

When Will Legal Be Good for AI?

Fixing legal work before automating it

Ryan McDonough

CONTENTS

PART I: THE WRONG QUESTION

Chapter 1: Why we keep asking when AI will be good enough for legal

Chapter 2: Where AI already works, and why

Chapter 3: The messy middle of legal work

PART II: WHAT AI EXPOSES ABOUT LEGAL WORK

Chapter 4: Under-specified tasks

Chapter 5: Inconsistency as an invisible dependency

Chapter 6: When judgement is doing too much work

Chapter 7: Process that lives in people

PART III: BECOMING GOOD FOR AI (WITHOUT CENTRING AI)

Chapter 8: Define work properly

Chapter 9: Write down assumptions

Chapter 10: Standardise language, not thinking

Chapter 11: Feedback as professional hygiene

Chapter 12: Process is infrastructure

PART IV: REFRAMING ADOPTION

Chapter 13: Where AI should not be used

Chapter 14: From tools to systems

Chapter 15: A better set of questions

CONCLUSION

The profession's side of the bargain

Why we keep asking when AI will be good enough for legal

In boardrooms, legal operations meetings, and legal tech conferences across the profession, one question dominates. It's asked by General Counsels worrying about their budget, by Managing Partners worrying about their business model, and by junior lawyers worrying about their careers.

"When will AI be good enough for legal work?"

It's a sensible, rational question. Legal work is high-stakes. It's the operating system of commerce and society, the mechanism by which we define rights, enforce obligations, and structure risk. A mistake in a contract can cost millions. A hallucination in a court filing can cost a licence to practice, as recent cases have shown. Precision matters. Provenance matters. Predictability matters.

So when a new technology arrives, particularly one as fluid, probabilistic, and occasionally confident in its errors as Generative AI, it's right that the profession pauses. It's right to ask for evidence. It's right to demand that the tool meets the standard of the task.

The question, sensible as it seems, rests on an assumption: that "legal work" is a fixed, static target, a high bar of quality and structure that AI must jump over, and that the current way legal work is done is optimal, rigorous, and ready to be automated, if only the machines could get a little bit smarter.

This assumption is wrong, and because it's wrong, the question "When will AI be good enough?" is leading us down a dead end. It frames the challenge entirely as a technology problem. It suggests that if we just wait for the next model, the next parameter count, the next breakthrough in reasoning, the problem will solve itself. We can sit back, arms folded, and wait for the vendors to impress us.

AI is often failing in legal contexts not because the AI is stupid, but because the legal work it's being asked to do is messy.

We hand over tasks that are poorly defined. We rely on implicit context that exists only in the heads of senior partners. We tolerate inconsistency in our drafting and our advice because human lawyers are very good at smoothing over chaos with judgement. We treat "review this contract" as a complete instruction, when in reality it's a shroud for a dozen unwritten preferences, risk appetites, and stylistic biases that we have never written down.

We are asking a calculator to solve an equation we haven't finished writing.

Early in my career, a partner handed me a printed form and said "I want this as a web form." Six months later, after no responses to clarification emails, I delivered what I had built. As you can imagine, it was not what they expected. The problem was not my technical skill, it was that "this as a web form" could mean a hundred different things, and I had guessed wrong.

The safety of the waiting room

There is a comfort in asking when AI will be good enough. It puts the burden of change on someone else.

If the problem is that the models hallucinate, then we wait for them to stop hallucinating. If the problem is that they cannot handle a 400-page document, we wait for the context window to expand. If the problem is that they miss subtle cross-references, we wait for better reasoning capabilities.

This waiting room is safe. We can feel tech-savvy (we are "monitoring the market") without doing the hard, unglamorous work of changing how we operate. We maintain the status quo while nodding sagely about the future.

However while we wait, the ground is shifting. The capabilities of these systems are improving at a pace that has no historical precedent. We have moved from amusing curiosity to passing bar exams to sophisticated reasoning in less time than it takes to train a trainee solicitor.

The gap is closing, but it's not closing evenly.

In tasks where legal work is structured, standard, and clearly defined, AI is already not just "good enough", but awesome. In e-disclosure, in clear-cut contract review, in structured data extraction, the debate is largely over. However, in the vast middle ground of daily legal practice, the advisory emails, the negotiation points, the drafting of bespoke clauses—progress feels stalled. We see impressive demos, but when we bring them back to our desks, they falter. They miss the point. They sound slightly wrong. They fail to grasp the specific, unwritten "way we do things here".

This is not a failure of intelligence. It's a failure of specification.

The mirror

When you try to deploy AI into a legal workflow, you are rarely just testing software. You are testing the workflow itself.

AI acts as a merciless mirror. It reflects exactly what you give it. If your precedents are inconsistent, the AI will be inconsistent. If your instructions are vague, the AI will be vague. If your risk tolerance is unspoken, the AI will guess, and it will often guess wrong.

Human lawyers are forgiving. A junior associate knows that when you say "turn this round quickly", you don't mean "ignore all the risks". They know that when you haven't specified a governing law, you probably mean the one you usually use. They absorb the ambiguity of human communication and fill in the gaps with common sense and cultural knowledge.

AI has no common sense. It has no cultural knowledge of your specific firm or department unless you explicitly provide it. It reveals the gaps in our processes that humans have been filling for decades.

This is why the project of adopting AI is so difficult. It forces us to confront the fact that much of what we call "legal expertise" is actually a mix of genuine judgement and compensatory admin. We spend a huge amount of expensive human time compensating for the fact that we haven't standardised our inputs, clarified our outputs, or written down our rules.

The leaders who are seeing success with AI today are not the ones who bought the most expensive tool. They are the ones who looked in the mirror and decided to tidy up the room.

A new question

This book is not a guide to prompting. It's not a comparison of Large Language Models. It's not a forecast of when the "robot lawyer" will arrive to take your job.

It's a proposal for a different way of thinking about legal operations. It argues that the barrier to AI adoption is often not the AI itself, the barrier is frequently our own processes and practices.

We need to stop asking "When will AI be good enough for legal?" and start asking "When will legal be good for AI?"

When will our work be defined clearly enough that a machine can reliably assist with it? When will our data be clean enough to train on? When will our expectations be explicit enough to measure against?

The surprising, happy truth is that the work required to answer these questions, the work of standardisation, of clarity, of process hygiene, is good for us anyway. Even if AI vanished tomorrow, a legal team that has defined its work, written down its assumptions, and standardised its outputs is a better, happier, more efficient team.

AI is just the catalyst. The real work is ours to do.

Where AI already works, and why

The debate about AI in legal is often framed as if we are still waiting for the technology to prove itself. But in several corners of legal practice, that debate is over. AI is not just working, it's dominant, reliable, and in some cases, indispensable.

Understanding where AI succeeds, and why, is the fastest route to understanding where it struggles.

E-disclosure: the change that almost no one noticed

In 2010, if you were a junior lawyer at a large commercial firm, there was a reasonable chance you would spend weeks in a windowless room reviewing printed documents. Privilege review. Relevance tagging. Redaction. It was tedious, expensive, and error-prone.

Today, that work is substantially automated. Technology-assisted review (TAR) systems use machine learning to identify relevant documents, predict privilege, and flag sensitive information. They do this faster, cheaper, and more consistently than humans ever could. Privilege review still requires significant human oversight, but the volume of documents that need manual review has been reduced by orders of magnitude.

The technology is not perfect, it makes mistakes, but the key difference is consistency. Human performance naturally varies, concentration drifts over long sessions, interpretation can shift between document 1 and document 10,000. AI applies the same logic throughout. The AI's mistakes are measurable, auditable, and improvable. The system gets better with feedback in ways that are difficult to replicate with human review at scale.

E-disclosure works because the task has clear boundaries. The question is not "What does this document mean in the broader strategic context of the case?" it's "Does this document contain information relevant to the defined scope of disclosure?" That question has an answer. It can be tested. It can be validated against a sample set.

The task is also high-volume and repetitive. You are not reviewing one contract. You are reviewing 200,000 emails. The cost of human review is prohibitive, and the risk of inconsistency is high. AI does not get bored. It does not drift. It applies the same logic to document 1 and document 200,000.

This is the pattern. AI works when the task is defined, bounded, and repeatable.

Contract review: the spectrum of success

Contract review is more complex than e-disclosure, and the results are more mixed. Even here though, there are pockets of genuine success.

If you ask an AI system to extract defined data points from a standard contract, party names, governing law, termination notice periods, it will do so with near-perfect accuracy. This is not impressive in a cognitive sense. It's pattern matching. But it's useful, and it's reliable.

If you ask the same system to "review this contract for risk", the results will vary wildly depending on how well you have defined "risk". If you mean "flag any clause that deviates from our standard position on liability caps", and you have given the system your standard position, it will succeed. If you mean "tell me if this is a good deal", it will fail, because "good" is not a specification.

The difference is not the AI. The difference is the task.

I have seen contract review concepts succeed brilliantly and fail embarrassingly, often using the same underlying technology. The variable is not the model. It's whether those deploying it have done the work to define what "review" means in their context.

Drafting: the illusion of creativity

Generative AI can draft clauses, emails, and even entire contracts with alarming fluency, mimicking tone, structure, and legal style. This feels like magic, and it's easy to assume that the hard problem is solved, but fluency is not the same as correctness. A well-drafted clause is not just grammatically coherent. It reflects a specific risk position, a commercial intent, and often a negotiation history. It fits into a broader document structure. It anticipates how it will be read by the counterparty, by a judge, or by an auditor.

AI can generate text that looks like a clause. Whether that clause is fit for purpose depends entirely on how well you have specified what "fit for purpose" means.

Where I have seen AI drafting work well, it's in highly standardised contexts. Generating a first draft of a non-disclosure agreement from a pre-approved template. Populating a standard set of definitions. Adapting a precedent clause to a new jurisdiction based on explicit rules.

Where I have seen it fail, it's where the drafter is expected to "just know" what the client wants, what the market norm is, or what the senior partner's stylistic preferences are. The AI does not "just know". It guesses. And sometimes it guesses confidently and wrongly.

Legal research: speed and surface area

Legal research tools using AI have transformed the speed at which you can find relevant case law, statutes, and commentary. They can surface connections between cases that a human might miss and summarise long judgments in seconds. However, they do not replace the need to read the judgment, assess whether the reasoning in a 20-year-old Court of Appeal decision is still good law, or decide whether a case is materially similar to yours or merely analogous.

What they do is expand the surface area of what you can cover in a given amount of time. They let you cast a wider net. They reduce the risk that you miss something important because you did not think to search for it.

This is valuable. But it's not the same as legal analysis. It's a tool that makes analysis faster and broader. The analysis itself still requires a human.

The common thread

E-disclosure, contract data extraction, template-based drafting, and legal research all share a set of characteristics:

- The task is defined clearly enough that success can be measured.
- The inputs are structured or can be made structured.
- The outputs are predictable and testable.
- The volume or complexity justifies automation.

Where these conditions hold, AI is not just "good enough". It's often better than humans, at least on the dimensions that matter: speed, consistency, cost, and scalability.

Where these conditions do not hold where the task is ambiguous, the inputs are messy, the outputs are subjective, or the context is implicit, AI struggles. Not because it's unintelligent, but because it has not been given a solvable problem.

The lesson is not "AI works in some areas and not others". The lesson is "AI works where the work has been made ready for it".

The question is not whether your area of law is suitable for AI. The question is whether you have done the work to make it suitable.

The messy middle of legal work

There is a category of legal work that doesn't fit neatly into any category—work that is neither clerical nor purely strategic, not high-volume enough to justify industrial-scale automation, but too common to ignore.

It's the work that fills the day of most lawyers. The advisory email to a business unit. The negotiation call with a counterparty. The review of a non-standard clause in an otherwise standard contract. The drafting of a board paper on a regulatory risk.

This is the messy middle. And this is where AI adoption stalls.

Too complex for a checklist, too routine for deep thought

The messy middle is characterised by tasks that require judgement, but not the kind of judgement that gets written up in law journals. It's the judgement of "what does the client actually need here?" or "how hard should I push on this point?" or "is this worth escalating?"

A junior lawyer learns this through osmosis. They watch how a senior partner responds to a vague question from a client. They notice which risks get flagged in a contract review and which get accepted without comment. They absorb the unwritten rules about tone, escalation, and commercial context.

This knowledge is real, valuable, and almost entirely tacit. It exists in the heads of experienced lawyers, and it's transmitted through apprenticeship, not documentation.

When you try to hand this work to an AI, the system has no senior partner to watch. It has no sense of "how we do things here". It does not know that when the client asks for a "quick view", they mean a two-paragraph email, not a ten-page memo. It does not know that a liability cap is non-negotiable with this client but flexible with that one.

The AI is not stupid, it's under-briefed.

The illusion of simplicity

The messy middle often looks simple from the outside: "Just review this contract." "Just draft a response to this email." "Just check if this is compliant." However, each of these instructions is a compression of a dozen unstated assumptions. What does "review" mean? Review for what? Against what standard? With what risk tolerance? For what purpose?

When a senior lawyer asks a junior to "review this contract", the junior knows, or learns quickly, that this means:

- Check it against our standard template.
- Flag any deviations that increase our risk.
- Ignore minor wording changes that do not affect substance.
- Escalate anything that touches on the three issues we discussed last week.
- Keep it to one page unless there is something serious.

None of this is written down. It's context and context is the thing that AI lacks.

The compensatory expertise trap

Experienced lawyers are extraordinarily good at compensating for poor process. They can take a vague instruction, a messy precedent, and an incomplete set of facts, and still produce something useful. They fill in the gaps. They make assumptions. They apply common sense.

This ability is a form of expertise, and it's highly valued. But it's also a trap.

When expertise is spent compensating for the absence of structure, it's not available for the work that genuinely requires expertise. There is a difference between helping a client articulate their business needs (which is valuable counseling) and compensating for vague instructions from internal stakeholders because no one has defined the process. The senior lawyer who spends an hour tracking down the last version of a clause, or reconciling two inconsistent instructions from different parts of the business, or guessing at risk thresholds that should be documented, is not doing senior lawyer work. They are doing project management and archaeology.

Agentic AI can do some of this compensatory work. It can ask clarifying questions, search for previous versions, and make reasonable inferences, but all of this is expensive, unreliable, and misses the point. If your process requires an AI agent to spend its time hunting for information that should be documented, reconciling instructions that should be consistent, and guessing at thresholds that should be explicit, you have not automated the work, you have just automated the compensation for poor process.

This creates tension. It feels like the AI is failing, but often, the AI is just refusing to pretend that the task was well-defined in the first place.

Why the messy middle matters

The messy middle is where most legal work happens. It's where most of the cost sits. It's where most of the frustration lives, both for lawyers and for clients.

Clients do not want a ten-page memo when they asked for a quick view. They do not want to pay senior lawyer rates for work that feels routine. They do not want to wait three days for an answer to a question that feels simple.

Lawyers do not want to spend their time on repetitive, under-stimulating work that does not use their training. They do not want to be the bottleneck. They do not want to work weekends because the volume is unpredictable and the process is inefficient.

The messy middle is the zone where AI could have the most impact. But it's also the zone where AI is least ready to help, because the work itself is least ready to be helped.

The path forward is not more AI

The instinct, when AI fails in the messy middle, is to wait for better AI: smarter models, better reasoning, longer context windows, more training data. However, the problem is not the AI—the problem is that we are asking it to operate in a fog.

If you cannot explain to a competent junior lawyer what you want them to do, you cannot explain it to an AI. If your process relies on people "just knowing" what is expected, it will not survive automation.

The messy middle will remain messy until we do the work to clarify it. That work is not technical. It's organisational. It's about defining tasks, writing down assumptions, and making implicit knowledge explicit.

This is hard and unglamorous work that does not come with a software licence or a press release. But it's the only way through.

AI is not the solution to the messy middle. Clarity is. AI is just the thing that makes the cost of unclarity impossible to ignore.

Under-specified tasks

"Review this contract."

it's one of the most common instructions in legal practice. It's also one of the least useful, well at least when you want a predictable, delegable outcome.

Review it for what? Against what standard? With what level of scrutiny? For what purpose? What should the output look like? What risks matter? What risks can be ignored?

When a senior lawyer gives this instruction to a junior, the junior does not ask these questions. They infer the answers from context, from past experience, from the culture of the team. They know or they learn what "review" means in this situation. Sometimes this vagueness is intentional: the senior lawyer wants the junior to exercise judgement, to learn what matters.

When you want to delegate work reliably whether to a junior who has not yet learned your preferences, or to an AI system, or to a process that can scale vagueness becomes a problem. The instruction needs to be explicit. When you give "review this contract" to an AI, it has no context to infer from. It will do something. But whether that something is useful depends on luck, not design.

The unrecognised work of task definition

Human communication is efficient because we leave out the obvious. We do not say "Review this contract for compliance with our standard risk position on liability, indemnity, and termination, and produce a one-page summary of deviations with a recommendation on whether to escalate." We say "Review this contract."

The rest is implied.

This works when the person receiving the instruction shares your context. It fails when they do not. And AI does not share your context unless you give it explicitly.

I once tried to automate the generation of due diligence reports. It depended on data being collated in a standardised way, but the underlying process was people sending information around in emails and Word documents. The automation project became a process redesign project, because you cannot automate chaos.

The problem is not that AI is bad at following instructions. The problem is that "Review this contract" is not an instruction. It's a placeholder for an instruction that you have not yet written down.

The software engineering parallel

In software engineering, there is a concept called a "user story". It's a structured way of defining a task so that it can be implemented reliably.

A bad user story looks like this: "Make the system faster."

A good user story looks like this: "As a user searching for a document, I want search results to return in under 2 seconds for queries with fewer than 10,000 results, so that I do not lose focus while waiting."

The difference is specificity. The good user story defines who is affected, what success looks like, and why it matters. It's testable. You can measure whether you have achieved it.

Internal legal instructions often lack this level of specificity. There is a reliance on shared understanding, professional norms, and the ability of skilled people to interpret vague instructions.

This works, until you try to automate it.

What "review" actually means

Let's take the instruction "Review this contract" and unpack what it might mean in practice.

It could mean:

- Extract key commercial terms and populate a tracker.
- Check for compliance with our standard template and flag deviations.
- Identify any clauses that increase our liability beyond agreed thresholds.
- Assess whether the contract is consistent with the commercial deal as described in the email chain.
- Check for any terms that conflict with our regulatory obligations.
- Provide a risk rating and a recommendation on whether to sign.

These are all legitimate interpretations of "review". But they are different tasks. They require different inputs, different expertise, and different outputs.

A human lawyer will usually figure out which one you meant. An AI will not. It will guess, or it will do all of them badly, or it will do the easiest one and ignore the rest.

The cost of under-specification

Under-specification has always been expensive. It leads to rework, misunderstanding, and wasted effort. But humans are good at absorbing this cost, they'll ask clarifying questions, make reasonable assumptions and iterate.

AI does not iterate gracefully. It does what you told it to do, even if what you told it to do was not what you meant. And because AI is fast, it can produce a large volume of wrong output before you notice.

This is why early AI pilots in legal often fail. The technology works, but the task was never defined well enough for any system human or machine, to succeed reliably.

The discipline of specification

Defining tasks properly is hard. It requires you to think clearly about what you actually want, not just what you have always asked for.

It requires you to separate the essential from the incidental. To distinguish between "this is how we have always done it" and "this is what actually matters".

It requires you to write things down. To make the implicit explicit. To turn "you know what I mean" into a specification that someone who does not know what you mean can still follow.

This feels bureaucratic and like it will slow you down. However, the alternative is worse: remaining dependent on a small number of people who "just know" how things work, unable to scale, delegate reliably, or automate. When those people leave, the knowledge leaves with them.

A practical test

If you want to know whether a task is well-specified, try this:

Write down the instruction you would give to an AI. Then give that same instruction to a competent junior lawyer who has never worked on this type of matter before.

If they can complete the task to your satisfaction without asking clarifying questions, the task is well-specified.

If they cannot, the task is under-specified and no amount of AI capability will fix that.

The benefit beyond AI

The discipline of specifying tasks clearly is valuable even if you never use AI.

It makes delegation easier, training faster, and quality more consistent, while reducing the risk of misunderstanding. It protects you from the "bus factor"—the risk that critical knowledge exists only in one person's head—and makes your team more resilient, scalable, and efficient.

AI is just the forcing function. It's the thing that makes under-specification visible and costly. But the solution, clarity, is good for you regardless.

See Appendix: Template 1 (Task Definition Template) for a practical template you can use immediately.

Inconsistency as an invisible dependency

If you ask ten lawyers in the same firm to draft a confidentiality clause, you will get ten different clauses. Some of the differences will be substantive. Most will not. Different word order. Different defined terms. Different levels of detail. Different stylistic preferences.

This inconsistency is so normal that we barely notice it. We treat it as a feature of legal work, not a bug. After all, law is not code. There is no single "correct" way to draft a clause. However, inconsistency has a cost, and that cost becomes visible the moment you try to automate anything.

The human tolerance for chaos

Humans are extraordinarily good at dealing with inconsistency. We can read a clause drafted in an unfamiliar style and still extract the meaning. We can reconcile two conflicting instructions by inferring intent. We can adapt to local conventions without being told what they are.

This ability is a form of intelligence, and it's one of the reasons legal work has resisted automation for so long. The messiness that would break a rigid system is smoothed over by human adaptability. However, this adaptability comes at a cost. Every time a lawyer has to interpret an inconsistent precedent, or reconcile conflicting drafting styles, or figure out what a vague term means in this context, they are spending cognitive effort on a problem that should not exist.

AI does not have this tolerance. If your precedents are inconsistent, the AI will be inconsistent. If your terminology drifts, the AI will drift. If your risk thresholds are unwritten and variable, the AI will apply them unpredictably.

AI does not smooth over chaos, it reflects the chaos back at you.

The myth of bespoke quality

There is a belief in legal that inconsistency is a sign of quality. That each matter is unique, and therefore each document should be crafted individually to reflect the specific circumstances.

This is true for genuinely bespoke work. A complex cross-border acquisition is not the same as a standard software licence. The documents should be different. However, it's important to distinguish between three types of variation:

Contextual variation is legitimate. Different clients have different risk appetites. Different industries have different market standards. Different jurisdictions require different approaches. A liability cap that is standard in a software licence may be inappropriate in a construction contract. This variation reflects genuine differences in context and should be preserved.

Evolutionary drift is understandable. Legal practice evolves. A precedent from five years ago may reflect older market practice or regulatory requirements. The variation is not intentional, but it reflects the passage of time rather than poor discipline.

Accidental inconsistency is the problem. This is the result of different lawyers having different habits. Different training. Different preferences for how to structure a sentence or define a term. It's the result of precedents being copied and adapted over years without anyone standardising them. Same client, same context, different wording for no reason.

This kind of inconsistency does not add value. It adds friction.

It makes documents harder to review, because you cannot assume that similar wording means the same thing. It makes precedents less useful, because you have to spend time figuring out which version to use. It makes training harder, because there is no single "right" way to do things, and it makes automation impossible, because the system has no stable foundation to build on.

The engineering parallel: code before linters

In the early days of software development, every programmer had their own style. Some used tabs, some used spaces. Some put the opening brace on the same line, some put it on the next line. Some used verbose variable names, some used abbreviations.

This inconsistency made code harder to read, harder to review, and harder to maintain. Bugs would hide in the gaps between styles and new developers would waste time figuring out the local conventions.

The solution was not to hire "better" programmers. The solution was to standardise, to introduce linters, formatters, and style guides, to agree on a set of conventions and enforce them automatically.

The result was code that was easier to read, easier to review, and easier to maintain. Not because the programmers were more skilled, but because the inconsistency that obscured their skill had been removed.

When standardisation doesn't apply

Some legal work is genuinely bespoke and should remain so. Novel regulatory issues with no precedent, first-of-kind transactions, high-stakes litigation strategy, and cross-border deals with unique jurisdictional complexities cannot and should not be standardised. These are not "poor process"—they are genuinely one-off.

The goal is not to force standardisation onto work that is genuinely unique. The goal is to identify the routine work that is currently treated as bespoke when it does not need to be.

The test is simple: have we done this type of work before? If yes, it should have a standard approach. If no, it may not. The problem is not that all legal work should be identical. The problem is that work that should be consistent often is not.

This was initially resisted. Programmers argued that style was a matter of personal preference, that enforcing consistency would stifle creativity, that it was a waste of time. However, the teams that adopted these practices became faster, more reliable, and more scalable. The consistency did not reduce quality. It freed up cognitive effort for the work that actually mattered.

Legal is in the same position now. The inconsistency that we tolerate is not a sign of quality. It's a sign of immaturity.

I have worked with engineering teams that rejected source control and continuous deployment. It was easier to paste code directly onto the server and create a new folder for each version. This worked until someone thought they were on the development server but were actually on production, and replaced live code and live data. The crisis changed minds faster than any argument about best practice.

Where inconsistency hides

Inconsistency in legal work is not always obvious. It hides in:

- **Terminology:** Is it "confidential information" or "proprietary information"? Does "material breach" mean the same thing in every contract?
- **Structure:** Does the liability clause come before or after the indemnity clause? Are definitions in a schedule or inline?
- **Thresholds:** What counts as a "material" change? When does a risk need to be escalated?
- **Tone:** Is advice direct or hedged? How much context is included?
- **Process:** Who reviews what? When is a second pair of eyes required?

Each of these inconsistencies is small. But they compound. They create an environment where nothing is predictable, and everything requires interpretation.

The cost of inconsistency

Inconsistency makes everything slower and more expensive.

It makes review harder, because you cannot pattern-match. Every document has to be read as if it's the first time you have seen this type of clause.

It makes delegation harder, because you cannot give clear instructions. "Draft it like the last one" only works if "the last one" is consistent with the one before it.

It makes training harder, because there is no single standard to teach. New joiners have to learn not just the law, but the unwritten preferences of every senior lawyer they work with, and it makes reliable automation extremely difficult, because the system has no stable input to learn from.

Standardisation is not rigidity

The objection to standardisation is usually that it will make legal work robotic. That it will remove the flexibility to adapt to different circumstances. That it will reduce quality to the lowest common denominator.

These concerns are not unreasonable. Lawyers have seen "standard" clauses fail in unexpected contexts. There is a professional duty to adapt to specific circumstances. And there is a legitimate fear that over-standardisation could create rigidity that leads to malpractice when the standard proves inadequate for a novel situation.

However, this is a misunderstanding of what good standardisation means.

Standardisation is not about making every contract identical. It's about making the default predictable, so that deviations are intentional, documented, and visible.

it's about agreeing that "confidential information" means the same thing in every contract, unless there is a specific reason to define it differently, and when you do define it differently, you document why. It's about having a standard structure, so that when you deviate from it, you know why, and the deviation is a considered choice rather than an accident.

it's about reducing the cognitive load of the routine, so that you have more capacity for the genuinely complex. The goal is a safe default with documented deviations, not a rigid requirement that cannot be questioned.

The first step: naming things

The simplest form of standardisation is agreeing on what things are called.

If your team uses three different terms for the same concept, pick one and use it consistently. If your contracts define "business day" in five different ways, pick the best one and make it the default.

This sounds trivial. It's not. Consistent terminology makes documents easier to read, easier to search, and easier to automate.

It also forces clarity. If you cannot agree on what to call something, it's often because you have not agreed on what it actually is.

The second step: templates as defaults, not constraints

A good template is not a straitjacket. It's a starting point.

It says "This is the default structure, the default wording, the default risk position. If you need to deviate, you can. But you should know that you are deviating, and you should have a reason."

This makes deviations visible. It makes them intentional. It makes them easier to review, because the reviewer is not trying to figure out whether the difference is meaningful or accidental, and it makes automation possible, because the system can learn the default and flag the deviations.

The benefit beyond AI

Consistency is not just about making AI work. It's about making your team work better.

It reduces errors, speeds up review, makes training easier, delegation more reliable, and your output more professional.

It also makes your work more defensible. If you can show that you followed a standard process, applied a standard template, and deviated only for documented reasons, you are in a much stronger position than if every document is a bespoke creation with no clear lineage.

AI is just the thing that makes the cost of inconsistency impossible to ignore. But the solution, standardisation, is good for you anyway.

When judgement is doing too much work

Legal judgement is the thing that makes lawyers valuable. It's the ability to assess risk, weigh competing interests, interpret ambiguous language, and make decisions in the face of uncertainty. However, judgement is often doing work it should not have to do.

Compensatory judgement fills gaps left by poor process: missing information, undocumented rules, and inconsistencies that should never have existed.

This is not good judgement. This is judgement being wasted.

The difference between deciding and compensating

Good judgement decides between options. It weighs risks. It chooses a path when there is no clear answer.

Compensatory judgement is different. It's the work you do when the task itself is broken. When the instructions are vague. When the precedent is inconsistent. When internal stakeholders have not articulated what they need.

Compensatory judgement is not a sign of expertise but a sign of poor infrastructure.

To be clear: helping a client understand and articulate their business needs is valuable senior lawyer work. Understanding unspoken concerns, reading organisational dynamics, and counseling on strategy are core legal skills. But that is different from compensating for internal process failures. A senior lawyer who spends an hour figuring out which version of a template is current, or reconciling contradictory instructions from different business units, or guessing at risk thresholds that should be documented, is not exercising legal judgement. They are doing project management and archaeology.

This work is necessary. But it's necessary only because the underlying process is broken.

The hero lawyer problem

In many legal teams, there is a senior lawyer who "just gets it". They know what the client wants before the client does. They can look at a contract and immediately spot the issue. They can draft a clause that works in any context.

This person is invaluable. They are also a single point of failure.

Their expertise is not documented. It's not transferable. It exists in their head, and when they leave, it leaves with them.

Worse, their expertise is often compensatory. They are good at their job not because the job is well-designed, but because they have learned to navigate a badly-designed system.

They know which clients are risk-averse and which are commercial. They know which clauses the senior partner cares about and which can be ignored. They know how to interpret a vague instruction and turn it into something useful.

This is real skill, but it's a skill that should not be necessary.

If your process relied on clear instructions, consistent precedents, and documented risk thresholds, you would not need a hero. You could delegate the work to a competent junior and get a predictable result.

The AI mirror

When you try to deploy AI into a workflow that relies on compensatory judgement, the AI fails. Not because it's unintelligent, but because it cannot compensate.

It cannot guess what the client wants. It cannot infer the risk threshold from the tone of an email. It cannot "just know" which version of a clause to use.

The AI forces the question: what is the actual rule here? Often, the answer is: there is no rule. There is just a senior lawyer who knows.

This is not sustainable—it does not scale, survive staff turnover, or support automation.

Separating judgement from hygiene

The goal is not to eliminate judgement, the goal is to protect it.

Judgement should be reserved for the decisions that genuinely require it. The strategic choices. The risk trade-offs. The interpretation of ambiguous law.

It should not be spent on:

- Figuring out what the client actually asked for.
- Reconciling inconsistent precedents.

- Guessing what the risk threshold is.
- Compensating for missing information.

These are process problems. They should be solved with process, not with expensive human judgement.

A practical example

Imagine a lawyer is asked to review a contract. The instruction is vague. The precedent is inconsistent. The client has not specified their risk tolerance.

The lawyer spends:

- 10 minutes figuring out what "review" means in this context.
- 15 minutes finding the right precedent and reconciling it with the current draft.
- 20 minutes making assumptions about what the client will and will not accept.
- 15 minutes actually applying legal judgement to assess the risk.

Only a quarter of the time is spent on the work that requires a lawyer. The rest is compensatory effort.

Now imagine the same task in a well-structured environment:

- The instruction is clear: "Check for deviations from our standard position on liability and indemnity, and flag anything that exceeds our agreed risk threshold."
- The precedent is standardised and up to date.
- The risk threshold is documented.

The lawyer spends:

- 5 minutes reviewing the contract against the standard.
- 10 minutes applying judgement to assess whether a flagged deviation is acceptable in this context.

The total time is lower, and the proportion of time spent on actual legal judgement is higher. The lawyer is doing lawyer work, not project management.

The engineering parallel: technical debt

In software engineering, there is a concept called "technical debt". It's the cost of taking shortcuts. Of building systems that work, but are hard to maintain, hard to extend, and hard to understand.

Technical debt accumulates. And eventually, it becomes so expensive that the team spends more time working around the debt than building new features.

The solution is not to hire better engineers. The solution is to pay down the debt. To refactor the code. To document the decisions. To standardise the patterns.

Legal has the same problem. We have accumulated process debt. We have built systems that rely on hero lawyers compensating for poor infrastructure, and now, as we try to scale, or automate, or simply cope with increasing volume, the debt is becoming unmanageable.

Paying down the debt

Paying down process debt is unglamorous work that is not billable, does not win clients, and does not get you promoted, but it's necessary.

It means:

- Writing down the rules that currently exist only in people's heads.
- Standardising the precedents that are currently inconsistent.
- Documenting the risk thresholds that are currently implicit.
- Clarifying the instructions that are currently vague.

This work protects judgement. It ensures that when a lawyer exercises judgement, they are deciding something that matters, not compensating for something that is broken.

The benefit beyond AI

Even if you never use AI, reducing the burden of compensatory judgement makes your team better.

It makes work more predictable, delegation more reliable, training faster, and reduces the risk of errors.

It also makes the work more satisfying. Lawyers did not train for years to spend their time compensating for vague internal instructions or reconciling inconsistent precedents. They trained to exercise judgement on complex problems.

Give them the infrastructure to do that, and they will be happier, more productive, and more valuable.

AI is just the thing that makes the cost of compensatory judgement impossible to ignore. But the solution, better process, is good for you anyway.

Process that lives in people

If you ask a new joiner how to complete a task in a legal team, you will often get a person's name rather than a written procedure.

"Ask Sarah, she knows how we handle these."

"Check with Tom, he did the last one."

"It depends. If it's for this client, do it this way. If it's for that client, ask Maria."

The process exists. But it exists in people, not in documentation. It's transmitted through observation, conversation, and trial and error.

This works, until it does not.

The bus factor

In software engineering, there is a dark joke called the "bus factor". It's the number of people who would need to be hit by a bus before a project becomes unmaintainable.

If the bus factor is one, you have a problem. All the critical knowledge is in one person's head. If they leave, get sick, or actually get hit by a bus, the project is in trouble.

Legal teams often have a bus factor of one. Sometimes less.

There is one person who knows how to handle a particular type of matter. One person who knows what the client's risk appetite is. One person who knows which version of the template to use.

This is not because that person is hoarding knowledge but because no one has written it down.

Knowledge by osmosis

Legal training has always relied on apprenticeship. You learn by watching, by doing, by making mistakes and being corrected.

This is a good model for developing judgement. You cannot learn to assess risk from a manual. You need to see it in context, applied to real situations, with real consequences. However, not everything requires apprenticeship. Some things are just facts. Rules. Procedures. Preferences.

These things can be written down. They should be written down. But often, they are not.

Instead, they are transmitted by osmosis. A junior lawyer learns that this client prefers short emails by sending a long one and being told to shorten it. They learn that this type of clause is non-negotiable by trying to negotiate it and being overruled.

This is inefficient. It's also fragile. If the senior lawyer who holds this knowledge leaves, the knowledge leaves with them.

The cost of tacit knowledge

Tacit knowledge is expensive in several ways.

It makes onboarding slow. A new joiner cannot just read the manual and get started. They have to ask questions, make mistakes, and gradually absorb the unwritten rules.

It makes delegation risky. You cannot confidently hand work to someone who does not "just know" how it should be done.

It makes scaling impossible. If the process exists only in people's heads, you cannot hire your way out of a capacity problem. You can only hire people and wait for them to absorb the knowledge, and it makes automation impossible. You cannot automate a process that you cannot describe.

Early in my career, an engineer left who was the only person who knew how a certain library compiled. The source code existed only on his machine. I spent weeks reverse-engineering the library to make sense of it. Proper source control and documentation solved this problem permanently, but only after we had paid the cost of not having them.

The illusion of complexity

When you ask someone to document a process, they often resist. They say "It's too complex to write down" or "It depends on the situation" or "You just have to know."

Sometimes this is true. Some processes genuinely require judgement at every step, and cannot be reduced to a checklist. However, often it's an illusion—the process is not complex, it's just undocumented.

When you force someone to write it down, they discover that most of it's routine. There are standard steps. Standard inputs. Standard outputs. The complexity is in a small number of decision points, not in the entire process.

Once you separate the routine from the genuinely complex, you can document the routine and reserve human judgement for the complex.

What to document

Not everything needs to be documented. Some knowledge is genuinely tacit and cannot be reduced to writing. Some information should not be documented for privilege or confidentiality reasons. And some regulatory requirements create necessary process friction that cannot be eliminated. However, the following should be documented where possible:

- **Standard processes:** How do we handle a particular type of matter? What are the steps? Who is involved? What are the outputs?
- **Decision rules:** What is our risk threshold? When do we escalate? What deviations from the standard are acceptable?
- **Precedents and templates:** Which version is current? When should each be used? What are the standard positions on key clauses?
- **Client preferences:** What does this client care about? What is their risk appetite? What tone do they prefer?

This is not a bureaucratic exercise. It's infrastructure. It's the foundation for delegation, scaling, and automation. And it should be done with awareness of what cannot or should not be documented, rather than attempting to codify everything.

The privilege tension

Privilege and confidentiality create real constraints on documentation. Documenting risk thresholds could waive privilege if those documents become discoverable in litigation. Process documentation might reveal litigation strategy. Client-specific decision frameworks might be confidential. And feedback loops that track when advice was wrong could create discoverable records that undermine your position in future disputes.

These are not theoretical concerns. They are real risks that need to be managed thoughtfully.

This does not mean documentation is impossible. It means you need to be deliberate about what you document, where you store it, and how you label it. Work with litigation counsel to identify what can be documented without creating discovery risk. In many cases, the answer is "more than you think, but less than ideal." You can document standard processes without documenting client-specific strategies. You can document decision frameworks without documenting specific decisions. You can create feedback mechanisms that improve quality without creating a discoverable record of every mistake.

The goal is not to avoid all documentation out of fear of discovery. The goal is to be strategic about what you document and how you protect it.

The resistance to documentation

Documentation is often seen as a chore. It's not billable. It does not win clients. It feels like it slows you down.

These are not just excuses. They reflect real structural constraints.

The economic barrier

The billable hour model creates a fundamental conflict. Firms are paid for time spent, not for efficiency gained. Documentation reduces billable hours. Standardisation makes work delegable to cheaper resources. Process improvement is economically rational for clients but economically irrational for many firms.

For law firm partners, documentation can threaten their book of business. If a junior lawyer can follow documented processes to complete work that previously required a senior lawyer's tacit knowledge, the senior lawyer becomes less essential. The partner who documents their expertise is, in some sense, making themselves replaceable. This is not a failure of will. It's a rational response to economic incentives.

For in-house legal teams, the constraint is different but equally real. Many legal teams are deliberately kept lean due to budget pressures. The GC may want to document processes, but the team is already stretched responding to urgent business demands. There is no capacity for process improvement work, and no budget to hire someone to do it. Knowledge concentration is often a resource problem, not just a discipline problem.

Clients, meanwhile, resist paying for process improvement as a line item, even though they would benefit from it in the long run. "Document your processes" does not fit neatly into a matter budget or a quarterly business review.

This is a structural problem that requires structural solutions: alternative fee arrangements, value-based pricing, or in-house teams with different incentives. Until the economic model changes, documentation will remain undervalued, regardless of how much sense it makes operationally.

The cultural barrier

There is also a cultural resistance. Legal work is seen as bespoke, intellectual, and judgement-led. Documentation feels reductive. It feels like you are turning professional work into a factory process. However, this is a false dichotomy. Documenting the routine does not eliminate judgement. It protects it. It ensures that judgement is spent on the decisions that matter, not on figuring out what the process is. While the economic model may not reward documentation today, the cost of not documenting—in terms of risk, inefficiency, and inability to scale—is becoming impossible to ignore.

A practical test

If you want to know whether a process is adequately documented, try this:

Imagine a competent lawyer joins your team tomorrow. They have the right technical skills, but no knowledge of your specific clients, precedents, or ways of working.

Could they complete a routine task to your standard using only written documentation, without asking anyone for help?

If the answer is no, your process lives in people, not in documentation.

The benefit of documentation

Documenting processes has benefits that go far beyond AI.

It makes onboarding faster. New joiners can get up to speed without relying on the availability of a busy senior lawyer.

It makes delegation more reliable. You can hand work to someone and be confident that they know what to do.

It makes quality more consistent. Everyone is following the same process, not their own interpretation of it.

It makes your team more resilient. If someone leaves, the knowledge does not leave with them, and it makes continuous improvement possible. You cannot improve a process that you have not defined.

The first step: write down what you already do

The hardest part of documentation is starting. It feels like a huge task. Where do you even begin?

The answer is: start small. Pick one routine task. Write down the steps. Write down the decision points. Write down the standard outputs.

Do not aim for perfection. Aim for "better than nothing".

Once you have a draft, test it. Give it to someone who has not done the task before and see if they can follow it. Refine it based on their feedback.

Then move on to the next task.

The reality of documentation

Documentation is harder than it sounds. Tacit knowledge is genuinely difficult to articulate—the expert often does not know what they know, or cannot easily put it into words. Time constraints are real: "document your processes" competes with "respond to this urgent board request." Knowledge changes: by the time you document a process, the law or the business context may have evolved. And documentation creates a maintenance burden: unmaintained documentation becomes worse than no documentation, because people follow outdated instructions.

These difficulties are real. They are not excuses for avoiding all documentation, but they are reasons to be realistic about the effort required and strategic about what you document.

Start with high-volume, high-value processes where the investment will pay off quickly. Accept that documentation will be imperfect. Plan for maintenance from the start—assign someone to review and update documentation quarterly. And recognise that some knowledge may genuinely resist documentation, which is fine as long as you are making deliberate choices about what to document and what to leave tacit.

Over time, you will build a library of documented processes. And you will discover that most of what felt like tacit knowledge was actually just undocumented knowledge.

The AI forcing function

AI makes the cost of undocumented processes impossible to ignore.

If you cannot explain a process to a human in writing, you cannot explain it to an AI. If your process relies on people "just knowing" what to do, it cannot be automated, but the solution, documentation, is not just about AI. It's about building a more resilient, more scalable, more professional operation.

AI is just the catalyst. The real work is making your knowledge explicit, your processes repeatable, and your team less dependent on any single person.

If a process cannot survive staff turnover, it's not a process. It's a dependency. And dependencies are risks.

See Appendix: Template 3 (Process Documentation Template) for a practical template you can use immediately.

Define work properly

The single most important step in becoming ready for AI is also the simplest: define what you actually want done.

Not what you have always asked for. Not what sounds right. What you actually want, in terms specific enough that someone who does not share your context can deliver it.

This is harder than it sounds. But it's not optional.

The illusion of clarity

Most legal instructions feel clear when you give them. "Review this contract." "Draft a response." "Advise on this issue."

But clarity is not the same as specificity. Clarity is about tone and confidence. Specificity is about boundaries and outputs.

An instruction can be clear and still be useless. "Make it better" is clear. It's also meaningless.

The components of a well-defined task

A well-defined task has five components:

1. **Input:** What are you starting with? A contract, a set of facts, a question, a precedent?
2. **Output:** What should the result look like? A memo, a marked-up document, a risk rating, a yes/no answer?
3. **Scope:** What is in scope and what is out of scope? Are you reviewing the entire contract or just the liability clauses? Are you advising on all risks or just regulatory risks?
4. **Standard:** What does good look like? What is the risk threshold? What level of detail is expected? What tone should be used?
5. **Context:** What background information is needed? What is the commercial objective? What has already been agreed? What constraints apply?

If any of these components is missing, the task is under-defined.

A practical example

Bad instruction: "Review this contract."

Better instruction: "Review this software licence agreement against our standard template. Flag any deviations in the liability, indemnity, and IP clauses. Provide a one-page summary with a recommendation on whether to escalate to the General Counsel."

The second version is not longer because it's bureaucratic. It's longer because it's specific. It tells you what to review, what to look for, and what the output should be.

It's also testable. You can check whether the task has been done correctly. You can give it to two different people and expect similar results.

Decomposing complex tasks

Not all legal work can be reduced to a simple instruction. Some tasks are genuinely complex, with multiple steps and decision points.

But complexity is not the same as vagueness. A complex task can still be well-defined.

The technique is decomposition. Break the task into smaller, more specific sub-tasks. Define each sub-task clearly. Identify the decision points where judgement is required.

For example, "advise on whether we should proceed with this acquisition" is complex. But it can be decomposed:

1. Review the target's financial statements and flag any material liabilities.
2. Assess the regulatory approval requirements in each jurisdiction.
3. Identify any IP or employment issues that could affect valuation.
4. Summarise the key risks and provide a recommendation on whether to proceed to due diligence.

Each of these sub-tasks is more specific than the original instruction. Each has a clear input, output, and scope. And the final step, the recommendation, is where judgement is applied.

The role of acceptance criteria

In software development, tasks are often defined with "acceptance criteria". These are the conditions that must be met for the task to be considered complete.

For example:

- The output must be in the agreed format.
- All deviations from the standard must be flagged.
- The response time must be under 24 hours.
- The recommendation must include a risk rating.

Acceptance criteria make expectations explicit. They remove ambiguity. They make it possible to assess whether a task has been done correctly without relying on subjective judgement.

Internal legal instructions often lack explicit acceptance criteria. But they should have them.

Bounding responsibility

One of the hardest parts of defining work is deciding what is in scope and what is not.

Legal work has a tendency to expand. A simple contract review becomes a full risk assessment. A quick question becomes a ten-page memo. This is often well-intentioned. The lawyer wants to be thorough. They want to add value.

Some legal work is genuinely unbounded, but unbounded work is expensive, unpredictable, and hard to manage.

Defining work properly means being explicit about boundaries. If the task is to review liability clauses, it's not to review the entire contract. If the task is to provide a preliminary view, it's not to conduct a full legal analysis.

This does not mean doing less work. It means doing the right work, and being clear about what that is.

The benefit of constraints

Constraints are often seen as limiting. But in practice, they are liberating.

If you know exactly what is expected, you can focus. You do not waste time second-guessing whether you have done enough. You do not gold-plate the work because you are not sure what "good enough" looks like.

Constraints also make delegation easier. If the task is well-defined, you can hand it to someone less experienced and be confident in the result, and they make feedback easier. If the output does not meet the acceptance criteria, you can point to the gap. If it does, you can move on.

The engineering parallel: user stories

In agile software development, work is defined using "user stories". A user story has a standard format:

"As a [user], I want [goal], so that [benefit]."

For example: "As a contract manager, I want to see all contracts expiring in the next 90 days, so that I can plan renewals."

This format forces clarity. It makes you think about who the work is for, what they need, and why they need it.

Legal work could benefit from the same discipline. Instead of "Review this contract", you could say:

"As the General Counsel, I need to know whether this contract exposes us to liability beyond our agreed risk threshold, so that I can decide whether to escalate to the Board."

This is not bureaucratic but precise. And precision is what makes work manageable.

Starting small

Defining work properly feels like a big change. It feels like it will slow you down, but you do not have to do it all at once. Start with one type of task. Define it clearly. Test it. Refine it.

Then move on to the next task.

Over time, you will build a library of well-defined tasks. And you will discover that the time spent defining the work is more than recovered in the time saved on rework, clarification, and misunderstanding.

The AI benefit

Well-defined tasks are essential for AI. But they are also essential for humans.

They make delegation more reliable, training faster, quality more consistent, and work more predictable.

AI is just the thing that makes the cost of poorly-defined work impossible to ignore. But the solution, clarity, is good for you anyway.

If you cannot define a task clearly enough for a junior lawyer to complete it without asking questions, you cannot define it clearly enough for an AI. And if you cannot define it clearly enough for an AI, you probably have not defined it clearly enough for yourself.

Write down assumptions

Every piece of legal advice rests on assumptions: about the facts, the law, what the client wants, and what risks are acceptable.

Most of these assumptions are never written down.

They exist in the lawyer's head, shaped by experience, by previous conversations, by the culture of the team. They are transmitted through context, not documentation.

This works, until it does not.

The invisible foundation

Assumptions are the foundation of legal work—they define the boundaries of the advice, determine what is relevant, and shape the risk assessment.

But because they are often implicit, they are also fragile. If the assumption changes, the advice may no longer be valid. But if the assumption was never written down, no one knows to revisit the advice.

A lawyer advises that a contract is acceptable based on the assumption that the client's risk appetite is moderate. Six months later, the client's risk appetite changes. The contract is still in the file, marked as "approved". But the approval was conditional on an assumption that no longer holds.

If the assumption had been written down, someone would know to revisit the advice. But it was not, so they do not.

The cost of implicit assumptions

Implicit assumptions create several problems:

- **Misunderstanding:** Different people make different assumptions. If those assumptions are not explicit, the advice will be inconsistent.
- **Fragility:** If the assumption changes, the advice may no longer be valid. But if no one knows what the assumption was, no one knows to update the advice.
- **Inefficiency:** People waste time re-deriving assumptions that should have been documented.

- **Risk:** Advice that rests on unstated assumptions is harder to defend if challenged.

What to write down

Not every assumption needs to be documented. But the following should be:

- **Factual assumptions:** What facts are you relying on? What has the client told you? What have you assumed to be true?
- **Legal assumptions:** What is your interpretation of the law? What precedents are you relying on? What uncertainties exist?
- **Commercial assumptions:** What is the client's objective? What is their risk appetite? What constraints apply?
- **Scope assumptions:** What is in scope and what is out of scope? What questions are you answering and what questions are you not?

Writing these down does not mean you are uncertain. It means you are being clear about the foundation of your advice.

A practical example

Bad advice: "This contract is acceptable."

Better advice: "This contract is acceptable on the following assumptions:

- The client's risk appetite is moderate (as discussed in the meeting on 15 January).
- The counterparty is a reputable organisation with no history of disputes.
- The liability cap of £1m is consistent with the value of the contract.
- The governing law is English law, which we are familiar with.

If any of these assumptions change, the advice should be revisited."

The second version is not longer because it's defensive. It's longer because it's clear. It makes the foundation of the advice explicit.

It's also maintainable. If the client's risk appetite changes, someone reading the file will know to revisit the advice.

The discipline of articulation

Writing down assumptions forces you to articulate them. And articulation forces clarity.

Often, when you try to write down an assumption, you realise that it's not as clear as you thought. Or that it's not actually an assumption, but a guess. Or that it's an assumption you should test, not accept.

This discipline is valuable because it makes your advice more robust, your thinking more rigorous, and your work more defensible.

The engineering parallel: design decisions

In software engineering, major design decisions are often documented in "architecture decision records" (ADRs). These are short documents that explain:

- What decision was made.
- What alternatives were considered.
- What assumptions were made.
- What the consequences are.

The purpose is not to justify the decision. The purpose is to make the reasoning explicit, so that future developers understand why the system is the way it's.

Legal work would benefit from the same discipline. Not for every piece of advice, but for significant decisions.

Why did we accept this risk? What alternatives did we consider? What assumptions did we make? What would cause us to revisit this?

Assumptions and AI

AI cannot make assumptions. It can only work with what you give it.

If you ask an AI to review a contract, it will apply some default assumptions. But those assumptions may not match yours. And if you have not written down what your assumptions are, you will not know whether the AI's output is valid.

Writing down assumptions makes AI more reliable. It gives the system the context it needs to produce useful output.

It also makes your own work more reliable. Even if you never use AI, explicit assumptions make your advice clearer, more maintainable, and defensible.

The limits of documentation

Risk tolerance, in particular, is genuinely context-dependent in ways that are hard to document in advance. A GC's risk tolerance for a liability cap might legitimately depend on the specific counterparty's financial strength, the broader commercial relationship, the current regulatory environment, the company's financial position that quarter, whether the deal is strategic or tactical, and a dozen other factors that cannot be known when you write the documentation.

These factors are real and legitimate. They are not "poor process." They are the exercise of informed judgment in context.

The goal is not to eliminate this judgment. The goal is to document the **framework** for making risk decisions. Document the factors that should be considered. Document the questions that should be asked. Document the escalation thresholds. The final decision may still require judgment, but the process for reaching that decision should be clear.

This is the difference between "our risk threshold for liability caps is £5 million" (which is too rigid and will be wrong in many contexts) and "when assessing liability caps, consider: counterparty financial strength, strategic importance of relationship, regulatory exposure, our insurance coverage, and precedent with similar counterparties. Escalate to GC if cap exceeds £5 million or if any of these factors create unusual risk" (which is a framework that preserves judgment while making the decision process explicit).

The cultural shift

Writing down assumptions requires a cultural shift. It requires accepting that not everything is certain. That advice is conditional. That the foundation of your reasoning is worth documenting.

This challenges tradition. There is a tradition in legal of presenting advice with confidence. Of not showing your working. Of giving the answer, not the reasoning.

But this tradition is not serving the profession well. It makes advice harder to challenge, harder to update, and harder to learn from.

Writing down assumptions is not a sign of weakness but a sign of rigour.

Starting small

You do not need to document every assumption in every piece of advice. Start with the significant ones.

When you give advice that rests on a key factual or commercial assumption, write it down. When you make a judgement call on a point of law, note the reasoning.

Over time, this will become a habit. And you will discover that the discipline of writing down assumptions makes your thinking clearer and your advice more robust.

The benefit beyond AI

Explicit assumptions make legal work better, regardless of whether you use AI.

They make advice more maintainable, handovers easier, quality review more effective, and your work more defensible.

They also make learning easier. Junior lawyers can see not just the advice, but the reasoning behind it. They can understand the assumptions that shaped the decision.

AI is just the thing that makes the cost of implicit assumptions impossible to ignore. But the solution, documentation, is good for you anyway.

Assumptions exist whether you write them down or not. Writing them down just makes them visible, testable, and maintainable.

See Appendix: Template 2 (Assumption Documentation Template) for a practical template you can use immediately.

Standardise language, not thinking

One of the most common objections to standardisation in legal work is that it will stifle creativity. That it will reduce lawyers to box-tickers. That it will eliminate the judgement and nuance that makes legal work valuable.

This objection misunderstands what standardisation is for.

Standardisation is not about making every contract identical. It's not about eliminating judgement. It's about creating a stable foundation so that judgement can be applied effectively.

The goal is to standardise language, not thinking.

The cost of linguistic drift

Legal language drifts. The same concept is described in different ways. "Confidential information" in one contract, "proprietary information" in another. "Material breach" with no definition in one place, a detailed definition in another.

This drift is usually accidental. Different lawyers have different habits, different precedents use different terms and just over time, the language diverges.

The problem is that linguistic drift creates ambiguity. If you use different terms for the same concept, it's not clear whether you mean the same thing. If you use the same term with different definitions, it's not clear which definition applies.

This ambiguity is expensive—it makes contracts harder to review, precedents less useful, and automation much harder.

The difference between substance and style

Standardising language does not mean standardising substance.

If you agree that "confidential information" is the term you will use, you are not agreeing on what should be confidential. You are just agreeing on what to call it.

If you agree on a standard structure for a liability clause, you are not agreeing on what the liability cap should be. You are just agreeing on where the cap will be stated and how it will be expressed.

The substance, the risk position, the commercial terms, the judgement calls, remains flexible. The language is just the container.

The benefit of a shared vocabulary

A shared vocabulary makes communication faster and more reliable.

If everyone in your team uses the same term for the same concept, you do not have to spend time clarifying what you mean. You do not have to check whether "material breach" in this contract means the same thing as "material breach" in that contract.

A shared vocabulary also makes precedents more useful. If your templates use consistent terminology, you can reuse clauses with confidence. You do not have to adapt the language every time, and it makes automation possible. If your contracts use consistent terms, you can search for them, extract them, and analyse them reliably.

The engineering parallel: naming conventions

In software engineering, naming conventions are standard practice. Variables, functions, and files follow agreed patterns. This is not because programmers lack creativity. It's because consistency makes code easier to read, easier to maintain, and easier to debug.

A good naming convention does not constrain what the code does. It just makes it easier to understand what it does.

Legal work would benefit from the same discipline. Not because lawyers lack creativity, but because consistency makes legal work easier to read, easier to review, and easier to manage.

What to standardise

You do not need to standardise everything. But the following are good candidates:

- **Defined terms:** Agree on a standard set of terms for common concepts. Use them consistently.
- **Clause structure:** Agree on a standard structure for common clauses. Where does the liability cap go? How is the indemnity worded?
- **Document structure:** Agree on a standard order for clauses. Definitions first, then operative provisions, then boilerplate.

- **Risk language:** Agree on how to describe risk. What does "high risk" mean? What does "material" mean?

This is not a straitjacket but a foundation. You can deviate when there is a good reason. But the deviation should be intentional, not accidental.

Templates as defaults

A good template is not a constraint. It's a default.

It says "This is the standard wording, the standard structure, the standard risk position. If you need to change it, you can. But you should know that you are changing it, and you should have a reason."

This makes deviations visible. It makes them intentional. It makes them easier to review, because the reviewer is not trying to figure out whether the difference is meaningful or accidental, and it makes automation possible, because the system can learn the default and flag the deviations.

The resistance to standardisation

Standardisation is often resisted because it feels reductive. It feels like you are turning professional work into a factory process.

All of this is a false dichotomy. Standardising the routine does not eliminate the complex, it protects it.

If you do not have to spend time reconciling inconsistent terminology, or figuring out which version of a clause to use, you have more time for the work that genuinely requires judgement.

Standardisation is not about reducing legal work to a checklist, it's about reducing the friction so that you can focus on the work that matters.

The cultural benefit

Standardisation also has a cultural benefit: it makes expectations clear, reduces the cognitive load of routine decisions, and makes onboarding faster.

A new joiner does not have to learn the stylistic preferences of every senior lawyer. They can learn the standard, and then learn the exceptions.

This makes the team more cohesive, the work more predictable, and quality more consistent.

Starting small

You do not need to standardise everything at once. Start with the most common terms and clauses.

Pick one defined term that is used inconsistently. Agree on a standard definition. Use it in all new contracts.

Pick one clause that appears in most of your contracts. Agree on a standard structure. Make it the default in your template.

Over time, you will build a library of standardised language and you will discover that the consistency makes your work faster, clearer, and more reliable.

The AI benefit

Standardised language is essential for AI, but it's also essential for humans.

It makes contracts easier to read, precedents more useful, review faster, and automation possible.

AI is just the thing that makes the cost of linguistic drift impossible to ignore. But the solution, standardisation, is good for you anyway.

Standardise the language, not the thinking. Standardise the container, not the content. Standardise the foundation, so that judgement can be applied effectively.

Consistency enables judgement. It does not replace it.

See Appendix: Template 5 (Standardisation Decision Log) for a practical template you can use immediately.

Feedback as professional hygiene

Legal advice is often a one-way transaction. The lawyer gives the advice. The client acts on it. And then... nothing.

Feedback loops are often absent or informal. The lawyer may not find out whether the advice was useful, whether it was clear, whether it was acted on, or whether the outcome was what the client expected.

This absence of feedback is a problem. Because without feedback, you cannot improve.

The missing signal

In most professions, feedback is built in. A doctor sees whether the treatment worked. An engineer sees whether the bridge stands. A teacher sees whether the student understood.

But legal advice often lacks this signal. The advice is given, and then the lawyer moves on to the next matter. They do not find out whether the contract was signed, whether the risk materialised, whether the client was satisfied.

This is not because lawyers do not care. It's because legal feedback is structurally harder to capture than in other professions. Much legal work is preventative—the goal is to avoid problems, which means success is often invisible. Many matters take years to resolve, making timely feedback impossible. And in many cases, the best outcome is that nothing happens: no litigation, no dispute, no regulatory action. The absence of problems is the goal, but it provides no feedback signal.

There are other structural barriers. Clients are often bound by confidentiality and cannot tell you how things turned out. Relationship dynamics make it awkward to criticise your lawyer when you need them again next week. And there is no counterfactual—you cannot know what would have happened with different advice. How do you know if your advice was "good" when you cannot test the alternative?

These barriers are real, not excuses. They make outcome feedback (was the advice correct?) genuinely difficult to obtain. The solution is to focus on **process feedback** (was the advice clear? timely? useful? actionable?) which is possible to capture even when outcome feedback is not.

Despite these structural challenges, even process feedback is often not built into the workflow where it could be.

Why feedback matters

Feedback is how you learn. It's how you calibrate your judgement. It's how you discover that your assumptions were wrong, or that your advice was unclear, or that the client needed something different.

Without feedback, you are operating blind. You do not know whether your advice is hitting the mark. You do not know whether your risk assessments are accurate. You do not know whether your communication style is effective.

You can be confident, experienced, and still be systematically wrong about something, because no one has ever told you.

The illusion of expertise

Expertise without feedback is fragile. It's based on assumptions that may or may not be true.

One thing that still strikes me when I move between engineering and legal teams is how differently feedback works. In engineering, feedback is relentless. You push a change and within minutes something tells you whether it was a good idea. Tests fail, monitoring spikes, alerts fire. In legal teams, I have often seen the opposite. Work goes out the door, sometimes at huge effort and cost, and then silence. No structured way of knowing whether the advice was useful, whether the clause was reused, whether the risk actually materialised. I once sat in a session where a team wanted to deploy an AI review tool. When I asked how they currently knew whether a review was good... eventually someone said, "We know when it's wrong." Engineers would never accept a system where correctness is only visible in failure.

A lawyer may believe that their advice is clear, because no one has ever told them it's not. They may believe that their risk assessments are accurate, because they have never tracked the outcomes. They may believe that their clients are satisfied, because no one has complained, but absence of complaint is not the same as presence of quality. It may just mean that the client does not feel comfortable giving feedback, or that they have learned to work around the problem.

The engineering parallel: testing and monitoring

In software engineering, feedback is continuous. Code is tested. Systems are monitored. Errors are logged. Performance is measured.

This is not because engineers are more rigorous than lawyers. It's because the cost of failure is immediate and visible. If the code does not work, the system breaks. If the system is slow, users complain.

Much legal work lacks this immediacy. The cost of poor advice may not be visible for months or years. By the time it becomes clear that the advice was wrong, the lawyer has moved on, but this does not mean feedback is impossible. It just means it has to be intentional.

What to measure

Not everything can be measured. But some things can, and should be:

- **Clarity:** Did the client understand the advice? Did they have to ask for clarification?
- **Usefulness:** Did the advice help the client make a decision? Was it actionable?
- **Accuracy:** Did the risk materialise? Was the legal position as predicted?
- **Timeliness:** Was the advice delivered when the client needed it?
- **Tone:** Was the advice appropriately calibrated to the client's risk appetite and commercial context?

These are not metrics for a performance review. They are signals for learning.

Light feedback is better than no feedback

Feedback does not have to be formal. It does not have to be a survey or a scorecard. It can be as simple as a follow-up question.

"Did that advice make sense?"

"Was there anything you needed that I did not cover?"

"How did the negotiation go?"

These questions take seconds to ask. But they provide valuable information.

They tell you whether your advice was clear. Whether it was useful. Whether the client's needs were met.

And over time, they help you calibrate your judgement.

The cultural barrier

Legal culture does not encourage feedback. There is a tradition of deference. Clients do not challenge lawyers. Junior lawyers do not challenge senior lawyers. Advice is given with authority, and questioning it feels inappropriate.

This culture is not serving the profession well. It creates an environment where mistakes are not surfaced, where assumptions are not tested, and where learning is slow.

Breaking this culture requires deliberate effort. It requires creating safe channels for feedback. It requires asking for feedback, not waiting for it. It requires treating feedback as a gift, not a criticism.

Feedback and AI

AI makes the absence of feedback even more costly.

If you deploy an AI system without a feedback loop, you have no way of knowing whether it's working. You do not know whether the outputs are useful. You do not know whether the system is drifting. You do not know whether the assumptions it was trained on are still valid.

Feedback is how AI systems improve. They learn from corrections. They adapt to new patterns. But if no one is providing feedback, the system cannot improve.

The same is true for humans. Without feedback, you cannot improve. You can only repeat what you have always done, and hope it's still working.

Building a feedback loop

Building a feedback loop does not require a complex system. It just requires intention.

Start small. Pick one type of advice. After you give it, follow up. Ask whether it was useful. Ask whether anything was unclear. Ask whether the outcome was as expected.

Track the responses. Look for patterns. If multiple clients say the advice was unclear, that is a signal. If the outcomes are consistently different from your predictions, that is a signal.

Use those signals to adjust. To clarify your communication. To recalibrate your risk assessments. To refine your process.

Over time, this will become a habit. And you will discover that feedback is not a burden. It's a tool.

The benefit beyond AI

Feedback makes legal work better, regardless of whether you use AI.

It makes advice more useful. It makes communication clearer. It makes risk assessments more accurate. It makes clients more satisfied.

It also makes the work more satisfying. Lawyers want to do good work. But without feedback, they do not know whether they are succeeding.

Feedback closes the loop. It tells you whether your work is hitting the mark. And it gives you the information you need to improve.

The professional responsibility

Feedback is not just good practice. It's a professional responsibility.

If you are giving advice that affects significant decisions, you have a responsibility to know whether that advice is sound. You cannot know that without feedback.

If you are deploying AI systems that affect legal outcomes, you have a responsibility to know whether those systems are working. You cannot know that without feedback.

Feedback is not optional. It's hygiene. It's the minimum standard for professional practice.

See Appendix: Template 4 (Feedback Loop Template) for a practical template you can use immediately.

AI is just the thing that makes the cost of operating without feedback impossible to ignore. But the solution, building a feedback loop, is good for you anyway.

No system improves without feedback. Not AI. Not humans. Not legal teams.

If you want to get better, you need to know how you are doing. And the only way to know is to ask.

Process is infrastructure

The word "process" has a bad reputation in legal. It sounds bureaucratic. It sounds like box-ticking. It sounds like the opposite of professional judgement.

But this is a misunderstanding of what process is for.

Process is not overhead but infrastructure—the foundation that makes everything else possible.

The cost of no process

In the absence of process, every task is bespoke. Decisions are made from scratch. Work depends on the availability and expertise of specific individuals.

This feels flexible. It feels responsive. It feels like you are treating every matter with the care it deserves.

But it's also fragile—it does not scale, survive staff turnover, or adapt to change, and it becomes expensive, unpredictable, and exhausting.

Without process, you cannot delegate reliably, train efficiently, measure quality, or improve systematically.

You are dependent on hero lawyers who compensate for the absence of structure. And when those lawyers leave, the knowledge leaves with them.

What process actually is

Process is not bureaucracy but repeatability.

Process answers the question: "How do we do this type of work?" It documents the steps, standard inputs, expected outputs, decision points, and escalation triggers. This documentation lets you delegate work confidently, assess whether outputs meet the standard, and improve systematically by measuring what works and adjusting what doesn't.

The engineering parallel: CI/CD

In software engineering, there is a concept called "continuous integration and continuous deployment" (CI/CD). It's a set of automated processes that ensure code is tested, reviewed, and deployed reliably.

Before CI/CD, deployment was manual. It was error-prone. It depended on the expertise of specific engineers. It was slow, risky, and stressful.

CI/CD did not eliminate judgement. Engineers still decide what to build and how to build it but it eliminated the manual, error-prone steps that surrounded that judgement.

The result is that teams can deploy more frequently, with higher confidence, and with less stress.

Legal work needs the same discipline. Not because lawyers are less capable than engineers, but because the absence of process creates the same problems: unpredictability, fragility, and dependence on individuals.

Process as enabler, not constraint

The objection to process is usually that it will slow things down. That it will add bureaucracy. That it will prevent you from responding to the specific needs of each matter.

But good process does the opposite. It speeds things up, because you do not have to reinvent the wheel every time. It reduces bureaucracy, because the routine steps are documented and repeatable. It makes you more responsive, because you can focus on the genuinely bespoke elements, not on figuring out the basics.

Process is not a constraint but a platform—the stable foundation for moving faster and more confidently.

What to process

Not everything needs a formal process. But the following should be:

- **Routine tasks:** Contract review, NDA approval, standard advice requests. If you do it more than once a month, it should have a process.
- **Handovers:** What happens when a matter is handed from one lawyer to another? What information is transferred? What is the checklist?
- **Quality review:** How is work reviewed? What is the standard? Who reviews what?

- **Escalation:** When does a matter get escalated? To whom? What information is provided?
- **Client communication:** What is the expected response time? What is the standard format for advice?

These are not bureaucratic exercises. They are infrastructure. They are the things that make your team reliable, scalable, and resilient.

The discipline of documentation

Process only works if it's documented. If the process exists only in people's heads, it's not a process but a dependency.

Documentation does not have to be elaborate. It can be a checklist, a flowchart, a one-page guide. The format does not matter. What matters is that it's written down, accessible, and maintained.

Good documentation answers the question: "If I have never done this before, can I follow this process and get the right result?"

If the answer is no, the documentation is not good enough.

Process and flexibility

Process does not eliminate flexibility. It makes flexibility intentional.

If you have a standard process, you can deviate from it when there is a good reason, but the deviation is visible—a conscious choice rather than an accident.

This is important. It means that when you look back at a matter, you can see where the standard was followed and where it was not. You can assess whether the deviation was justified. You can learn from it.

Without a standard, there is no deviation. There is just chaos.

The benefit of predictability

Predictability is undervalued in legal work. There is a belief that every matter is unique, and therefore unpredictability is inevitable.

But most unpredictability is not inherent to the work but rather a result of poor process.

If you have a standard process for contract review, you can predict how long it will take, what the output will look like, and what the quality will be. If you do not, every contract review is a surprise.

Predictability makes planning possible. It makes delegation reliable. It makes clients happier, because they know what to expect, and it makes your work less stressful, because you are not constantly firefighting.

Process and AI

AI depends on process. If you cannot describe a process, you cannot automate it.

Process also makes AI more effective. If you have a standard process, you can identify the steps that are routine and the steps that require judgement. You can automate the routine and reserve human effort for the judgement.

Without process, you cannot make this distinction. You cannot identify what to automate, because you do not know what the standard is.

Starting small

Building process feels like a big task. It feels like it will take time and effort that you do not have, but you do not have to do it all at once. Start with one routine task. Document the steps. Test it. Refine it.

Then move on to the next task.

Over time, you will build a library of processes. And you will discover that the time spent building process is more than recovered in the time saved on rework, clarification, and firefighting.

The cultural shift

Building process requires a cultural shift. It requires accepting that not everything is bespoke. That repeatability is valuable. That documentation is not bureaucracy.

This challenges professional identity. Legal culture values craft, judgement, and individual expertise. Process can feel like it's reducing professional work to a factory operation.

But this is a false dichotomy. Process does not eliminate craft. It protects it. It ensures that craft is spent on the work that genuinely requires it, not on compensating for the absence of structure.

The professional responsibility

Process is not just good practice but a professional responsibility.

If you are giving advice that affects significant decisions, you have a responsibility to ensure that the advice is reliable, consistent, and defensible. You cannot do that without process.

If you are managing a team, you have a responsibility to ensure that the work is predictable, scalable, and resilient. You cannot do that without process.

Process is not overhead but capacity—it lets you do more, with higher quality, and with less stress.

AI is just the thing that makes the cost of operating without process impossible to ignore. But the solution, building infrastructure, is good for you anyway.

Process is not the opposite of judgement but the foundation that makes judgement effective.

See Appendix: Template 3 (Process Documentation Template) for a practical template you can use immediately.

Where AI should not be used

The most important decision in AI adoption is not what to automate. It's what not to automate.

There are categories of legal work where AI is not appropriate. Not because the technology is not good enough, but because the work itself should not be delegated to a machine.

Understanding these boundaries is not a sign of caution but a sign of maturity.

High-stakes, low-volume judgement

Some legal decisions are genuinely one-off. They are strategic, context-heavy, and consequential. They require deep understanding of the client's business, the commercial landscape, and the broader legal and regulatory environment.

These decisions should not be automated.

Examples include:

- Whether to proceed with a major acquisition.
- How to respond to a regulatory investigation.
- Whether to settle or litigate a high-value dispute.
- How to structure a novel transaction in an uncertain regulatory environment.

These are not tasks where you want speed or consistency. They are tasks where you want the best human judgement you can get.

AI can support these decisions. It can surface relevant information, identify risks, and provide analysis. But the decision itself should be human.

Work where the cost of error is unacceptable

Some legal work has asymmetric risk. The cost of getting it wrong is catastrophic. The benefit of getting it right is just doing your job.

In these contexts, AI should not be used unless the system is provably reliable, and even then, only with human oversight.

Examples include:

- Court filings where an error could result in professional sanctions.
- Advice on matters where the client's liberty or fundamental rights are at stake.
- Regulatory submissions where non-compliance could result in criminal liability.

The problem is not the frequency of errors. The problem is that when AI does get these wrong, the consequences are unacceptable.

Work that requires genuine creativity

Legal work is not usually creative in the artistic sense. But some tasks do require genuine originality. They require you to see a problem in a new way, to construct an argument that has not been made before, to design a structure that does not yet exist.

AI is not good at this. It's good at pattern-matching, at applying known solutions to known problems. It's not good at inventing new solutions.

Examples include:

- Developing a novel legal argument in a case of first impression.
- Structuring a transaction in a way that has not been done before.
- Drafting legislation that balances competing policy objectives.

AI can assist with research, with drafting, with analysis. But the creative insight, the thing that makes the work valuable, should be human.

Work where the client relationship is the value

Some legal work is not primarily about the legal output. It's about the relationship. The trust. The understanding of the client's business and their concerns.

In these contexts, automating the work misses the point. The value is often in understanding unspoken concerns, reading organisational dynamics, knowing what not to say, and understanding the political context that shapes what the client can actually do. This tacit knowledge of how the client's organisation works, who has real authority, what battles have already been fought, what constraints are not being articulated, cannot be documented and cannot be replicated by AI.

Examples include:

- Strategic advice to a long-standing client where the value is understanding their unspoken constraints.
- Sensitive negotiations where tone, timing, and reading the room matter as much as the legal position.
- Board-level counsel where the value is in the conversation, the ability to sense what the board is really worried about, and knowing what issues to surface versus what to let pass.

AI can prepare materials. It can draft documents. It can surface information. But it cannot replace the human relationship or the tacit organisational knowledge that makes senior counsel valuable.

Work where explainability is essential

Some legal advice must be explainable. Not just "here is the answer", but "here is why this is the answer, and here is the reasoning that supports it".

AI explanations can be unreliable. While many systems can produce an output and attempt to explain their reasoning, those explanations may not meet legal standards for transparency.

If the client, or a court, or a regulator needs to understand the reasoning, AI should not be the sole source of the advice.

Regulatory and professional conduct constraints

Professional conduct rules and regulatory requirements create genuine constraints on AI adoption. Lawyers have a duty to provide advice tailored to specific circumstances. Over-standardisation or over-reliance on automated outputs could be seen as failing this duty. Some regulators explicitly require "fresh eyes" human review for certain types of work. Some jurisdictions have rules about what work can be delegated to non-lawyers or to systems.

These are not theoretical concerns. They are real constraints that must be navigated thoughtfully. They do not mean AI cannot be used, but they do mean you need to understand the regulatory environment in your jurisdiction and practice area. What is permissible for contract review may not be permissible for court filings. What is acceptable for in-house counsel may not be acceptable for external advisors.

These constraints are not excuses for avoiding all AI adoption. They are reasons to be deliberate about where and how you use AI, and to ensure that human oversight and professional judgment remain in the loop where required.

The discipline of saying no

Saying no to AI is harder than it sounds. There is immense pressure to adopt, with pressure to be seen as innovative and pressure to reduce costs.

But adopting AI in the wrong context is worse than not adopting it at all. It creates risk. It damages trust. It undermines the credibility of AI in the contexts where it could be useful.

The discipline of saying no is a sign of maturity. It shows that you understand the technology. That you understand your work. That you are making deliberate choices, not following hype.

The test: would you delegate this to a junior?

A simple test for whether AI is appropriate: would you delegate this task to a competent but inexperienced junior lawyer?

If yes, AI may be appropriate with similar oversight. Most junior lawyer work is reviewed, and AI can work in the same model: the AI drafts, the human reviews. The key question is whether the task can be broken down into steps that can be verified, or whether it requires end-to-end judgement that cannot be meaningfully reviewed.

If you would not delegate the task even with review—because it requires deep strategic judgement, carries unacceptable risk, or depends fundamentally on relationships and tacit knowledge, then AI is probably not appropriate either.

AI can assist. But the human should remain in control.

The benefit of restraint

Restraint in AI adoption has several benefits.

It protects you from high profile failures. It preserves trust with clients. It ensures that AI is used in contexts where it adds value, not where it creates risk.

It also makes your AI adoption more credible. If you are selective about where you use AI, clients are more likely to trust that you have thought it through.

Saying "We use AI for contract review, but not for strategic advice" is more credible than saying "We use AI for everything".

The evolving boundary

The boundary between what AI can and cannot do is not fixed, it's constantly evolving.

Tasks that were too complex for AI five years ago are routine today. Tasks that are too risky today may be safe tomorrow.

But the principle remains: AI should be used where it adds value and where the risk is acceptable. Not everywhere.

The goal is not to maximise AI adoption. The goal is to use AI well.

The human responsibility

Deciding where to use AI is a human responsibility. It cannot be delegated to the technology. It cannot be delegated to the vendor.

It requires judgement. It requires understanding of the work, the risks, and the client's needs.

This is not a technical decision. It's a professional decision and it's one of the most important decisions you will make.

AI is a tool. Like any tool, it's appropriate for some tasks and not for others. The skill is in knowing the difference.

From tools to systems

The most common mistake in AI adoption is treating it as a tool problem.

A law firm or legal department sees a demo of an AI contract review system. It looks impressive. They buy it. They roll it out. And then... nothing happens.

The tool sits unused. Or it's used badly. Or it produces outputs that no one trusts.

The problem is not the tool. The problem is that the tool was introduced into a system that was not ready for it.

Tools reflect systems

A tool is only as good as the system it operates within.

If your contract review process is inconsistent, an AI contract review tool will be inconsistent. If your precedents are messy, the AI will produce messy outputs. If your instructions are vague, the AI will produce vague results.

The tool does not fix the system. It reflects it.

This is why AI pilots often fail. The technology works. But the underlying system is broken, and the tool just makes the brokenness more visible.

The procurement theatre

There is a pattern in legal AI adoption that I have seen many times.

A senior leader attends a conference. They see a demo. They are impressed. They instruct the team to "explore AI".

The team runs a procurement process. They evaluate vendors. They negotiate contracts. They run a pilot.

The pilot produces mixed results. The team concludes that "AI is not ready for our work". The project is shelved.

This is procurement theatre. It gives the appearance of progress without addressing the underlying problem.

The problem is not that AI is not ready. The problem is that the organisation has not done the work to be ready for AI.

What readiness looks like

An organisation that is ready for AI has:

- **Defined tasks:** They know what they want the AI to do. They have clear inputs, outputs, and success criteria.
- **Standardised processes:** They have consistent precedents, templates, and workflows.
- **Documented knowledge:** The rules, assumptions, and preferences are written down, not locked in people's heads.
- **Feedback loops:** They have a way to measure whether the AI is working and to improve it over time.
- **Realistic expectations:** They understand what AI can and cannot do. They are not expecting magic.

If these foundations are not in place, buying an AI tool will not help. It will just expose the gaps.

The systems-first approach

The alternative to procurement theatre is a systems-first approach.

Instead of starting with the tool, start with the system. Identify the workflow you want to improve. Define the tasks. Standardise the inputs. Document the process.

Then, and only then, ask whether AI can help.

Often, you will discover that just doing this work, clarifying the tasks, standardising the process, delivers most of the benefit. The AI is just the final step.

The engineering parallel: frameworks without architecture

In software engineering, there is a common mistake: buying a framework before you have an architecture.

A team hears about a new framework. It promises to make development faster, easier, more scalable. They adopt it and then they discover that the framework does not fit their architecture. Or worse, they do not have an architecture. They have a collection of scripts held together with hope.

The framework does not fix this, it just makes the problem more visible.

The solution is not to find a better framework. The solution is to build an architecture first. To define the components, the interfaces, the data flows. To make the system coherent.

Then the framework can help, but without the architecture, it's just another layer of complexity.

Legal AI is the same. The tool is not the solution. The system is the solution. The tool just makes the system more efficient.

AI amplifies strengths and weaknesses

AI does not fix broken processes. It amplifies them.

If your process is clear, consistent, well-documented then AI will make it faster and more scalable.

If your process is vague, inconsistent, undocumented then AI will make it worse. It will produce unreliable outputs. It will create confusion. It will undermine trust.

This is not a flaw in AI but a feature—AI is a mirror that shows you what you have built.

The cultural challenge

The systems-first approach requires a cultural shift.

It requires accepting that the problem is not the technology but the organisation.

It requires investing time and effort in unglamorous work: documenting processes, standardising templates, writing down assumptions.

It requires resisting the temptation to buy a tool and hope it solves the problem.

This is hard and not exciting work that does not come with a press release. But it's the only way to make AI work.

The client expectations challenge

Client expectations create an additional layer of complexity, particularly for external law firms. Some clients prohibit AI use contractually. Some require disclosure and consent. Some have not articulated a position but are nervous about it.

These are real constraints that must be navigated. The solution is not to avoid the conversation, but to have it proactively. Explain what AI will and will not do. Explain the oversight. Explain the benefits. Clients who understand that AI is being used to improve consistency and reduce errors, not to replace judgement, are often more comfortable than clients who discover AI use after the fact.

For in-house teams, the conversation is internal but no less important. Business stakeholders may have concerns about AI that need to be addressed before deployment.

The vendor relationship

Vendors will tell you that their tool is the solution. That is their job.

But a good vendor will also ask you about your processes, your data, your readiness. They will tell you if you are not ready. They will help you prepare.

A bad vendor will sell you the tool regardless. They will take your money, run a pilot, and leave you with a system that does not work.

The difference is not the technology but whether the vendor understands that AI is a systems problem, not a tool problem.

Starting small

You do not need to fix everything before you start using AI. But you do need to fix the specific workflow you are trying to automate.

Pick one task. Define it clearly. Standardise the process. Document the rules. Build the feedback loop.

Then introduce AI. Test it. Measure it. Refine it.

Once that works, move on to the next task.

This is slower than buying a tool and rolling it out across the organisation. But it's also more likely to succeed.

The long-term view

AI adoption is not a project. It's a transformation.

It requires changing how you define work, document knowledge, measure quality, and train your team.

This takes time. It takes effort. It takes leadership.

But the result is not just better AI. The result is a better organisation. One that is more efficient, more scalable, more resilient.

AI is just the catalyst. The real work is building the system.

The honest question

Before you buy an AI tool, ask yourself this:

"If we introduced this tool tomorrow, would our processes, our data, and our people be ready to use it effectively?"

If the answer is no, do not buy the tool. Fix the system first.

If the answer is yes, the tool will help. But even then, the tool is not the solution. The system is the solution. The tool just makes it faster.

Buying AI without fixing workflows guarantees disappointment. Not because the AI is bad, but because the system is not ready.

Fix the system. Then add the tool.

See Appendix: Template 6 (AI Readiness Checklist) for a practical checklist you can use immediately.

A better set of questions

We started this book with a question: "When will AI be good enough for legal?"

That question is not wrong. But it's incomplete. It frames the challenge as a technology problem, when the real challenge is organisational.

If you want to make progress with AI, you need better questions. Questions that focus on readiness, not capability. Questions that focus on your work, not the technology.

Here are the questions you should be asking instead.

Is this task defined well enough to automate?

Before you ask whether AI can do a task, ask whether you have defined the task clearly enough for anyone, human or machine, to do it reliably.

Can you describe the inputs, the outputs, the scope, and the success criteria? Can you write down the rules? Can you specify what good looks like?

If you cannot, the problem is not AI. The problem is that the task is under-specified.

Fix the specification first. Then ask whether AI can help.

Where does judgement truly add value?

Not all legal work requires judgement. Some of it's routine. Some of it's compensatory, filling in gaps left by poor process.

Before you deploy AI, ask: where in this workflow does judgement genuinely add value?

Identify those points. Protect them. Reserve them for humans.

Then look at the rest of the workflow. The parts that do not require judgement. Those are the parts where AI can help.

The goal is not to eliminate judgement. The goal is to free it from the routine, so that it can be applied where it matters.

What happens when the output is wrong?

AI will make mistakes. The question is not whether it will fail, but what happens when it does.

Before you deploy AI, ask: what is the cost of a mistake? Can you detect it? Can you correct it? Can you tolerate it?

If the cost of a mistake is catastrophic, and you cannot detect it reliably, AI is not appropriate.

If the cost of a mistake is low, and you have a feedback loop to catch and correct errors, AI may be appropriate.

This is not about perfection. It's about risk management.

Who owns escalation?

AI should not make final decisions in high-stakes contexts. It should support decisions.

But this only works if there is a clear escalation path. If the AI is uncertain, or if the task is outside its scope, who does it escalate to? What information does it provide? What is the process?

Before you deploy AI, define the escalation path. Make it explicit. Test it.

If you cannot define who owns escalation, you are not ready to deploy AI.

Are our precedents and templates consistent?

AI learns from your data. If your precedents are inconsistent, the AI will be inconsistent.

Before you deploy AI, ask: are our templates standardised? Are our precedents up to date? Do we use consistent terminology?

If the answer is no, fix that first. Standardise the language. Clean up the precedents. Make the data coherent.

This is not just for AI. It's good practice. But AI makes the cost of inconsistency impossible to ignore.

Have we documented the rules?

AI cannot infer your risk appetite, your client preferences, or your internal policies. You have to tell it.

Before you deploy AI, ask: have we written down the rules? Do we have documented risk thresholds? Do we have clear policies on what is acceptable and what is not?

If the rules exist only in people's heads, the AI cannot follow them. And neither can a new joiner.

Document the rules. Make them explicit. Make them testable.

Do we have a feedback loop?

AI does not improve on its own. It improves through feedback.

Before you deploy AI, ask: how will we know if it's working? How will we measure success? How will we collect feedback? How will we use that feedback to improve the system?

If you do not have a feedback loop, you are deploying AI blind. You will not know if it's working. You will not know if it's drifting. You will not be able to improve it.

Build the feedback loop first. Then deploy the AI.

What would we do if AI disappeared tomorrow?

This is the most important question.

If AI disappeared tomorrow, would your processes still work? Would your team still be effective? Would your work still be high quality?

If the answer is no, you have not built a sustainable system. You have built a dependency.

The work you do to become ready for AI, defining tasks, standardising processes, documenting knowledge, should make you better regardless of whether you use AI.

AI is the catalyst. But the real work is building a robust, scalable, resilient operation.

If you have done that work, AI will help. If you have not, AI will just expose the gaps.

The shift in mindset

These questions require a shift in mindset.

They require you to stop thinking about AI as a magic solution, and start thinking about it as a tool that requires preparation.

They require you to stop asking "What can AI do for us?" and start asking "What do we need to do to be ready for AI?"

They require you to accept that the barrier to AI adoption is not the technology. It's the organisation.

The practical benefit

These questions are not just about AI. They are about operational excellence.

If you can answer these questions, you have a well-run legal team. You have clear processes. You have documented knowledge. You have feedback loops. You have realistic risk management.

AI is just the thing that makes the value of these practices impossible to ignore.

The honest conversation

These questions are hard. They expose gaps. They require you to look honestly at how your team operates.

But they are also liberating. They give you a roadmap. They tell you what to fix. They make the path to AI adoption clear.

The conversation about AI should not be "When will the technology be ready?"

The conversation should be "Are we ready? And if not, what do we need to do?"

That is a better question. And it's one you can actually answer.

See Appendix: Practical Templates for actionable tools to help you answer these questions.

The profession's side of the bargain

AI will keep improving. The models will get smarter. The context windows will get longer. The reasoning will get better. The hallucinations will reduce.

This is inevitable. The trajectory is clear. The investment is enormous. The progress is relentless.

But legal cannot wait for AI to be perfect and it should not.

Because the barrier to AI adoption in legal is not primarily the technology. It's the readiness of legal work itself.

The choice

Legal work, as it's currently practised in many organisations, is not ready for AI.

Tasks are under-specified. Processes are undocumented. Knowledge lives in people's heads. Precedents are inconsistent. Assumptions are implicit. Feedback is absent.

This is not a criticism of lawyers. It's a description of how the profession has evolved. Legal work has always relied on skilled humans compensating for the absence of structure and skilled humans are very good at this.

But this model does not scale. It does not survive staff turnover. It does not support delegation, because it cannot be automated.

None of this is new. Legal operations professionals, legal project managers, and process improvement experts have been identifying these problems for years. They have advocated for standardisation, documentation, and structured workflows. But the profession has largely treated these as optional improvements, valuable for efficiency, perhaps, but not essential to the practice of law.

AI is not the cause of this problem. AI is just the thing that makes the problem impossible to ignore.

The profession's responsibility

The legal profession has a choice.

It can wait for AI to get better, and hope that eventually the technology will be good enough to work with messy, under-specified, inconsistent processes.

Or it can do the work to make legal work ready for AI.

The second path is harder. It requires effort. It requires change. It requires confronting hard truths about how work is currently done.

But it's also the only path that leads to sustainable progress.

The work is good for you anyway

The surprising, happy truth is that the work required to become ready for AI is good for you regardless of whether you ever use AI.

Defining tasks clearly makes delegation easier. Standardising processes makes quality more consistent. Documenting knowledge makes teams more resilient. Building feedback loops makes work more effective.

These are not AI practices, it's just good practice. They are the foundation of any well-run professional operation.

AI is just the catalyst. It's the thing that makes the cost of not doing this work impossible to ignore.

The benefit of going first

The organisations that do this work first will have an advantage.

They will be able to adopt AI more effectively. They will be able to scale more reliably. They will be able to deliver higher quality at lower cost.

They will also be better organisations. They will have clearer processes, more consistent outputs, more resilient teams.

The competitive advantage is not the AI. The competitive advantage is the operational excellence that makes AI possible.

The risk of waiting

The risk of waiting is not that AI will pass you by. The risk is that your competitors will do the work, and you will not.

They will define their tasks. They will standardise their processes. They will document their knowledge. They will build feedback loops.

And then they will adopt AI. And they will be faster, cheaper, and more reliable than you.

Not because they have better AI. Because they have better systems.

The role of leadership

This work does not happen by accident. It requires leadership.

It requires someone to say "We are going to do this properly. We are going to define our work. We are going to standardise our processes. We are going to document our knowledge."

It requires someone to invest time and effort in unglamorous work that does not produce immediate results.

It requires someone to resist the temptation to buy a tool and hope it solves the problem.

This is not technical leadership. This is professional leadership. And it's one of the most important responsibilities of legal leaders today.

The long view

AI adoption is not a project. It's a transformation.

It will take years, not months. It will require sustained effort, not a one-off initiative. It will require cultural change, not just technical change.

But the organisations that commit to this path will emerge stronger. Not just because they have AI, but because they have built the foundations that make AI effective.

One of the best outcomes I have seen came from a team that actively delayed automation, even though they were under pressure to "do something with AI." Before any tooling decisions, they spent the time doing deeply unglamorous work: writing down how work actually flowed, arguing about naming, standardising inputs, deciding what "done" really meant.

It felt really slow at the time.

People complained that other teams were already piloting tools while they were still arguing over spreadsheets. When they finally automated, it was almost boring, the automation simply mirrored a process that already worked. Errors dropped because edge cases had been surfaced earlier, and new joiners ramped faster because the process was explicit. The successful team was not clever with technology. They were disciplined with structure.

The bargain

AI is offering the legal profession a bargain.

The technology will keep improving. It will get smarter, faster, more capable, and in turn it'll handle more complex tasks and will make fewer mistakes.

In return though, the profession has to do its part. It has to make legal work ready for AI.

It has to define tasks clearly. It has to standardise processes. It has to document knowledge. It has to build feedback loops. It has to separate judgement from compensation.

This is the profession's side of the bargain and it's not optional.

The final question

The question is not "When will AI be good enough for legal?"

The question is "When will legal be good for AI?"

And the answer is: when we do the work.

The work is hard, unglamorous, and not exciting. But it's necessary.

And the organisations that do it will not just be ready for AI. They will be better at everything.

AI is not the future of legal work. Clarity, consistency, and process discipline are the future of legal work.

AI is just the thing that makes that future inevitable.

The profession's side of the bargain is to be ready, not someday, but now.