the Series 400 housing. Includes internal email, test reports, warranty claims, supplier correspondence, and meeting notes. Qualifies whether or not the writer accepts that a defect exists.

2. State the exclusions

What is not responsive is where classification drifts, because absent an instruction the model applies a general reading and pulls in adjacent material. Exclusions are obvious to the team and invisible on the page.

WEAK

Communications about the acquisition.

STRONG

Communications about the acquisition of Northbrook Industries between 1 January 2023 and closing. Not responsive: deal-administration logistics with no substantive content, including scheduling, dial-in details, transfer confirmations, and signature routing. Not responsive: other acquisitions unless Northbrook is also discussed in the same document.

3. Name the actors, systems, products and dates

Criteria written in general terms produce general results. Every proper noun narrows the reading.

WEAK

Communications involving senior management about pricing.

STRONG

Communications to, from, or copying any of [named list of 11 individuals with titles]. Subject matter: list price, discount approval, floor pricing, or rebate structure for the Meridian product line. Date range: 1 March 2022 to 30 September 2023 inclusive. Internal project names in use: "Lantern", "LP2".

4. Do not use terms of art as instructions

Material, reasonable, significant, substantial, appropriate. These carry precise meaning inside a legal argument and almost none inside an instruction. Replace each with an operational definition. If you cannot write one, the term is doing work that has not been decided yet, and no amount of tuning will resolve it.

WEAK

Documents showing a material change in the financial position.

STRONG

Documents stating, forecasting, or discussing a change of 10 percent or more in quarterly revenue, EBITDA, or cash position, or any statement that a covenant has been or may be breached.

5. One concept per tag

Coding manuals combine concepts because a human can hold two ideas at once. A tag covering both a transaction and the communications about it will produce inconsistent calls indefinitely, and no validation round fixes it, because the inconsistency is in the specification. Split compound tags and combine the results afterwards.

6. Write for someone who has not seen the matter

Read the finished criteria as though you know nothing about the case. Every time you supply context from your own knowledge to make a sentence work, that context is missing from the page. Better still, hand it to a colleague not on the matter and ask them to code fifteen documents. Where they hesitate is where the classifier will produce noise.

7. Draft against edge cases you already know

Every matter has documents the team has already argued about. The near-miss. The one responsive only because of the attachment. The forwarded chain where only the bottom message qualifies. Pull five to ten before writing and check the draft against each. If the criteria do not clearly produce the answer the team agreed on, they are not finished. These also become the start of the control set in Section 5.

Assembling the specification

A complete specification for one tag contains the tag name, a one-line definition, the inclusion statement, explicit exclusions, named entities and date ranges, document types in scope, the treatment of families and attachments, and a version number with date and approver. Appendix A is the template.

Decide the family and attachment rule explicitly. Teams forget it, then find mid-review that attachments are being coded independently while the coding manual assumed family-level calls.

04 Privilege, and the log as a byproduct

Privilege review runs last, under deadline, and produces an artifact assembled by hand from whatever was recorded along the way. That sequencing is the problem. Entries get written by people reconstructing decisions made weeks earlier, which is the condition under which descriptions become inconsistent and thin.

Writing privilege criteria

The seven rules apply, with three additions.

Supply the attorney list. Names, and where possible email domains, for in-house counsel, outside counsel, and anyone acting in a legal capacity whose title does not say so. Without it the classifier infers who the lawyers are from context.

Distinguish legal capacity from role. In-house counsel send a great deal of email that is not legal advice. Say so explicitly, and describe what business-capacity communication looks like on this matter. That is where over-inclusion comes from.

Separate the claims. Attorney-client communication and work product are different claims with different requirements. One combined tag produces entries that assert both and establish neither. Write them separately and let a document carry both where it genuinely does.

What the classifier returns

In Venio Review, privilege output arrives with the work already recorded. Attorneys are identified in a separate field, which makes the results straightforward to validate against your own list. The elements of privilege are specified per document, so the strength of each call is visible rather than assumed. And a summary of the document with the reasons for the privilege is written into a log entry field that can be used or edited as needed.

The significance is sequencing. What a log entry needs is captured at the point of classification rather than reconstructed afterwards, so the log accumulates during review instead of being built at the end of it.

That does not remove the attorney review. A drafted entry is a draft. It removes the blank page, and the time usually spent in front of it.

The one check worth running

If you do a single quality check on privilege output, check the named attorneys against your own list. It is fast, objective, and catches both failure modes: a non-attorney treated as counsel, which inflates the log, and an attorney not recognized as one, which is how privileged material reaches a production. Appendix C is the full pass.

05 Validation

You validate an AI-assisted review for the same reason you sample a production before it goes out the door. Not because you expect a problem, but because "we checked, here is what we found" is a materially different position than "we were confident."

This section covers how to run a validation round. It does not set the methodology for your matter, and you should be wary of any guide that claims to.

5.1 Set the methodology with someone qualified

Sample sizes, confidence levels, and what an acceptable result looks like are matter-specific decisions. They depend on the collection, on how rare the responsive material is, on what is at stake, and on what your obligations actually require.

Set them with someone qualified to set them, before the first round runs, and write down what you agreed. A target chosen after seeing a disappointing result is not a target, it is a negotiation. Everything below assumes that conversation has happened.

5.2 What you are measuring

Recall. Of all the responsive documents in the collection, what share the process found. This is the defensibility number. A production is challenged on what is missing.

Precision. Of everything coded responsive, what share actually is. This is the cost number. Poor precision sends reviewers to documents that were never going to matter.

Elusion. Of everything set aside, what share is actually responsive. Recall cannot be measured directly without reviewing the whole collection, so what you can measure is how much responsive material is sitting in the pile nobody plans to open.

Precision is the one teams instinctively check, because the responsive set is right there. It is the least important of the three. Nobody will challenge you for having been too inclusive.

5.3 Build the control set before you look

Take a random sample from the collection and have qualified reviewers code it against the same written criteria the classifier received, before anyone sees the classifier's calls.

The order matters. A reviewer shown a classification and asked whether they agree will agree more often than one coding the document cold. That is not a comment on diligence, it is how review works under time pressure. Code the control set afterwards and you have measured how persuasive the output looks, not whether it is right.

Random means random. Not the custodians you care about, not the documents surfaced by search terms, not a convenient batch. Any selection logic becomes a bias in every number that follows.

Record who coded it, when, and against which criteria version. A control set coded against version one cannot validate version three.

5.4 Sample the pile you discarded

The instinct is to check the responsive set, because it is in front of you and it is what the review produced. It tells you the least. What you have not looked at is where the exposure is.

So a validation round samples the set-aside pile, coded cold, by reviewers working from the same criteria. That is the measurement that speaks to completeness.

One warning about rare tags. Where responsive material is a very small share of the collection, a sample drawn at a size that felt adequate will contain only a handful of responsive documents, and conclusions drawn from a handful move sharply if one or two are adjudicated differently. This is the most common way a validation round produces a confident number that means nothing. It is also exactly the situation where the sizing conversation in 5.1 earns its keep.

5.5 When the classifier and a reviewer disagree

Adjudicate by reading the document, not by assuming the human was right. A disagreement between a contract reviewer at hour six and a classifier applying written criteria is genuinely ambiguous evidence, and treating the human call as ground truth by default means every round reports criteria problems as classifier error.

Sort adjudicated disagreements into three buckets.

- **Ambiguous criteria.** Both readings defensible. Section 3 fixes it.
- **Reviewer error.** Fatigue, a missed attachment, a misread date. Note it and move on.
- **Classifier error on unambiguous language.** The only bucket that is a model problem, and usually the smallest.

Keep the counts. Over three rounds they tell you whether you are improving the specification or chasing noise.

5.6 When the results come back badly

Work the diagnosis in order. The first four are far more common than the fifth.

1. **The tag is doing two jobs.** Split it and re-run.

2. **An exclusion was never stated.** The misses cluster around a document type, custodian, or date range the criteria never mentioned.

3. **A term of art was read loosely.** Replace with an operational definition.

4. **A document type was never contemplated.** Spreadsheets, chat transcripts, scanned exhibits with poor text. Check whether the misses share a file type first.

5. **The classifier cannot make the call.** Real, and rarer than the four above. The answer is a human pass over that band, not an abandoned workflow.

Then fix the criteria and re-run on a fresh sample. Never tune against the sample you just used: once you have adjusted to catch the documents it flagged, it can no longer measure anything.

5.7 How many rounds

A round is one pass of the check, not a pass over the collection: run the criteria, sample, code the sample cold, compare against the target. The collection is classified once. What repeats is the measurement, and each round measures a different version of the criteria.

Two or three, against the target agreed in 5.1. Round one establishes a baseline and usually exposes criteria problems. Round two measures whether the rewrite fixed them, on a fresh sample. Round three confirms. If the results have not converged after three, the problem is not another round of tuning, it is that the criteria are trying to capture something that has not been decided.

Stop when you reach the target you set, not when you reach a result you are happy with.

06 The record

Validation produces numbers. The record is what makes them mean something to a person who was not there. Anything that would change how a reader interprets the output gets written down at the time, because reconstruction is where inconsistency enters.

Keep	Because
Criteria text, every version, with date and approver	Establishes what the process was instructed to do, and when that changed
Who wrote and who approved each version	Establishes that a qualified person specified the work
Validation results per round, with sample size and method	Without method and sample size the results are unverifiable
Adjudicated disagreements and their resolution	Shows the calls were examined rather than accepted
The reason recorded against each classification	Turns an individual call from an assertion into something inspectable
QC overrides, with who made them	Establishes human oversight as a fact rather than a policy
The family and attachment decision	Most frequently forgotten, and a common source of apparent inconsistency

Retention periods, and whether any of this is discoverable in a given jurisdiction, are questions for counsel rather than for a guide.

The practical form is one page attached to the matter file: what the criteria were, how many rounds ran, what the final numbers were, and what changed along the way. Teams that produce it are rarely asked for more.

07 A thirty-day rollout

The fastest way to find out whether this works on your matters is to run it on one where you already know the answers. Pick a closed matter your team reviewed conventionally: you have the coding decisions, the arguments the team had, and a basis for comparison no demonstration on sample data can give you.

Week	What happens
1	Select the closed matter. Pull five to ten documents the team already argued about. Write criteria for one tag using Section 3. Have someone not on the matter read them cold.
2	Run the classification. Build the control set from the human coding that already exists. Measure the classifier against it. Expect round one to disappoint.
3	Adjudicate disagreements into the three buckets. Rewrite the criteria against what you find. Re-run on a fresh sample.
4	Third round if needed. Write the one-page record. Decide whether the result would have changed how you staffed the original matter.

Two rules. Pick one tag, not the whole coding manual, because the point is to learn the specification skill rather than reproduce a review. And when round one comes back poorly, fix the criteria rather than the tool.

At the end you will know the thing no vendor evaluation can tell you: whether your team can write criteria well enough for this to work. That is the actual variable.

Appendix A. Criteria specification template

One per tag. Version it and date it.

Field	Content
Tag name	
Version and date	
Written by / approved by	
One-line definition	
Inclusion statement	What makes an individual document fall inside this tag, written for someone who has not seen the matter
Explicit exclusions	What does not qualify, including the adjacent material a general reading would pull in
Named entities	People with titles. Companies. Products and code names. Systems. Internal project names
Date range	Inclusive start and end
Document types in scope	
Family and attachment rule	Coded independently or at family level. State it explicitly
Edge cases checked	The documents the team already argued about, and the answer the criteria produce for each

Appendix B. Validation round worksheet

One per round. Sampling method and target are set before round one, per Section 5.1.

Field	Round 1	Round 2	Round 3
Criteria version used			
Date run			
Sampling method agreed with			
Sample size and how it was set			
Who coded the sample			
Result against the agreed target			
Disagreements: ambiguous criteria			
Disagreements: reviewer error			
Disagreements: classifier error			
Criteria changes made after this round			
Adjudicated by			

Appendix C. Privilege QC checklist

Run this on classifier output before any entry reaches a log.

- Every name identified as an attorney appears on the matter attorney list.
- Every attorney on the matter list is recognized by the criteria, including anyone whose title does not say counsel.
- Attorney-client communication and work product are distinguished rather than asserted together.
- Each entry identifies the participants and the capacity in which the lawyer was acting.
- No entry is a conclusion with no supporting facts.
- Documents of the same type are described consistently across entries.
- Third parties on the chain are addressed rather than ignored.
- In-house counsel communications in a business capacity are excluded, and the basis is recorded.
- Family and attachment treatment matches the criteria.
- A qualified attorney has reviewed and signed off the final entries.

Appendix D. Vendor diligence questions

Neutral questions for any platform under evaluation, including this one.

Model and output

- Does classification require a training or seed set, or does it work from written criteria?
- Is a reason recorded for each classification, and is it retained and exportable?
- Are criteria stored with version history, so an earlier run can be reproduced?
- Can the same criteria be applied to a new collection without reconfiguration?

Privilege

- Are attorneys identified in a dedicated field we can check against our own list?
- Does the output include drafted log entry text, and can we edit it before export?
- Are attorney-client and work product claims handled separately?

Validation

- What sampling is built in, and can we draw a random sample from the null set directly?
- Can validation results be exported with sample size and method attached?

Deployment and data

- Where does classification run in each deployment model offered?
- Does document text leave our environment at any point, and if so, where does it go?
- What is retained by you or any subprocessor, and for how long?
- Is our data used to train or improve any model?

Cost

- Is AI classification metered by document, by volume, or included?
- Does running classification on a small matter carry a separate charge?
- What drives the bill over the life of a long matter, and what does not?

A note on this guide

This is how Venio suggests approaching AI-assisted review. It is not a standard, and it is not legal advice.

The rules in Section 3, the checks in Section 4, and the procedure in Section 5 are practical recommendations drawn from how this work tends to go wrong. No authority has set them, and teams that do things differently are not doing them incorrectly. What is defensible in a given matter depends on your obligations, your jurisdiction and your facts, and those questions belong with counsel and with whoever sets your validation methodology.

Take what is useful, adapt what is not, and confirm anything that will carry weight.