CHRISTOPHER DALY / WHITE PAPER

Building organizational AI capability

A practical approach to incremental AI adoption

Christopher Daly
Independent software and AI consultant
Draft for organizational leaders · November 2026

Executive overview

An organization can build substantial AI capability by repeatedly solving small, recognizable problems. A tool that turns an email into a checked request record can later support a response assistant. The same document lookup capability can serve customer support, proposal preparation, and employee onboarding. As useful components accumulate, the cost and difficulty of attempting the next improvement can fall.

The practical starting point is a question: **what is holding you back?** Follow the work closely enough to identify a task that is straightforward but consumes disproportionate time. Choose the simplest worthwhile improvement whose risks the organization can accept. Give the technology a specific job, measure the result in everyday use, and retain the parts that prove useful.

This approach creates two assets together. The organization gains operational capabilities it can reuse, and its people gain direct experience of where those capabilities are dependable. A skeptical team can develop confidence through its own results. Leadership can authorize a bounded commitment with a visible outcome and decide separately whether to expand.

> **The investment principle**
>
> A first project should deliver an immediate improvement **and leave behind something the next project can use**.

The following sections explain that proposition through a detailed service-business example, a method for selecting opportunities, an operating model, and an illustrative economic assessment.

1 Where AI fits

Traditional business software works particularly well when information arrives in agreed fields and the rules are explicit. Much organizational work happens before that point: someone reads an email, interprets an attachment, works out what the sender means, finds the relevant records, and prepares an entry that another system can process.

AI creates new ways to bridge this gap between informal information and structured work. A language model can be asked to extract requested services from a message, compare a proposal with a set of requirements, or prepare an explanation using supplied records. Ordinary software can then check identifiers, calculate totals, enforce permissions, and carry out an approved action.

For example, an incoming message may describe a familiar request in unfamiliar words. An AI component proposes the request type and relevant details. A database lookup checks the customer record, a rule identifies required information, and a staff member confirms the prepared case. Each part has a distinct job.

This division makes useful applications possible inside familiar tools: email, document repositories, spreadsheets, customer records, and task queues. Buying access to a model is only one part of the work. The practical application also needs the right information, a place in the existing process, and a clear definition of an acceptable result.

What the evidence shows

In a study published in *The Quarterly Journal of Economics* in 2025, Brynjolfsson, Li, and Raymond examined AI assistance for 5,172 customer-support agents. Issues resolved per hour increased by 15% on average. Gains differed substantially across workers, with less experienced workers benefiting more. This was assistance within a particular support operation, with people remaining responsible for the conversation. 1

A randomized experiment with 758 consultants found that, across 18 tasks within the tested model’s capabilities, AI users completed work 25.1% more quickly and completed 12.2% more tasks. On a separate task selected to be outside those capabilities, the combined AI groups were 19 percentage points less likely to reach the correct solution. The study tested GPT-4 and was published in *Organization Science* in 2026. 2

These results support a practical inference: useful gains are achievable, and the task, workflow, and people involved determine their size. The percentages belong to those studies. A specific organization needs its own baseline and evidence for the application it intends to use.

2 Choosing the first target

Begin with an outcome leadership already values: shorter response times, fewer incomplete submissions, more proposals completed, or less administrative burden on specialists. Then examine the work that contributes to that outcome.

Take a sample of recent cases and reconstruct what happened. Identify who touched each case, what they needed, which systems they consulted, and where the time went. Separate active work from waiting. A six-minute preparation task and a two-day approval queue are different constraints; improving one does not automatically remove the other.

Ask the people doing the work to identify recurring effort that is easy to describe but tedious to carry out. Examples include finding the same information in differently formatted documents, comparing two versions, checking whether a submission contains required material, or assembling facts before someone can make a decision.

Compare the candidates

A useful opportunity record answers six questions:

Consideration	Evidence to gather
Business outcome	What gets better, who benefits, and who would use the released capacity
Workload	Case volume, active time, waiting time, and current rework
Inputs and output	Real examples of inputs and an acceptable output
Error visibility	How a mistake would be recognized: required checks, source material, and reviewer expertise
Consequences	Who is affected by an error, how it is detected, and whether it can be reversed
Dependencies	Data access, integration, process changes, ownership, and ongoing effort

Use these answers to compare nearby alternatives. Automatically issuing a customer quote may offer substantial savings but also create pricing and contractual exposure. Preparing the information a staff member needs to issue that quote may offer a smaller benefit with a much easier boundary to manage.

Risk can disqualify a candidate even when its expected benefit is large. Look for an adjacent task that advances the same goal with an acceptable downside. Keep the larger opportunity visible and record what would need to change before reconsidering it.

The aim is to find a good next move without conducting an exhaustive survey of every possible use of AI. Revisit the search after each improvement: the organization’s constraints, available components, and experience will have changed.

3 Customer request intake

Consider a fictional service business that receives requests through email and attachments. Staff read each request, establish the customer and location, search for previous work, identify missing information, and prepare a case for a coordinator. The coordinator decides what to offer and when the work can happen.

The business wants to respond more quickly. Its first target is preparing the case. Pricing, scheduling commitments, and customer communication remain with existing staff while this initial capability is evaluated.

The first useful output

A representative fictional request reads:

> *Could you arrange another visit for our North office next Friday? Same issue as in March. I have attached the updated equipment list.*

A useful system prepares a record like this:

Field	Prepared record
Requested work	Follow-up visit, taken from the message
Location	“North office,” preserved as the customer wrote it; exact site requires confirmation
Requested date	“Next Friday,” preserved for confirmation against the message date and scheduling context
Previous work	A March visit is referenced; the specific job is not yet identified
Attachment	Equipment list received, with a link to the original
Missing information	Exact site and prior job reference, according to the team’s agreed intake requirements
Suggested next action	Staff confirm the account and resolve the missing details before scheduling

The record makes the preparation work visible. A person can inspect the proposed fields alongside the source, correct them, and accept the result. The system distinguishes what the customer supplied, what a business record establishes, and what still needs confirmation.

Producing the output

The initial implementation can be narrow. Staff select an eligible request or forward it to an approved intake queue. The application stores a reference to the original message and reads the permitted attachments. A model proposes structured fields. Conventional validation checks required fields and allowed values. A review screen presents the proposed record, source excerpts, and unresolved items.

Customer identity requires particular care. The model may suggest a match, but an ambiguous name should produce candidate records for review. A confirmed customer identifier then constrains searches for prior work. This prevents a plausible similarity from silently becoming an authoritative account association.

The first version may simply return a prepared form for staff to use. Once the preparation proves useful, a confirmed record can be written into the existing case system through an ordinary integration. Separating these steps allows the organization to assess the value of the preparation before investing in a deeper connection.

At this point the business has built several things it can retain: an approved route for reading requests, a defined case format, extraction instructions, validation rules, a review interface, and a collection of examples that establish acceptable behavior.

Adding the next capability

Suppose staff also spend time finding the service guidance that applies to a request. A second application searches an approved collection of current documents and returns relevant passages with their source, version, and access restrictions. Staff use it to prepare answers to internal questions.

Once that lookup works in its own context, it can be combined with the intake record. The system now has both a description of the request and relevant business guidance. A further component can draft a response that identifies missing information or explains the next step. A staff member approves the message before it is sent.

The growing workflow is concrete:

1. Read the request and prepare its facts.
2. Confirm the customer and resolve ambiguity.
3. Retrieve the applicable information.
4. Prepare a response from those inputs.
5. Let the responsible person approve the communication.

Each addition has a useful result of its own. The organization can retain intake preparation even if it decides that response drafting adds little value. It can use the lookup capability elsewhere even while customer-facing work remains entirely manual.

4 Building reusable modules

An isolated automation solves one problem. A reusable capability exposes a dependable way to perform a recognizable piece of work. A module groups related capabilities behind a stable input and output. A workflow connects modules to achieve a business result.

In the service example, an intake module could accept a message and return a checked case record. An evidence module could accept a question and an authorized document collection, then return relevant source passages. A response module could use both outputs to prepare a message for review.

The distinction matters because a later application can reuse the same intake format, identity checks, document access, and review behavior. It adds the genuinely new parts of its task.

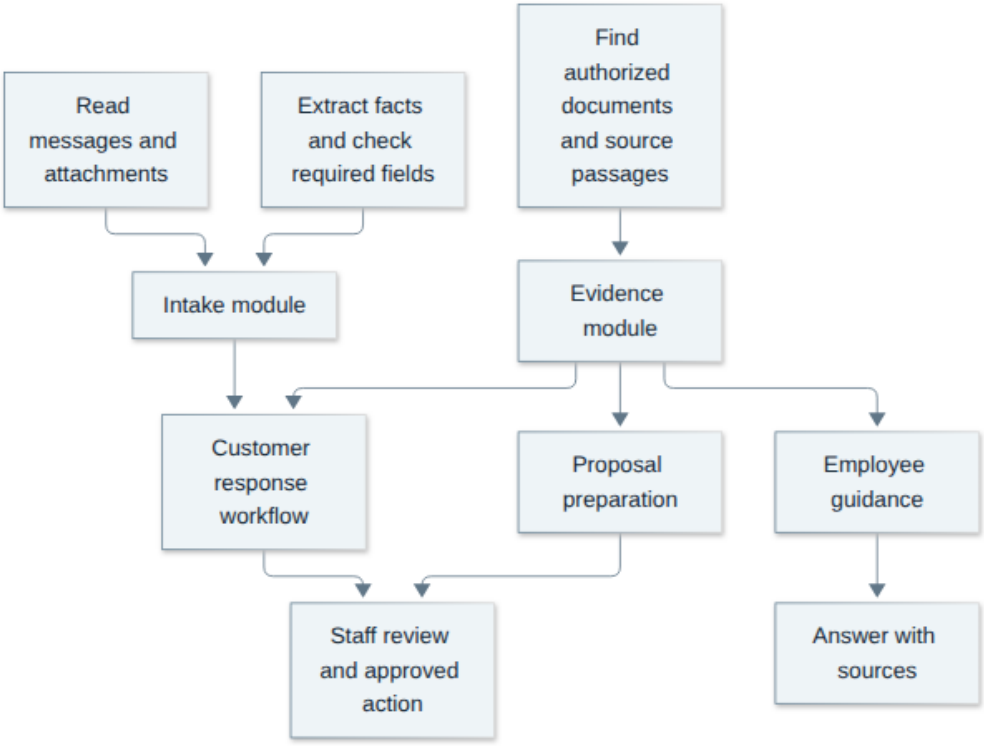


Figure 1. Reusable modules supporting several workflows. This is a possible architecture; each connection requires evaluation in the workflow where it is used.

Requirements for reuse

Reuse depends on a few concrete engineering choices:

Property	Practical requirement
Defined input and output	A request module returns named fields and unresolved items that another component can interpret
Preserved evidence	Extracted facts retain links to messages or documents, so reviewers can check their basis
Explicit authority	Reading a case, drafting a message, approving it, and sending it are separate permissions
Recognizable failure	Missing documents, ambiguous identities, and unavailable services produce visible states
Repeatable evaluation	Known examples can be rerun when prompts, models, rules, or source documents change
Ownership and versioning	Someone maintains the component and knows which workflows depend on it

The implementation can mix AI and conventional software. A model may interpret a request; a database establishes the account; rules check required fields; a calculation uses approved prices; an integration updates the record. This keeps each responsibility explicit.

Anthropic's engineering guidance describes the same useful distinction between workflows with predefined steps and agents that choose their own actions. It recommends starting with the simplest architecture that meets the need and adding complexity when warranted. This supports the architectural choice; it is not evidence of a particular business return. 3

Reducing the next project's effort

The next project may reuse five kinds of work: connections to existing systems, standardized information, tested capabilities, evaluation examples, and staff familiarity. It also benefits from decisions already made about data access, approval, and support.

For example, a proposal assistant can reuse document lookup and source citations while adding a requirements comparison and a proposal-specific review. It does not have to recreate every connection or establish every operating practice from the beginning.

This benefit requires deliberate reuse. A collection of unrelated subscriptions, private prompts, and duplicate document copies can increase maintenance without creating shared capability. Consolidate repeated functions when their common purpose becomes clear. Start with working components, then give the recurring patterns a stable form.

How the opportunities change

Initially, "prepare a complete customer response" may involve too much unknown work. After intake, document lookup, and review mechanisms exist, it becomes a smaller project focused on connecting and evaluating those pieces.

This is the strategic value of the ecosystem. The organization is improving the conditions under which future improvements become possible. A slightly smaller opportunity can be the better first investment when it delivers useful value and develops a component needed by several later opportunities.

That future value should remain identifiable. Name the plausible reuse and the new work still required. Avoid treating every first project as the foundation of an unspecified transformation.

5 Evaluating the result

The burden of proof belongs on the proposed application. Leadership should be able to see what it does, how the result compares with current practice, and what level of use the evidence supports.

For the intake example, a test should assess the entire assisted process. Start timing when the staff member begins handling the request and stop when the case is ready for the next person. Include opening the tool, reading its output, checking sources, making corrections, and transferring the result.

Use a representative set of cases, then test unusual cases deliberately. Include missing attachments, conflicting details, unfamiliar wording, duplicate requests, unreadable files, and ambiguous customer names. Keep a separate set of examples that was not used to tune the application.

Measure useful outcomes

An illustrative evaluation plan for an application that retains staff review would examine:

Dimension	Evidence to collect
Preparation time	Median and slower-case handling times, plus total staff time across comparable cases
Record quality	Missing required facts, invented details, incorrect account links, and substantive corrections
Useful coverage	Proportion of eligible requests successfully assisted; include fallbacks and abandoned attempts
Reviewer burden	Time spent verifying the output and whether staff can locate the evidence they need
Downstream effect	Coordinator rework, total response delay, and errors that survive the review
Operational effort	Tool charges, maintenance time, incidents, and time spent updating documents or examples

Agree the decision criteria before interpreting the result. For this example, leadership might require a meaningful reduction in preparation time, no increase in downstream rework, and an acceptable process for uncertain cases. Critical identity errors require individual investigation even when aggregate accuracy looks good.

Compare similar work and account for differences between staff members. Where practical, alternate assisted and unassisted cases or use comparable groups. A before-and-after improvement can also reflect changes in workload or training, so record those changes when drawing conclusions.

A trial supports a specific decision: assisted use for these request types, by these people, with these checks. A small trial that finds no serious errors does not establish the safety of unattended operation at much larger volume.

Test the connections

A correct extraction can still lead to a poor response if the lookup uses outdated guidance. A good document search can expose the wrong material if permissions are applied incorrectly. The complete workflow therefore needs cases that exercise its handoffs, access boundaries, and recovery behavior.

When an error appears, retain the input and expected outcome so it can become a future evaluation case. These examples accumulate into an organizational asset: a record of what good work looks like and where the system has needed correction.

NIST's AI Risk Management Framework places context, measurement, management, and continuing governance across the system's life cycle. The practical implication here is to repeat the relevant checks when the application, source material, or use changes. 4

6 Economics and review effort

The following numbers are assumptions for the fictional intake example, not observed client results, market prices, or a consulting quotation. Their purpose is to show which facts a decision-maker needs.

Assumption	Value
Eligible requests	800 per month
Current preparation	6 minutes per request
Assisted preparation	2 minutes per eligible request, including checking, correction, and manual fallback
Staff capacity value	$50 per hour, including employment overhead
Upkeep	4 staff hours per month
Software and processing	$250 per month
Initial setup	$6,000, including integration, testing, and internal staff effort

Current preparation consumes 80 staff hours monthly. Assisted preparation consumes 26.7 hours. After allowing four hours for upkeep, the net capacity released is approximately 49.3 hours. Valued at $50 per hour and reduced by the $250 operating cost, the monthly capacity value is approximately $2,217.

The calculation has three steps:

1. **Gross hours saved** = eligible requests × minutes saved per request ÷ 60.
2. **Net hours released** = gross hours saved − upkeep hours.
3. **Net capacity value** = net hours released × value per staff hour − software and processing costs.

Sensitivity to assisted preparation time

Assisted time includes checking, correction, and fallback. Net hours are monthly and include upkeep; net capacity value also deducts software and processing costs.

Assisted time per request	Net hours per month	Net capacity value per month
2 minutes	49.3	$2,217
4 minutes	22.7	$883
5.5 minutes	2.7	−$117

All other assumptions remain fixed. At 5.5 minutes, the recurring costs exceed the value of the time released. The organization could narrow the task, improve the review interface, or choose another target.

If total initial setup, integration, testing, and internal staff effort were valued at $6,000, the two-minute scenario would recover that amount in about 2.7 months on this capacity-value basis. The four-minute scenario would take about 6.8 months.

These are capacity calculations. Cash savings require an actual change in expenditure, such as reduced overtime or external processing costs. Additional revenue requires demand and an ability to use the released time productively. A team should identify what it will do with the capacity before treating it as a realized return.

Eligible volume matters too. Under the two-minute assumptions and fixed monthly costs, approximately 135 requests per month cover recurring costs, while 285 cover those costs plus recovery of the $6,000 setup over twelve months. Lower actual usage or higher integration effort changes the decision.

The same discipline applies to later modules. Evaluate each addition against the workflow that already exists. Do not count the intake time saving again when estimating the benefit of response drafting.

7 Applications across the organization

The service-business example has wider relevance because many organizations repeatedly interpret information, locate evidence, compare it with requirements, prepare a result, and seek approval. The application changes; parts of the work recur.

Proposals and tenders

A professional-services team could begin by extracting requirements from a request for proposal into a table linked to the original passages. Staff confirm the requirements and mark which ones need a response.

A later module could locate approved case studies, staff qualifications, and previous answers. Combining the requirements and evidence could produce a first proposal draft with unresolved claims clearly identified. The same comparison capability could check the finished draft for omissions.

The new capability is a traceable preparation process: each requirement has an owner, supporting material, and a response status. Final commitments and representations remain subject to the team’s approval.

Purchasing and supplier comparison

An operations team could start by placing supplier quotes into a common comparison format. It would extract product descriptions, quantities, currencies, delivery terms, and exclusions while preserving source references.

Once the extraction is useful, existing calculation rules could normalize quantities and compute comparable totals. A requirements module could flag differences from the requested specification. Staff would then receive an exception list and a comparison to review.

This reuses extraction, validation, comparison, and evidence handling. The organization adds supplier-specific formats and purchasing rules. Unit conversions and commercial terms need explicit handling; superficially similar line items may describe different products or obligations.

Grants and program administration

A nonprofit could first check whether an application contains the required documents and dates. A second capability could match a grant’s requirements to approved program descriptions and existing evidence.

Together, those modules could prepare a submission checklist, locate missing supporting material, and assemble a first narrative for staff review. After an award, the same requirements and evidence records could help prepare reporting material.

Funding eligibility, impact claims, and final submissions still need accountable review. The initial opportunity is to reduce the work of gathering, organizing, and checking what supports those decisions.

Management reporting

Once intake and review create consistent records, leadership can ask more useful questions: which information is repeatedly missing, where cases wait, which requests require the most rework, and what guidance staff struggle to find.

Conventional reporting can calculate counts and durations from the records. AI can help organize recurring free-text explanations and draft a narrative linked to those results. The benefit may be a better next investment decision: a form change, a clearer policy, staff training, or another automation.

These applications illustrate how the ecosystem can extend beyond producing documents. It can improve the information available for running the organization and reveal opportunities that were previously obscured by scattered, inconsistent records.

8 Operating and expanding

Incremental adoption works when each step has a place in daily operations. A pilot needs a process owner who can define acceptable work, staff who will use it, and technical responsibility for keeping it running. One person may hold more than one role in a small organization, but the responsibilities still need to be clear.

Start with the smallest adequate implementation. An existing approved product may already provide the required capability. A custom workflow may be justified when the value depends on particular records, integrations, or review behavior. Evaluate those routes against the same task and examples.

For the first application, establish which data it may read, where that information is processed, what gets retained, and which actions require approval. The ability to prepare a record does not itself grant permission to send messages, change commitments, or move money. Permissions should match the job being tested.

Treat incoming documents and messages as material to process. Instructions embedded in them should not gain authority to change the workflow or request unrelated actions. Enforce access restrictions and approval requirements in the application, rather than relying solely on instructions to the model. OWASP describes this risk as indirect prompt injection and recommends restricting privileges and protecting consequential actions. [5]

The implementation agreement should make the handover concrete: who can maintain the integration and instructions, how records and evaluation examples can be exported, who receives failure reports, and what the organization can continue using if it changes providers.

Make expansion a separate decision

First, establish whether the capability produces useful results on historical examples. Next, observe how staff use it on live work with the agreed review. Then make it a supported part of the process. At each stage, decide whether to continue, adjust, or stop.

Only after that evidence is available should leadership consider expanding to additional request types, sharing the component with another team, or permitting greater autonomy. Each changes something important: the inputs, the users, the consequences, or the operating conditions.

People can become comfortable at different speeds. Give them access to the sources, a way to correct results, and a route for reporting problems. Their experience improves both the application and the organization’s understanding of where it belongs.

As the pool of modules grows, maintain a simple register of their purpose, owner, inputs, outputs, dependencies, evaluation cases, and running costs. Shared components deserve particular attention because a change can affect several workflows. Introduce a revision to one bounded use first, then extend it once the relevant checks pass.

9 The first decision

A useful first decision is to authorize an examination of one recurring operational constraint. Choose a process that matters, involve the people who know it, and bring representative examples.

The resulting proposal should identify the simplest promising target, the information it needs, the result it would produce, the effort of checking it, and the boundaries of the initial trial. It should include an economic range based on actual volume and handling time, plus a clear account of what the organization would retain.

It should also identify one or two plausible next uses of the resulting capability. That makes the longer-term opportunity concrete while leaving the first project responsible for delivering its own value.

The governing question remains: **what is holding us back now, and what is the simplest useful improvement we can make?** Over time, the answer changes because the organization has acquired more working parts. That is how a modest first application can become the beginning of a much broader capability.

References

1. Brynjolfsson, E., Li, D., and Raymond, L. (2025). *Generative AI at Work*. The Quarterly Journal of Economics. Read source
2. Dell'Acqua, F., et al. (2026). *Navigating the Jagged Technological Frontier*. Organization Science. Read source
3. Anthropic (2024). *Building Effective Agents*. Engineering guidance. Read source
4. National Institute of Standards and Technology (2023). *AI Risk Management Framework 1.0: AI RMF Core*. Read source
5. OWASP (2025). *LLM01: Prompt Injection*. Top 10 for LLM Applications. Read source

About the author

Christopher Daly is an independent software and AI consultant whose work includes AI applications, architecture, integration, and application modernization. He helps organizations identify useful opportunities and carry them through planning, implementation, and handover.

cdaly.me · Contact Chris