

Ask a room of product leaders how often the premium AI models they rely on change, and you will get guesses ranging from "every few months" to "I honestly don't know." Ask a governance lead the same question, and the answer should be more precise, because the answer determines how often your documentation goes stale. One figure circulating in the industry is that a new model appears roughly every 4.5 days. I would not put that number in a board deck without checking it against your own vendor change logs. But the direction it points is clear: the model [how to measure human override rate](#) behind a decision in your workflow may not be the model that produced it last quarter.

This post is for people who own risk, compliance, and product decisions in regulated environments such as HR investigations, legal review, security operations, and behavioral threat assessment. The core argument is simple. Model churn is not just an engineering concern. It is a governance concern, and it needs a change control process, defensible documentation, and a named human who answers for each decision.

Why the pace matters more than the headline number

The exact release cadence matters less than three practical effects:

- **Outputs drift without announcement.** A vendor can update a model behind a stable name. The prompt you tested in March may behave differently in June.
- **Old justifications stop being reproducible.** If a case note says "the model flagged elevated risk," you need to know which model, which version, which prompt, and which settings produced that flag.
- **Teams start comparing models as a substitute for judgment.** When new options appear constantly, people reach for a second or third opinion instead of asking whether the evidence supports the conclusion.

Radomir Basta, CEO and Founder of Four Dots, has written publicly about the practical side of working with fast-moving AI tooling for marketing and business operations. The lesson that carries over to governance is that adoption speed tends to outrun documentation speed. Your controls have to be designed for change from the start.

Model replacement risk is a governance risk

Model replacement risk is the chance that a model you approved, tested, and documented is swapped out, updated, or deprecated in a way that changes the outcome of a process. It has three layers:

1. **Silent change:** the vendor updates the model without a version change you can see.
2. **Planned change:** you choose to switch models for cost, quality, or features.
3. **Forced change:** the vendor retires the model, and you must migrate on their schedule.

Each layer requires a different control. Silent change requires regression testing on a fixed set of reference cases. Planned change requires a formal approval step. Forced change requires a migration plan with a time box, so the team does not rush an untested replacement into a live investigation.

A change control process that fits model churn

Traditional change control boards meet monthly. That cadence does not work when the underlying tool may shift weekly. A workable process looks like this:

1. Maintain a register of every model, version, and configuration used in a decision-support workflow.

2. Maintain a fixed benchmark set of anonymized or synthetic cases with known expected handling.
3. Re-run the benchmark set on a schedule and whenever a vendor announces a change.
4. Route any material difference in outputs to a named reviewer before the new configuration goes live.
5. Record the approval, the reviewer, the date, and the rationale in the decision log.

Keep the approval step lightweight. The goal is not a committee for every model update. The goal is that a material change never reaches live work without a named person having looked at the evidence.

Threat assessment: false positives versus false negatives

In behavioral threat assessment, the two error types carry different costs, and a model does not weigh them for you. A false positive may lead to unnecessary monitoring, a stressful interview, or reputational harm to someone who did nothing wrong. A false negative may mean a real warning sign was missed. Both are serious. Neither is acceptable to ignore.

When you change models, your false positive and false negative rates can shift in ways nobody intended. A model that becomes more cautious will raise more flags. A model that becomes more concise may drop qualifying context. Your benchmark set should measure both error types separately, and the results should be reviewed by the person who owns the threshold decision.

Error type	Typical harm	Question to ask after a model change	Accountable owner
False positive	Unwarranted escalation, unfair treatment of an individual	Did the flag rate rise on benchmark cases with no real concern?	Head of Investigations, Dana Whitfield
False negative	Missed warning sign, delayed intervention	Did any known-concern benchmark case lose its flag?	Chief Security Officer, Marcus Oyelaran
	Documentation gap	Decision cannot be explained or reproduced later	Can we reconstruct the model, version, and prompt behind each flag?
			Governance Lead, the author of this post

Notice that each row names a person. A threshold without an owner is a threshold nobody will defend when it is challenged.

Multi-LLM disagreement is a signal, not a verdict

Multi-LLM orchestration tools, including platforms such as Suprmind.ai, let teams send the same case to several models and view side-by-side model outputs. This is useful. It is also easy to misuse.

The most common misuse is treating agreement as proof. If three models say a case is low risk, that does not establish low risk. They may share training data, share blind spots, or share the same flaw in how the prompt framed the question. Agreement tells you the models converged. It does not tell you they were right.

The more valuable use is treating disagreement as information. When models split sharply, the most likely explanation is that the case file is missing something: a date, a witness statement, a prior incident, or context about the individual's role. Disagreement is a prompt to go find out what is missing.

Three rules follow from this:

- **Do not average outputs.** A risk score of 6 from one model and 8 from another does not become 7 with meaning. Averaging hides the disagreement that matters most.
- **Do not treat outputs as a vote.** Two agreeing models are not a majority decision. The question is what evidence each model used, and whether that evidence is in the file.

- **Log disagreement as an open item.** When models split, record the split, the suspected information gap, and the follow-up owner.

Defensible documentation and named accountability

A decision that cannot be explained is a decision that cannot be defended. "The AI said so" is not an explanation. It is an admission that no human owned the reasoning. For any decision-support output that feeds a consequential action, your record should capture:

- The model name, version, and configuration at the time of the output
- The prompt or instruction template used
- The case inputs, or a reference to the locked version of them
- The outputs from each model consulted, including disagreements
- The human reviewer, their conclusion, and the reasons they gave
- The open questions that remained unresolved

That last item is easy to skip. In our case notes, we keep a running section titled "what we still do not know." It is often the most useful part of the record, because it shows the decision was made under known uncertainty rather than false confidence.

Pricing and access: what we can and cannot say

Several multi-LLM platforms offer a Start Free Trial with no credit card needed. That lowers the barrier to testing, which is good for evaluation. However, the scraped material available for this review does not list a price, so this post does not state one. Before you budget for any tool, get the current pricing directly from the vendor and record the date you checked. Pricing that changes with the model landscape is one more thing that belongs in your change register.





What we still do not know

- Whether the 4.5-day release figure holds up across providers and model families
- How often silent updates change outputs on our specific benchmark cases
- Whether our current false positive and false negative rates are stable across model versions
- How regulators in each of our jurisdictions will treat model-assisted assessments in disputes

A short checklist to start this week

1. Inventory every model and configuration in use. Owner: Governance Lead. Time box: 5 business days.
2. Build a benchmark set of 30 to 50 reference cases with known expected handling. Owner: Head of Investigations. Time box: 2 weeks.
3. Assign a named approver for any model change. Owner: Chief Security Officer. Time box: 1 week.
4. Add a "what we still do not know" section to every decision record. Owner: each case lead. Effective immediately.
5. Review the first full benchmark run and decide whether the change process needs adjustment. Owner: Governance Lead. Time box: 30 days.

The bottom line

Premium AI models are changing fast, and the exact pace will keep shifting. Governance cannot wait for the pace to settle. It needs a change control process that assumes change, a benchmark that measures both kinds of error, a disagreement practice that treats split outputs as open questions, and documentation that names a human owner for every consequential decision. Speed is not the enemy. Unexplained speed is.