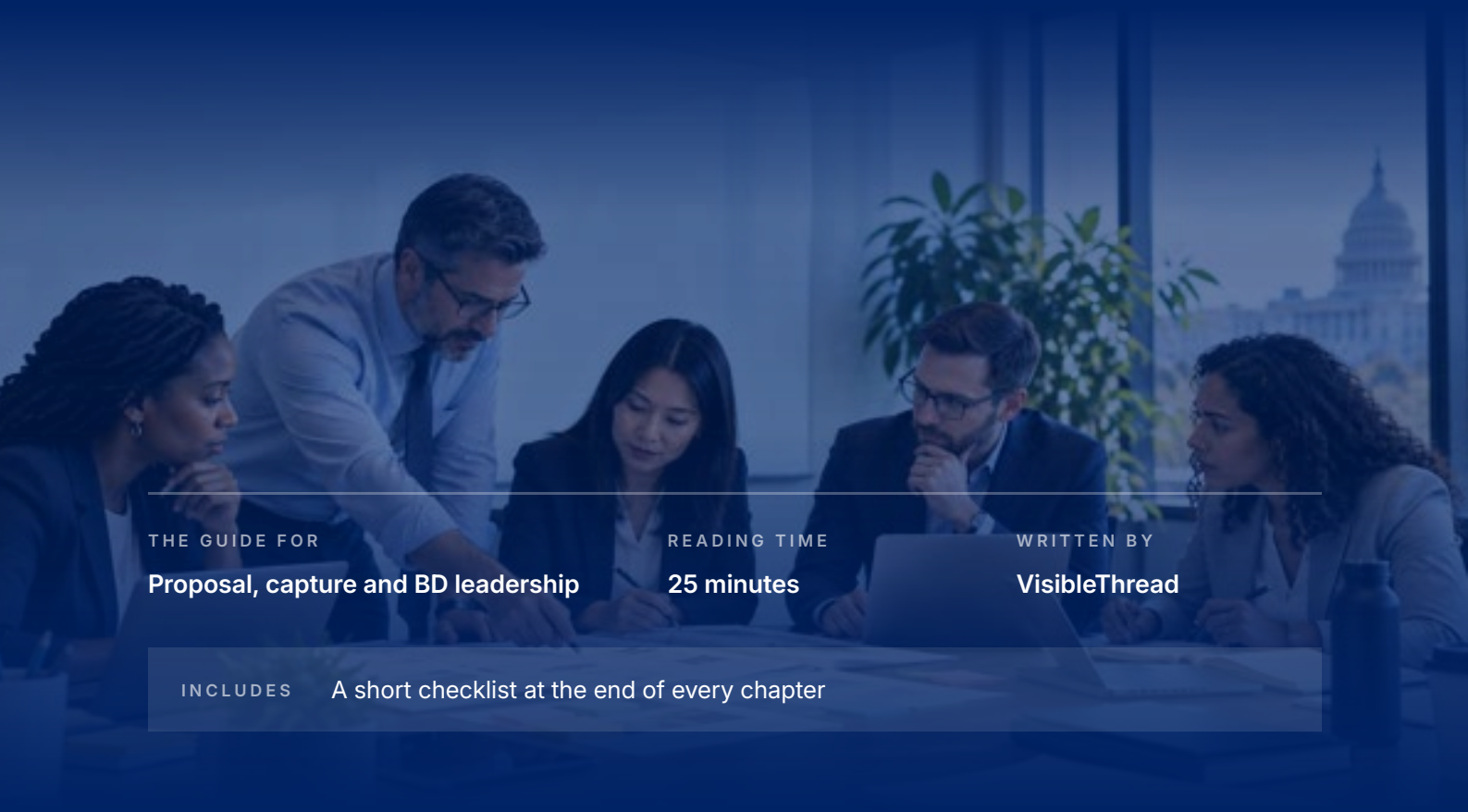
EXECUTIVE GUIDE · AI ADOPTION SERIES

Making it work

Designing hybrid workflows, scoping a pilot, training, change management, and measuring what matters.



THE GUIDE FOR
Proposal, capture and BD leadership

READING TIME
25 minutes

WRITTEN BY
VisibleThread

INCLUDES A short checklist at the end of every chapter

ABOUT THIS GUIDE

Buying the tool was the easy part.

What separates results from shelfware is how you redesign the work, and whether anyone is still doing it in six months.

Part 2 of a two-part series on AI adoption in bid and proposal teams, especially regulated, high-compliance environments. Part 1 covered readiness; this part starts after the solution decision.

Read the chapters in order. Design, pilot, training, governance, and measurement each build on the last.

Selected sections include a **Do this next** step, and every chapter closes with a checklist.

THE IDEA THAT RUNS THROUGHOUT

The most effective AI workflows are not purely generative. They combine deterministic processes, generative AI, and human judgment into one structure your team can trust.

THE ADOPTION LOOP

PART 1 · READINESS PART 2 · THIS GUIDE

Ready
Diagnose data and process readiness

Select
Choose a solution that fits your reality

Design
Build the hybrid workflow

Test
Prove it in a scoped pilot

Adopt
Train roles, retire the old path

Measure
Track outcomes, not logins

Improve
Feed findings back

One loop, two guides. Part 1 covered readiness and selection. This part carries the loop through design, testing, adoption, measurement, and improvement, and then back around: what you measure feeds what you improve.

WHERE THIS SITS IN THE SERIES

Contents

Part 1 covers readiness and selection. This part covers everything after the purchase decision.

PART 1

Are you ready?

Organizational and data readiness, and how to evaluate a solution against it. Includes the scored AI Readiness Assessment.

PART 2 · YOU ARE HERE

Making it work

Designing hybrid workflows, scoping a pilot, training, change management, and measuring what matters.

Designing hybrid AI workflows

The organizations seeing real results are not just deploying AI. They are redesigning how work gets done.

The teams that succeed redesign the work itself.

Now comes the step that separates teams that get real value from those that get an expensive demo.

AI adoption stalls when a new tool is bolted onto an old process, because the tool inherits all the friction that process already had.

No single layer is sufficient on its own.

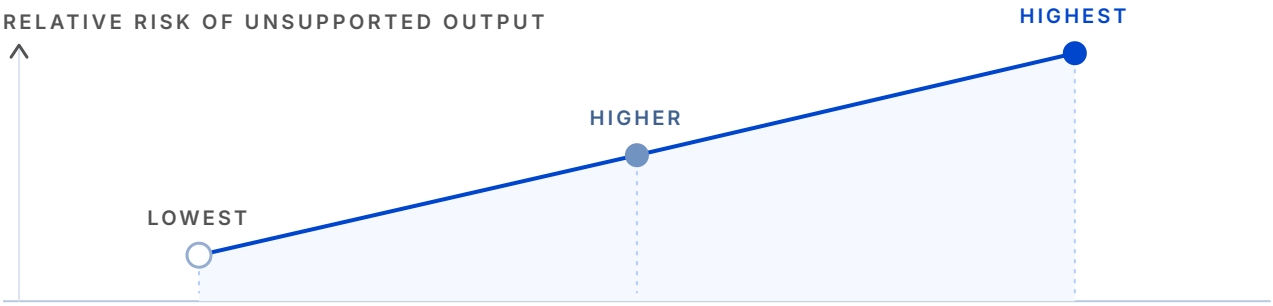
Deterministic tooling is exact and auditable but cannot write a compelling narrative. Generative AI is fast and fluent but will confidently invent things if left ungrounded. Human judgment provides context, ethics, and accountability but does not scale to a 200 page solicitation under a deadline. Put together, each layer covers the others' weaknesses.

TECHNICAL QUALIFICATION

One qualification: deterministic means repeatable and auditable where rules can be explicitly defined, not inherently correct. A badly defined rule fails consistently, and extraction breaks on malformed tables and scanned documents. The deterministic layer still has to be tested against real documents, which is what the **golden set in Chapter 4** is for.

Error risk rises as deterministic structure falls away

Relative likelihood of unsupported output, by task shape. Directional, not to scale.



Extractive QA	Open-ended generation	Multi-step agent chains
Answer exists in source	Drafting a narrative	System chains its own actions

LESS DETERMINISTIC STRUCTURE →

Directional, based on ranges reported in published LLM evaluations (Deepchecks, 2026, among others). Absolute rates vary widely with model, grounding, document quality, and review design. Read this as relative risk, not a benchmark.

WHAT THE GRADIENT MEANS FOR YOU

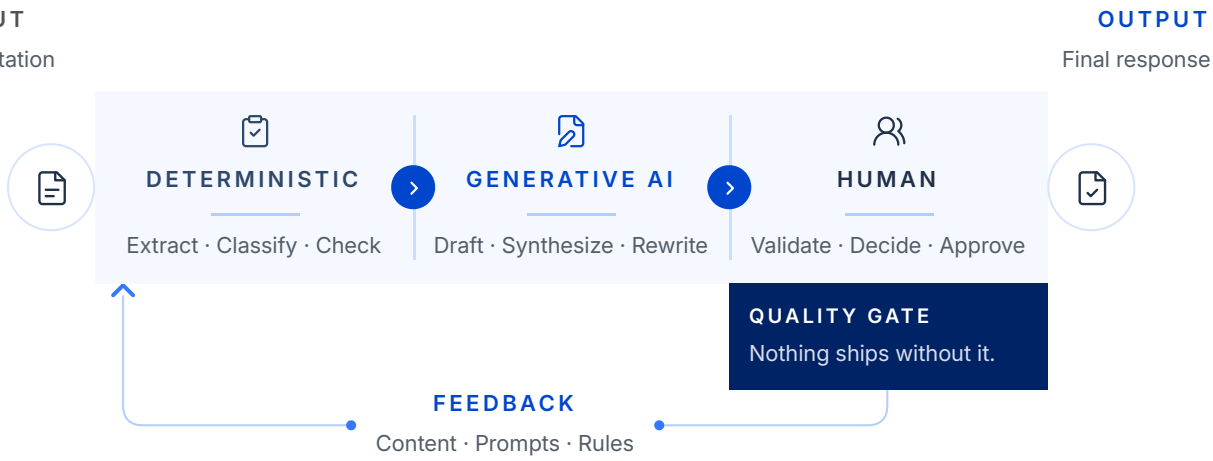
The less deterministic structure a task has, the more its accuracy depends on grounding and review.

The gradient maps almost exactly onto the tasks your team performs. Extracting every “shall” from a solicitation has one correct answer that exists in the document, so it belongs to the deterministic layer, where error risk is lowest and every result traces to a page and a line.

Drafting a technical approach is open-ended generation, where unsupported content is common enough that a human must validate the output before it moves. Letting a system read a solicitation, decide which sections matter, draft them, and assemble a response without stopping is the third category, and the one to be most careful with.

The point is not that generative AI is unreliable and should be limited. It is that error risk is a function of task structure, and task structure is something you design. Give a task more deterministic scaffolding and grounding in approved content, and you move it down the gradient.

One workflow, three layers working together



Why agentic chains compound

A wrong requirement classification in step one does not stay a step one problem. It propagates into the compliance matrix, into the response outline, into the draft written against that outline, and into the review that checks the draft against the matrix that was wrong to begin with. By the time a person sees the output, the error has been laundered through four steps that each looked reasonable.

This is what the handoffs are for. Every point where one layer passes work to another is a place to catch an error while it is still cheap, and the deterministic layer is doing double duty: it produces the exact work at the front of the process and it re-checks the generative work at the end.

DO THIS NEXT

Take one task your team does today and label each part as deterministic, generative, or human. Most teams find they are using a human or generative step where a deterministic one would be safer, accepting generative error risk on work that could run deterministically, with an audit trail.

What each layer is best at

LAYER	BEST AT	SHOULD NEVER OWN	EXAMPLE IN PROPOSALS
Deterministic	Exact, repeatable, auditable extraction and checks	Judgment calls and persuasive writing	Pulling every “shall” into a compliance matrix
Generative AI	Drafting, summarizing, and rephrasing at speed	Final accuracy or compliance sign-off	Producing a first-draft technical approach
Human oversight	Context, strategy, ethics, accountability	Manual work better handled by the other layers	Approving win themes and owning final quality

DO THIS NEXT Look for any place where generative AI currently owns final accuracy or compliance sign-off. That is the highest-risk handoff to fix first.

A worked example: compliance matrix and response

Here is how the three layers move together through a single, familiar task. Notice that the work passes deliberately from one layer to the next, and a human owns the final gate.

1. **DETERMINISTIC** — Extract every requirement from the solicitation; tag shall / should / may; build the compliance matrix and L/M crosswalk.
2. **GENERATIVE** — Draft response language for each requirement, grounded in the approved content library and tagged proof points.
3. **HUMAN** — SME validates technical accuracy; capture lead confirms win themes and differentiators are present.
4. **DETERMINISTIC** — Re-run compliance and readability checks; confirm every requirement is addressed and scoring is clean.
5. **HUMAN GATE** — The proposal manager owns the final quality gate and sign-off. Nothing ships without this step.

What grounding actually means

“Grounded in approved content” is the phrase that does the most work in this chapter, and it is worth being precise about, because it is also the phrase vendors use most loosely. Grounding is not a setting. It is four specific properties, and a system either has them or it does not.

01 Scope

The system draws only from a defined set of sources you control, not from the open model. You should be able to state exactly which libraries, folders, and document types are in scope for a given task, and change that scope per opportunity.

02 Attribution

Every generated passage traces back to the source content it came from, at a granularity a reviewer can check. A citation to a 90 page past proposal is not attribution. A citation to a tagged capability block is. In a regulated environment this is also a compliance control.

03 Freshness

The system knows how current its sources are and does not silently prefer a superseded version. This is where version control stops being housekeeping and starts being an accuracy control.

04 Refusal

When the approved content does not cover a requirement, the system says so instead of filling the gap from general knowledge. A model that never returns “no source found” is not grounded, it is just confident.

“No source found”

A grounded system refuses rather than fills the gap. Refusal is not a limitation; it is the property that makes the other three checkable.

Grounding is not something you buy outright. The vendor provides the mechanism; your content library determines whether the mechanism has anything to hold onto. A well-grounded system pointed at an unstructured archive still produces unreliable output, which is why the readiness work in Part 1 comes first.

Make governance proportional to consequence

Low risk

Brainstorming, internal summarization. Light-touch: approved tools, common sense.

Medium risk

Grounded internal drafting. Source scoping and spot review.

High risk

Customer deliverables, compliance, contractual interpretation, sensitive data. The full workflow in this chapter, every time.

Failure modes to design for

- **Hallucinated citations**: a reference that looks checkable and is not. Attribution at block level catches it.
- **Poisoned or stale sources**: the system faithfully reuses whatever the library holds, including planted or superseded content.
- **Prompt injection**: instructions hidden in a processed document. Treat inbound files as untrusted input.
- **Silent model changes**: vendor updates shift output. Re-run your golden set after releases.
- **Automation bias**: reviewers approve faster than they read. Rotate reviewers and audit samples.

A second example: responding to an amendment

The compliance matrix example is the clean case: a defined input, a known output, plenty of time. Amendments are the hard case, and they are a better test of whether a workflow actually holds. An amendment arrives late, changes an unknown number of things, and gives you days rather than weeks. It is exactly the situation in which teams abandon process and start working by memory.

It is also where the deterministic layer earns the most, because the first question an amendment poses is mechanical: what changed? That is a comparison task with one correct answer, and the single worst use of a human under time pressure. A person reading two 200 page documents side by side will miss changes, and will be slowest exactly when speed matters most.

WHY THIS MATTERS

A person reading two 200-page documents side by side will miss changes, and will be slowest exactly when speed matters most.

AMENDMENT WORKFLOW · SEVEN STEPS, THREE LANES

DETERMINISTIC	1	**Diff the amendment** against the prior version; produce a complete, auditable list of added, changed, and deleted requirements.
	2	**Map each change** to the affected proposal sections through the existing compliance matrix; flag every section now non-compliant or orphaned.
HUMAN	3	**The proposal manager triages the impact list**: what is trivial, what is substantive, what changes the bid decision or the solution itself.
GENERATIVE	4	**Redraft only the affected passages**, grounded in approved content and constrained to the surrounding narrative that did not change.
HUMAN	5	**SMEs validate the substantive changes**; the capture lead confirms win themes still hold against the revised requirements.
DETERMINISTIC	6	**Re-run full compliance and consistency checks** across the whole document, not just the changed sections.
HUMAN GATE	7	**Final quality gate and sign-off.**

STEP 3

Human on purpose, and early

The more consequential the judgment, the earlier a human takes it. Deciding which changes are substantive determines how much of the rest happens at all.

STEP 4

Deliberately narrow

Regenerating whole sections discards validated language, introduces unreviewed content, and makes the diff for reviewers larger than the actual change.

STEP 6

Checks everything

Amendment errors are usually second-order: a changed page limit, a renumbered section, a shifted definition, none of which touch the passages you rewrote.

THE PATTERN TO READ OFF THE LANES

Human judgment moves earlier as consequence increases.

If your hybrid workflow only works when there is time, it is not a workflow. It is a preference.

THE BOTTOM LINE

Human judgment remains the quality gate.

No generative step owns final accuracy or compliance sign-off.

Defining human review responsibilities

Accountability gets lost in the handoffs unless you name it. For every workflow, define where human review takes over and who owns each decision. A simple rule keeps it durable: the more consequential or judgment-heavy the output, the earlier and more thoroughly a human reviews it.

CHAPTER 03 · DO THIS

Three questions to ask of any workflow

01

Where can deterministic rules do the exact, repeatable work? Put them first.

02

Where does generative AI accelerate a draft, and what does it need to stay grounded?

03

Where does a human take over, and who specifically owns that decision and sign-off?

As processes mature, oversight becomes the more durable human role. Content creation increasingly shifts to the generative layer; what remains, and grows in value, is the judgment to direct, validate, and stand behind the output. That is a more defensible role than racing the machine at first-draft speed.

CHAPTER 3 · CHECKLIST

Redesigning the work

- 01 ☐ We labeled each step of a real task as deterministic, generative, or human.
- 02 ☐ Deterministic rules handle the exact, repeatable work, and go first.
- 03 ☐ Generative drafts are grounded in approved, tagged content.
- 04 ☐ Grounding is real, not nominal: scoped sources, checkable attribution, current versions, and a system that will say “no source found.”
- 05 ☐ No generative step owns final accuracy or compliance sign-off.
- 06 ☐ The workflow holds under time pressure, including on amendments, not just on planned responses.
- 07 ☐ Every handoff has a named human owner.
- 08 ☐ A human owns the final quality gate, and nothing ships without it.

Starting small

The workflow you designed is a hypothesis. A pilot is how you test it before the whole organization is depending on it.

Starting small is not caution. It is how you buy the right to go big.

You now have a workflow on paper with clear layers, explicit handoffs, and named owners. The temptation is to roll it out, because the design feels finished and the business case was approved months ago. Resist it.

A workflow that has never met a real pursuit is a set of assumptions, and the fastest way to convert them into evidence is to run them somewhere small, visible, and recoverable.

An enterprise rollout has one bad property: it produces its first real feedback at the moment when changing course is most expensive. A pilot inverts that. It gives you a small number of users, a short cycle, and the political room to change the workflow without anyone calling it a failure.

Two things follow that matter later: you produce internal proof, far more persuasive to a skeptical SME than any vendor case study, and you develop your own experts, so the second wave of training comes from colleagues rather than a vendor.

THE FAILURE MODE

The pilot that was never designed to end. A pilot with no exit criteria becomes a permanent side project for enthusiasts, which is indistinguishable from partial adoption.

EXPERIMENT	PILOT · THIS CHAPTER	OPERATIONALIZING
Days, one person	**60 to 90 days**	**The phased path**
Low-risk productivity uses. No formal structure needed.	For anything that touches an operational workflow. The structure below is for this one.	At the end of this chapter, once the pilot has answered its question.

Where to start

This is the question teams get wrong most often, usually by picking the most painful problem rather than the most winnable one. Two inputs decide it. The first is your Part 1 readiness result: your lowest pillar tells you what has to be shored up before a pilot is viable, and your strongest tells you where you already have the foundation to succeed. The second is the error-risk gradient from Chapter 3. Together they form a screen of value, feasibility, risk, and frequency: the best first use case is not the easiest task, it is the one frequent enough to measure, important enough to matter, and low-risk enough to learn from safely.

Choose a first use case that meets all five

High frequency	You need several cycles inside the pilot window. A task performed twice a year teaches you nothing in 60 days.
Deterministic-heavy	Tasks at the extractive end of the gradient carry the lowest error risk, so early results reflect the workflow rather than model variance.
Genuinely painful	The team must feel the improvement. A faster version of something nobody minded doing does not change behavior.
Measurable in hours	You need a before number and an after number that a skeptic accepts.
Low blast radius	An error should cost rework, not a bid.

START HERE

Compliance matrix construction and requirements extraction

Frequent, at the extractive end of the gradient where error risk is lowest, universally disliked, trivially measurable in hours, and an error surfaces inside your own review rather than in front of an evaluator. It is also where the deterministic layer does the most visible work, which teaches the three-layer model better than any training deck.

NOT YET

Executive summaries, win themes, anything strategic

They sit at the open-ended end of the gradient, quality is a matter of judgment rather than measurement, and they are the tasks your most senior people are most protective of. Winning there requires the credibility you have not built yet.

DO THIS NEXT

Name your first use case and write down the five criteria beside it. If it fails two or more, pick a different one. Starting on the wrong task costs a quarter and a reputation.

Scoping the pilot

Four boundaries, all set before anyone logs in.

01 One team, not volunteers from several

A single intact team shares pursuits, deadlines, and conversations, which produces real practice rather than isolated experiments. Include at least one skeptic. A pilot staffed entirely with enthusiasts proves only that enthusiasts will use software.

02 One deliverable type

The use case you just chose, on every qualifying pursuit in the window, not on whatever happens to be convenient.

03 A fixed window, 60 to 90 days

Long enough for several cycles and for novelty to wear off. Shorter than 60 days and you measure enthusiasm; longer than 90 and it becomes a parallel process nobody ever decides about.

04 Written success criteria, agreed in advance

Not “see how it goes.” Specific thresholds on the metrics you baselined, plus a named decision date and a named decision maker.

What to put in place before day one

Everything here is cheap now and expensive to retrofit.

BASELINE	Time the current process on three recent pursuits. Without a before number, any after number is an anecdote.
PILOT OWNER	A named pilot owner, distinct from whoever owns the software contract. Adoption needs someone whose job it is.
GOVERNANCE	Not the full model from Chapter 6, but the parts that touch what the pilot produces: who approves content entering the library, who owns the final quality gate, how AI-assisted output is recorded. Governance retrofitted after a pilot arrives as bureaucracy imposed on people who were doing fine. Governance present from day one is simply how the work is done.
INSTRUMENTATION	Measurement running from week one so you see trajectory rather than endpoints.
GOLDEN SET	Ten representative solicitations, fifty known requirements, a few amendment scenarios, and the expected answers. Run the tool against it before the pilot and after every vendor release, checking completeness, accuracy, citation correctness, and whether the system says “no source found” when it should. It is the cheapest way to separate model problems from library problems.
RETRAINING PATH	Described below, standing before you need it.

Governance and retraining are not phase two

This is the lesson from the defense contractor pilot in Chapter 6, and it is worth knowing before you run yours. That program lost nearly 40% of its usage when the reinforcement cadence ended. What saved it was not a better tool or a more motivated team, but that governance and a retraining path had been built in from the start: the drop was caught within a week, diagnosed against specific roles, and reversed with targeted retraining. Usage recovered to 90% of the pilot average in four weeks.

Retraining is a capability, not an event

Every part of that recovery depended on infrastructure that existed before the problem did. The measurement was already running, so the drop was visible. The governance already named owners, so there was someone to escalate to. The retraining path already existed, so the response was scheduling rather than designing.

01 Triggers that fire without anyone deciding

A measurable usage drop, a new hire, a product release that changes a step, a new deliverable type entering scope, or a quarter passing since the last refresher.

02 Short, role-specific formats

Thirty minutes on the two decisions that role owns, on a live pursuit. Not a repeat of onboarding.

03 A named owner

Responsible for noticing the trigger and running the response.

04 Materials that already exist

The role guides and prompt examples from Chapter 5 are the retraining content. Building them once during the pilot is what makes the response fast later.

Repetition is the only thing that moves behavior, and repetition has to be somebody’s job.

Training closes the knowledge gap but not the behavior gap. People return to the same workspace, the same habits, and the same deadline pressure.

A phased path

Phases exist so that expansion is a decision rather than a drift. Each one has an exit criterion, and you do not move until it is met.

PHASE 0 · 4–8 WEEKS

Foundation

Raise the lowest pillar enough to support the pilot scope: tag and mark the content the pilot will touch, name owners, baseline the process.

EXIT

The pilot use case has a trustworthy, findable, owned content set behind it.

PHASE 1 · 60–90 DAYS

Pilot

One team, one deliverable type, full reinforcement cadence, measurement from week one.

EXIT

Success criteria met, and the team says it would object to going back.

PHASE 2 · 1–2 QUARTERS

Extend

Add a second team or deliverable type. Retire the old path for the pilot team. Move governance from pilot scope to the full ownership model.

EXIT

Two groups running independently, old path closed, no fallback in use.

PHASE 3 · ONGOING

Standard

The workflow is the default rather than the initiative. Governance sits in the operating rhythm; retraining triggers are live.

EXIT

New hires learn it as the process, not as the new system.

EXPERIMENT → PILOT → EXTENSION → OPERATING STANDARD →

Phase 0 is the one that gets skipped

Skipping it is how a well-designed pilot produces mediocre results that get blamed on the tool. If the content behind your use case is not ready, the pilot is testing your library, not your workflow.

Phase 1's exit is behavioral, not just numerical

Metrics can meet threshold while a team quietly waits for the pilot to end. If the team would not object to reverting, you have a result rather than an adoption.

CHAPTER 04 · DO THIS

Give the pilot permission to fail

A pilot that cannot return a negative result is a rollout with extra steps, and everyone involved knows it. Decide in advance what would cause you to stop, narrow scope, or change vendors, and say so out loud when the pilot is announced.

It makes participants honest, because their feedback can actually change the outcome. And it protects the initiative: a pilot that stops for stated reasons is a disciplined organization managing risk, while a rollout that quietly fizzles is a failed AI project that makes the next attempt harder to fund.

CHAPTER 4 · CHECKLIST

Starting small

- 01 ☐ Our first use case meets all five criteria, and we can say why we did not pick a harder one.
- 02 ☐ The pilot is one intact team, one deliverable type, and a fixed 60 to 90 day window.
- 03 ☐ At least one skeptic is in the pilot group.
- 04 ☐ We baselined the current process before anyone logged in.
- 05 ☐ A named pilot owner exists, separate from the software owner.
- 06 ☐ Governance and measurement were in place on day one, not added later.
- 07 ☐ Retraining triggers, formats, and an owner are defined before they are needed.
- 08 ☐ Written success criteria, stop criteria, a decision date, and a decision maker all exist.
- 09 ☐ Each phase has an exit criterion, and expansion is a decision rather than a drift.
- 10 ☐ A golden set exists and is re-run after every vendor release.

Training & driving adoption

Adoption stalls after the early enthusiasm fades. Training is how you carry teams across that gap.

Training is the bridge between a tool being available and a tool being trusted.

What determines whether the workflow survives contact with the organization is whether people adopt it. Training is not a one-time event at launch.

The most effective programs are role-based, embedded in the real workflow, and reinforced over time. They teach the exact handoffs you designed in Chapter 3 and the decisions your pilot is about to put in front of people.

Train to the role, not the feature

Generic tool tours teach buttons, not behavior. Effective training maps to what each role actually does in the hybrid workflow and where their judgment matters most. These are responsibilities, not necessarily job titles; in smaller teams, one person may own several.

ROLE	WHAT THEY NEED TO LEARN	WHAT "GOOD" LOOKS LIKE
SMEs	How to contribute structured, reusable knowledge and validate AI drafts in their domain	Fast, confident validation; clean source content
Writers / proposal staff	How to assemble and refine AI-grounded drafts and where to apply judgment	Faster assembly, consistent quality, less rework
Capture / BD	How AI supports analysis and theme development without replacing strategy	Sharper, evidence-backed win themes
Proposal managers	How to own quality gates, compliance sign-off, and review handoffs	Clear accountability, nothing slips through
Admins / KM leads	How to maintain taxonomy, metadata, prompts, and guardrails	A library that stays structured and trusted

Practical training approaches that stick

01 Use a real solicitation, not a clean demo file, so people see the tool against genuine complexity.

02 Teach the workflow, not just the software. Where does each layer contribute, and where do humans take over?

03 Pair new users with a power user for the first few cycles to build confidence.

04 Create short, role-specific reference guides and prompt examples people can reuse.

05 Reinforce after launch with refreshers tied to lessons learned, not a single kickoff session.

Train inside the workflow

People learn the new way fastest when training happens in the context of real work, not in a separate sandbox. Anchor sessions to a live or recent opportunity. Show the deterministic, generative, and human steps in sequence. Let people practice the handoffs and the review decisions they will actually own.

Removing the old path

They build their own path, and that path costs money.

Adoption rarely sticks while the old workflow still runs underneath the new one. People fall back to a familiar path left open under deadline pressure, so closing it, with support and a clear reason, is often what turns a pilot into the standard way of working.

There is a second reason to close it, visible on the budget rather than adoption metrics. When the sanctioned workflow does not fit how people work, they do not stop working, they buy their own tools. That is rarely rule-breaking. It is usually the reasonable response of a team under deadline pressure that found the approved tool slower than the alternative.

37%

of company applications are employee-led purchases made outside IT

44%

of SaaS licenses go unused in a typical month

ZYLO, SAAS MANAGEMENT INDEX · BOTH FIGURES

IN A PROPOSAL ENVIRONMENT THE SHAPE OF IT IS FAMILIAR

A separate subscription for document comparison.

A personal drive holding the version of the past performance library people actually trust.

A consumer AI tool someone pastes requirement language into because it is faster.

CHAPTER 05 · DO THIS

Close the gap, then close the path

Each workaround is individually defensible and collectively expensive, and in a regulated environment the spend is the smaller problem. Content flowing through unsanctioned tools sits outside your version control, your approval workflow, your audit trail, and potentially your security boundary. Every one of the four readiness pillars from Part 1 leaks through it.

The counter-move is not enforcement. Shadow IT is a symptom of friction, and where a workaround has taken hold, it is usually pointing at a real gap. Find out what the workaround does better, close that gap in the sanctioned workflow, then retire the old path and say clearly why.

CHAPTER 5 · CHECKLIST

Bringing people along

- 01 ☐ Everyone who touches a proposal is mapped to a role.
- 02 ☐ Training is built around what each role decides, not around features.
- 03 ☐ We train on a real solicitation, not a clean demo file.
- 04 ☐ New users are paired with a power user for their first cycles.
- 05 ☐ Role-specific reference guides and prompt examples exist and are reused.
- 06 ☐ We have inventoried the workarounds and shadow tools in use, and know whether each one is a gap to close or a path to close.
- 07 ☐ We have set a date to retire the old workaround, with a clear reason.

DO THIS NEXT

Name the one workaround your team relies on most, and decide this week whether it is a gap to close or a path to close.

Change management & the journey

Early enthusiasm gives way to drift. Leaders build the operational structure that makes adoption stick.

Sustainable adoption is an operational discipline, not a launch event.

Training gets people using the new workflow. The harder problem is keeping them there after the novelty wears off. The pilot goes well, energy is high, and then attention shifts while old habits reassert themselves.

Drift is predictable. Enthusiasm fades, the workflow was never fully embedded, ownership is fuzzy, and there is no agreed-upon definition of success to hold against. Each of these has a direct counter-move.

WHY IT DRIFTS	WHAT LEADERS DO ABOUT IT
Energy fades after launch	Build a sustained cadence of reinforcement and visible wins
The old process still runs underneath	Redesign the workflow and retire manual workarounds
Unclear ownership of content and outputs	Assign explicit owners and approval responsibilities
No agreed definition of success	Define metrics up front and review them on a schedule
Governance is treated as a brake	Frame governance as the guardrail that lets teams move faster, safely

CASE STUDY · 60-DAY PILOT

Usage held exactly as long as the cadence held

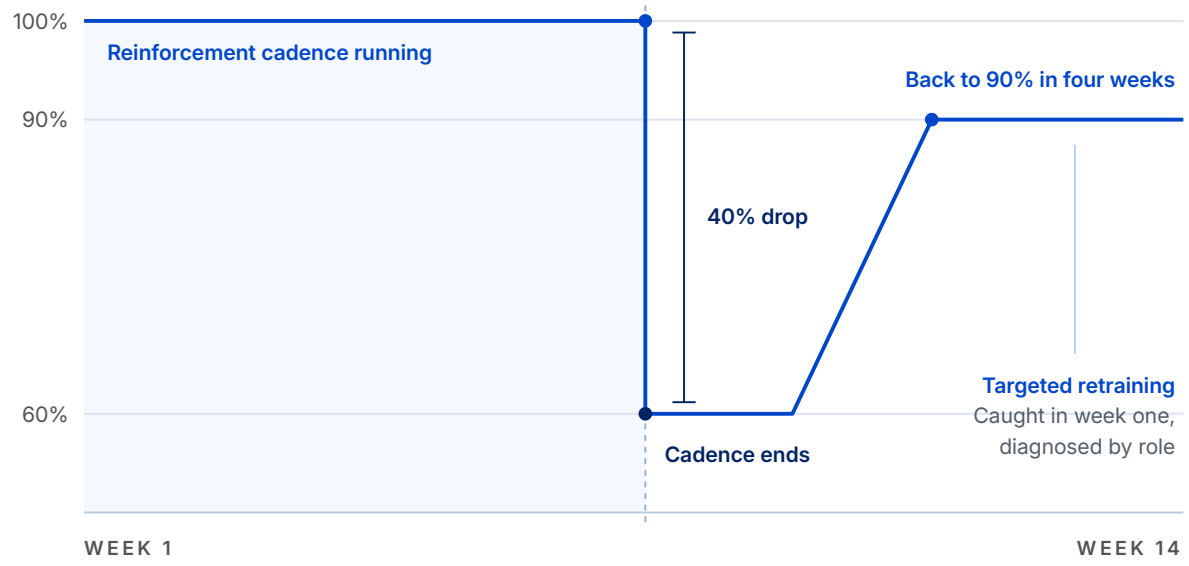
40%

usage decline when the reinforcement cadence ended

ANONYMIZED VISIBLETHREAD IMPLEMENTATION

An anonymized VisibleThread implementation with a large defense contractor: a structured 60 day pilot of weekly working sessions, training modules, and hands-on time. Nothing about the tool, the workflow, the library, or the people changed in that window. What was removed was the reinforcement.

WEEKLY ACTIVE USAGE AS A SHARE OF THE 60-DAY PILOT AVERAGE



Plan for the drop

Do not hope to avoid it. Reinforcement cadence is the variable that moves usage, and any cadence you run will eventually end.

Instrument before launch

Recovering 40% in four weeks was possible because the drop was caught in week one, not because the intervention was extraordinary.

Knowing is not the constraint. Repetition is.

Training closes the knowledge gap but not the behavior gap. People finish a session knowing how the workflow works, then return to the same workspace, the same habits, and the same friction that produced the old behavior.

THE VULNERABLE WINDOW

90 days after go-live

AppStudio's work on post-deployment adoption puts it plainly: this is the most vulnerable window in an application's lifecycle. Miss it and employees develop workarounds, shadow IT fills the gaps, and by the time leadership notices declining metrics the culture around the tool has calcified.

THE PRICE OF THE QUIET ENDING

$4,600 per employee, per year

Absent oversight, users revert to the legacy tools the new system was meant to replace. Zylo puts average SaaS spend near that figure, with 44% of licenses sitting unused in a typical month.

DO THIS NEXT

Look at your rollout plan and find the date the reinforcement cadence ends. That is your risk window. Put a measurement checkpoint and a named owner two weeks past it, before you need them.

Governance is a guardrail, not a brake.

Where governance belongs

Governance is the operational structure that keeps the whole system accountable. It is not about slowing people down; it is about defining the guardrails so AI stays aligned with your standards. In an AI-enabled process, oversight becomes the most enduring human role, making governance the backbone of the workflow rather than an afterthought.

Content standards

Version control

Review cycles

AT THE CENTER

Hybrid AI workflow

Metadata

Ownership

Incident handling

Content standards

Modular paragraphs rather than long narrative sections; clearly defined capability statements; proof points separated from narrative.

Consistent terminology, approved acronyms, and a standard tone and reading level.

Every claim supported by metrics or references, with customer names verified for release.

Key levers and controls

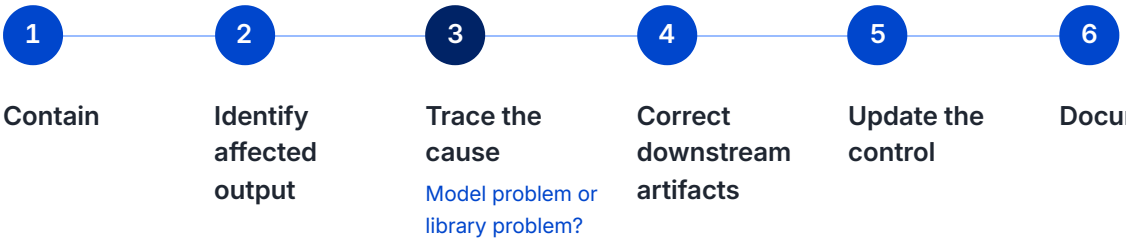
Version control so proposal content keeps a clear history.

Content review cycles so reusable content is checked periodically.

Metadata standards so content stays searchable and AI-retrievable.

When AI gets it materially wrong

Governance sets the rules; incident handling is what happens when the rules fail anyway: a fabricated proof point reaches a draft, a superseded requirement drives an outline. Teams need a path that is not improvised.



The trace step is why Chapter 3's attribution requirement matters: without it, you cannot tell a model problem from a library problem, and you fix the wrong thing.

Content ownership model

Accountability needs names. A simple ownership model removes ambiguity about who maintains, approves, and validates each kind of content. The rows are responsibilities, not headcount; in a small team, the same name may appear several times.

OWNER	RESPONSIBILITY	NAME
Content owners, SMEs	Accuracy of capabilities and technical approaches	
Proposal operations	Maintain taxonomies and content standards	
Marketing / growth leadership	Approve messaging and differentiators	
Contracts / legal	Validate compliance language	
AI / knowledge management lead	Ensure structured content practices are followed	

CHAPTER 06 · DO THIS

Making it stick

CHAPTER 6 · CHECKLIST

Governance and cadence

- [ ] **01** We identified our most likely drift cause and assigned its countermove.
- [ ] **02** We know the date our reinforcement cadence ends, and a measurement checkpoint and named owner sit just past it.
- [ ] **03** The two or three missing governance controls are being stood up now.
- [ ] **04** Every row of the content ownership model has a real name attached.
- [ ] **05** Governance is framed as a guardrail for speed, not a brake.
- [ ] **06** A recurring review cadence is on the calendar.
- [ ] **07** An incident path exists for when AI gets it materially wrong.
- [ ] **08** We are working toward the documented goal state, not a one-time launch.

THE GOAL STATE

01 A defined content taxonomy and standardized metadata tagging.

02 Designated content owners and documented approval workflows.

03 Periodic review cycles and clear evidence traceability.

04 A hybrid workflow that the whole team understands, trusts, and follows.

Adoption is not a finish line. The role of the proposal professional evolves alongside the workflow, shifting from manual production toward direction, validation, and oversight. The same lifecycle discipline applies to the AI itself: prompts, model versions, source collections, and golden-set benchmarks all age, and each needs an owner and a retirement path rather than a one-time approval.

Measuring success

Governance gives you the structure to sustain adoption. Measurement tells you whether it is working.

If success was never defined, adoption has nothing to hold itself against.

Defining success up front is one of the most reliable predictors of whether adoption sticks, and it is the missing piece behind the "no agreed definition of success" failure named in Part 1.

Metrics do two jobs: they demonstrate value to leadership, and they give teams a target to steer toward.

A balanced set of measures

01 Adoption

Are people actually using the workflow?

Active users · % of opportunities run through the workflow · fallback to manual processes

02 Efficiency

Is the work getting faster?

Time to first draft · compliance matrix build time · rework cycles · hours per response section

03 Quality

Is the output better and more consistent?

Compliance coverage · readability scores · review findings · consistency of messaging

04 Outcomes

Is it moving the business?

Win rate · proposals submitted per period · bid capacity · evaluator feedback

DO THIS NEXT

Choose one metric from each lens. Four well-chosen measures beat a dashboard nobody reads.

Participation ≠ Progress

Counting users rather than outcomes creates a false sense of progress. Active user counts and login rates are easy to collect, easy to report, and reliably move in the right direction while everyone is watching.

They tell you people opened the tool, not that the work got faster, more compliant, or better. A program can hold high participation while producing no measurable change in output. The four lenses are ordered deliberately: adoption comes first because it is the precondition for everything else, but it is also the lens most likely to be mistaken for the whole picture.

NEVER REPORT THIS ALONE	PAIR IT WITH
Active users	% of opportunities run end-to-end through the workflow
Sessions logged	Time to first draft
Training completion	Rework cycles or review findings

The participation number tells you people showed up. The outcome number tells you whether showing up changed anything.

When you measure changes what you see

Timing distorts the picture as much as metric choice does. Early data reflects novelty: usage during a pilot or the weeks after launch is inflated by attention, training cadence, and the fact that people are being watched. Late data has the opposite problem, smoothing over the adoption trends that mattered most and arriving after the window in which you could have acted on them.

The 60 day pilot in Chapter 6 is the case in point. Measured at day 60, that program was a success. Measured only at month six, it would have looked like a failure. The truth was neither, and it was only visible because measurement continued through the transition between the two.

01

Baseline before rollout

So you have a pre-novelty number to compare against.

02

Measure continuously

Not at milestones, so a change in trajectory is visible while you can still respond to it.

03

Keep measuring past the cadence

Because that is where the honest number lives.

Leading vs. lagging indicators

Win rate is the metric leadership wants, but it is a lagging indicator that takes months to move. Watch leading indicators first; they tell you whether the lagging results are coming.

LEADING **Watch first**

- Active usage and workflow adoption rate
- Drop in rework and review findings
- Reduction in time to first draft
- Content library coverage and tagging completeness

LAGGING **Confirm value**

- Win rate
- Evaluator scoring on compliance and quality
- Number of bids the team can pursue
- Revenue influenced by improved capacity

Leading indicators move before business outcomes: cause, then evidence.

CHAPTER 07 · DO THIS

Make metrics part of the cadence

01

Define what success looks like before rollout, not after.

02

Baseline your current state so improvement is measurable.

03

Review metrics on a set schedule and feed findings back into training and workflow design.

04

Share visible wins to sustain momentum and counter adoption drift.

Metrics only close the loop if someone reviews them on the same cadence as the rollout. A dashboard nobody checks is not governance, and a win nobody shares does not defend the program at renewal time.

CHAPTER 7 · CHECKLIST

Proving it works

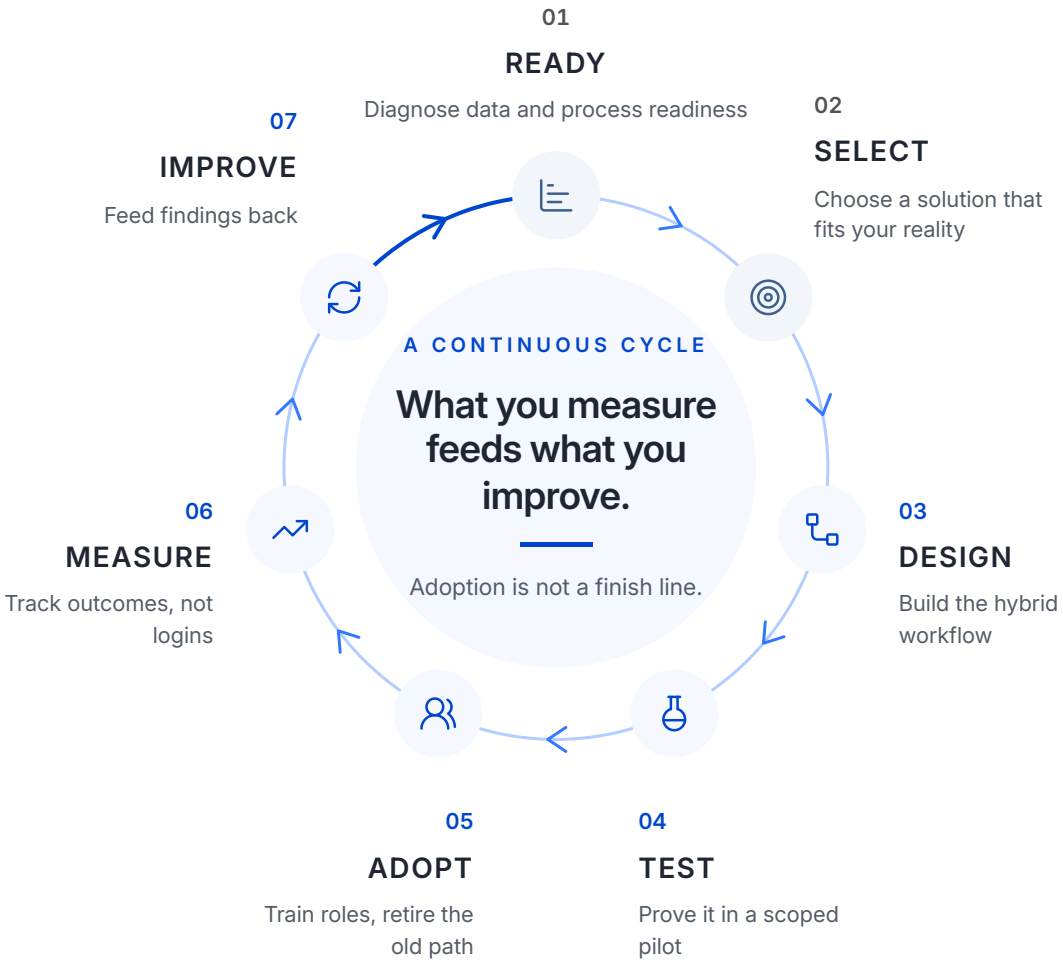
- 01 ☐ We chose one metric each for adoption, efficiency, quality, and outcomes.
- 02 ☐ Every participation metric is paired with an outcome metric.
- 03 ☐ We defined what success looks like before rollout, not after.
- 04 ☐ We baselined the current state, including a dated Part 1 readiness score.
- 05 ☐ We watch leading indicators early, not just win rate.
- 06 ☐ We measure continuously through the end of the reinforcement cadence, not just at milestones.
- 07 ☐ Metrics are reviewed on a set schedule and fed back into training and design.
- 08 ☐ We share visible wins to sustain momentum.

From planning to performance

Make AI real in your bids and proposals. Govern it, measure it, and keep improving.

ONE LOOP, TWO GUIDES

The arc of a successful AI journey is consistent. Part 1 covered readiness and selection; this guide carries the loop through design, testing, adoption, measurement, and improvement.



Get the data and organization ready, choose a solution that fits your reality, and redesign the work into hybrid workflows of deterministic processes, generative AI, and human judgment. Prove it in a small pilot, train people in the role, then govern, measure, and improve.

Notice how little of that is about the tool. Most AI projects fail not because models are weak, but because the work around them never gets redesigned. AI rewards teams with existing discipline in proposal work, and exposes those without it.

BRINGING IT TOGETHER

Where to go from here

This week, without spending anything: three things, roughly a half day in total, and none of them require a vendor, a budget, or a decision.

THREE THINGS TO DO THIS WEEK

01

Run the Part 1 assessment as a working session

Three to five people, scored independently, then compared. You finish with a baseline, a lowest pillar, and a specific first project. If you do only one thing from this guide, do this one.

02

Name your first use case

Test it against the five criteria in Chapter 4 of this guide. For most teams it will be compliance matrix construction, and for most teams that will be the right answer.

03

Put the four expectations in front of leadership, in writing

The cheapest work in the guide, the most frequently skipped, and the one that prevents the “it isn’t working” conversation in month two.

Those three steps put you ahead of most organizations that have already bought something.

Almost everything determining your outcome is already within your control, and most of it can start this week without a purchase order.

The webinar series

This guide is the companion to our two-part series on putting AI to work in bid and proposal organizations.

Session one · Readiness and selection

Working through the four pillars against real libraries, and the vendor conversations that separate fit from features. Part 1 of this series, with live examples and time for your questions.

Session two · Workflow, adoption, and making it stick

Designing the three layers, scoping a pilot, and the change management that survives the end of the pilot cadence. This guide, end to end.

Both sessions are recorded, and registering gets you the recording whether or not you attend live.

When you are ready to talk to a vendor

Talk to us, but use this guide when you do. Part 1 asks you to push vendor claims until you get specific answers, to run the demo on your own messy documents rather than a curated sample file, and to establish where the AI gets its answers and whether it can cite them. Those questions were not written for other vendors. Apply them to VisibleThread.

A useful first conversation looks like this. Bring a real solicitation, ideally a long and badly formatted one. Bring your Part 1 readiness result, particularly your lowest pillar, because it tells us what would actually help and what would not. Bring your hardest constraint, whether that is an authorization requirement, an export control restriction, or a flowdown obligation, since it is better for both of us to find a blocker in the first meeting than the fifth.

We will show you the tool against your document, not ours. If your readiness score says foundation work should come before a purchase, we will tell you that, because a customer who deploys before they are ready becomes a churn statistic and a bad reference.

Register for the series →

Book a working session →

IN CLOSING

The organizations that succeed with AI are not the ones that started earliest.

They are the ones that were ready when they started.

AND IF YOU ARE NOT READY

That is a legitimate outcome of reading this, and a more valuable one than a purchase made too early.

If the Part 1 assessment puts you in foundation-first territory, the highest-return work available to you has nothing to do with AI. Structure the library. Name the owners. Document the process. That work pays for itself in proposal quality and cycle time whether or not you ever deploy a tool, and it is the difference between AI amplifying your capability and AI amplifying your disorder.

ABOUT THE AUTHOR

Questions about your own environment? Bring them to Allison.

A certified APMP member and frequent GovCon speaker, Allison helps some of the largest government contracting teams turn strategic relationships into wins. She wrote this guide from the same conversations she has every week with proposal and capture leaders working through exactly this readiness question.

ALLISON RITZ

Director of Product Marketing,
VisibleThread · CF APMP

BI-WEEKLY LIVE SERIES · HOSTED BY ALLISON

VisibleThread Office Hours

Open, no-fluff sessions on winning more government work: AI, better proposals, and smarter pursuits. Come with questions, leave with answers.

Every other Thursday	30 minutes · live Q&A	Free · replays after

Save your spot for office hours »

Sources: Deepchecks LLM evaluation benchmarks, 2026 (relative error risk, Chapter 3) · Zylo SaaS management index (employee-led purchases and unused licenses, Chapters 5 and 6) · AppStudio (post-deployment adoption, Chapter 6) · The defense contractor pilot in Chapter 6 is an anonymized VisibleThread implementation, not independently published research.