

The Invisible 90%

A checklist for getting AI past your risk committee

The demo is the easy 10%. What decides whether an AI system reaches production in a regulated business is the work nobody demos: data lineage, model risk, evaluation, monitoring and sign-off.

This is the checklist I ran with clients. Work through it before you commit to a build, not after your second line of defence has said no. If you cannot answer an item, that is not a failure. It is the plan for the next two weeks.

1. Framing: is this the right problem

- ☐ The process is described end to end, including the exceptions, not just the happy path.
Most automation business cases are built on the happy path and die on the exceptions.
 - ☐ You can state the current cost in hours or FTE, measured rather than estimated.
 - ☐ You know what decision the output feeds, and who is accountable for that decision today.
 - ☐ You have agreed what 'good enough' means numerically before any model is built.
 - ☐ Someone has asked whether a rules engine would do the job.
If deterministic logic solves it, that is the cheaper and more defensible answer.
-

2. Data and privacy

- ☐ Every field the system touches has a known source, owner and lawful basis for use.
 - ☐ Personal data is identified, and you know whether it leaves your perimeter at any point.
 - ☐ Retention and deletion behaviour is defined for inputs, outputs and logs.
 - ☐ Training or fine-tuning data is separated from evaluation data, and neither leaks into the other.
 - ☐ You can reproduce the exact dataset used for any given model version.
-

3. Model risk

This is where most AI programmes stall, because it is treated as an approval gate rather than a design input.

- ☐ The model is registered in your model inventory, with a named owner.
- ☐ You have documented what the model is for, and explicitly what it must not be used for.
- ☐ Someone who did not build it has reviewed the design, data and assumptions.

- ☐ Failure modes are written down: what does a wrong output look like, and what is the downstream harm.
 - ☐ You know the fallback when the model is unavailable or its confidence is low.
 - ☐ For anything customer-affecting, you can explain an individual decision in language a customer would accept.
-

4. Evaluation

Without evals, every release is an act of faith and no reviewer can approve it honestly.

- ☐ You have a fixed evaluation set that reflects real cases, including hard and adversarial ones.
 - ☐ Evaluation is automated and runs on every change, not by hand before a demo.
 - ☐ You measure the thing the business cares about, not only model-level accuracy.
 - ☐ Performance is broken down by segment, so a good average cannot hide a bad subgroup.
Aggregate metrics are where fairness problems go to hide.
 - ☐ You have a documented threshold at which the system must not ship.
-

5. Operating it

- ☐ Inputs, outputs, model version and prompt or config version are logged for every call.
 - ☐ You can reconstruct, months later, exactly what the system did and why.
 - ☐ Drift and quality are monitored continuously, with a named person who sees the alert.
 - ☐ There is a documented rollback, and someone has actually tested it.
 - ☐ Cost per transaction is monitored alongside quality.
 - ☐ Human review is defined: which cases, by whom, and with what authority to overrule.
-

6. Sign-off

- ☐ Risk and compliance were involved in the design, not shown the result.
 - ☐ The audit trail would satisfy someone reconstructing a decision a year from now.
 - ☐ Third-party and model-supply-chain dependencies are documented, including what happens if a provider changes or withdraws a model.
 - ☐ Someone owns the system after go-live, with budget for monitoring and retraining.
 - ☐ The business case survives if you include the cost of all of the above.
-

If most of this is unanswered, that is normal.