

BEYOND AUTONOMOUS DEMOS

Bound the Decision Before You Build the Agent — A Method for Enterprise Agentic Workflows

A six-stage operating method for turning 2024's autonomous demonstrations into controlled, accountable enterprise services

Giovanni Leonardi

March 2024



Contents

1. Executive Summary	4
2. Why Agentic Delivery Needs a Different Method	5
3. Roles and Decision Rights	5
4. Stage One — Frame the Outcome and Test Agent Fit	6
4.1 Purpose	6
4.2 Inputs	6
4.3 Activities	7
4.4 Artefact: Outcome and Agent-Fit Brief	7
4.5 Gate: Agent-Fit Gate	7
5. Stage Two — Map the Decision Chain	8
5.1 Purpose	8
5.2 Activities	8
5.3 Artefact: Decision and Consequence Map	8
5.4 Gate: Decision-Clarity Gate	9
6. Stage Three — Bound Authority and Design the Agent Contract	9
6.1 Purpose	9
6.2 Authority Levels	9
6.3 The Agent Contract	10
6.4 Artefacts	10
6.5 Gate: Authority Gate	10
7. Stage Four — Build the Workflow and Its Evidence	11
7.1 Purpose	11
7.2 Architecture Components	11
7.3 Evaluation Pack	11
7.4 Worked Example: Invoice Exceptions	12
7.5 Artefacts	12
7.6 Gate: Evidence Gate	13
8. Stage Five — Release Through Controlled Autonomy	13
8.1 Purpose	13
8.2 Release Ladder	13
8.3 Operational Readiness	13
8.4 Artefact: Release Dossier	14
8.5 Gate: Release Gate	14
9. Stage Six — Operate, Learn and Expand Deliberately	14

9.1 Purpose 14

9.2 Monitoring Rhythm 14

9.3 Expansion Rules 14

9.4 Artefacts 15

9.5 Gate: Continuation Gate 15

10. Governance Cadence 15

11. The Minimum Viable Artefact Set 16

12. Final Practitioner Test 16

Bound the Decision Before You Build the Agent — A Method for Enterprise Agentic Workflows

Transformation · No. 391

Originally written March 2024 | Re-edited and re-rendered for the WhereBeyond Edition,

“The agent may act only inside authority that the organisation can state, test, observe and revoke.”

1. Executive Summary

Agentic workflows are moving artificial intelligence from producing an answer to pursuing an outcome. A large language model can be connected to enterprise data, given tools, asked to plan intermediate steps and allowed to act across a process. This creates genuine potential: less manual coordination, faster exception handling and services that respond to context rather than fixed rules.

It also changes the delivery problem. A conventional model makes a prediction that a person or system consumes. An agent can decide what information to seek, which tool to call, when to retry and whether to continue. The programme is no longer deploying only a model. It is delegating part of a decision process.

The method in this paper begins with that delegation, not with the agent architecture. It uses six stages:

1. Frame the outcome and test agent fit.
2. Map the decision chain.
3. Bound authority and design the agent contract.
4. Build the workflow and its evidence.
5. Release through controlled autonomy.
6. Operate, learn and expand deliberately.

Each stage produces a defined artefact and ends with a gate. The governing principle is simple: the agent may act only inside authority that the organisation can state, test, observe and revoke.

The method is deliberately cautious about autonomy and practical about value. It does not require every action to remain human-approved. It requires autonomy to be earned at the level of a decision, not granted to a technology in general.

2. Why Agentic Delivery Needs a Different Method

Most enterprise AI delivery methods assume a bounded output: a score, classification, forecast or generated draft. The system is assessed, integrated and monitored. Human users decide what to do next.

An agentic workflow can cross that boundary. It may interpret a goal, decompose work, select tools, retrieve information, write to a system, communicate with a user and evaluate whether the task is complete. Even when each tool call is technically constrained, the sequence can produce consequences nobody specified step by step.

This creates four delivery differences.

- The unit of risk is the decision path, not the individual output. A reasonable action can become unsafe in combination with previous actions.
- Authority becomes executable. A vague process permission turns into credentials, tool access and thresholds.
- Failure can propagate. An incorrect assumption may trigger several downstream actions before a person sees it.
- Operations shape behaviour. Changes to prompts, tools, source data, policies or user requests can alter performance without changing the underlying model.

A demonstration conceals much of this. It uses curated inputs, cooperative users and reversible actions. Enterprise operation contains ambiguity, incomplete records, adversarial instructions, unavailable systems and pressure to continue.

The method therefore treats an agent as a new operational actor with bounded authority, not as a clever interface placed over existing systems.

3. Roles and Decision Rights

The programme should establish the following roles before design. One person may hold more than one role in a small implementation, but no accountability should be left implicit.

Role	Accountable for	Cannot delegate
Outcome owner	Business value and acceptable operational consequence	The decision to use an agent for the outcome
Process owner	End-to-end process, exception policy and service measures	Authority boundaries inside the process
Agent product owner	Workflow behaviour, backlog and evidence across releases	Acceptance of the integrated agent service
Engineering lead	Orchestration, tools, identity, logging and resilience	Technical control of actions
Model lead	Model selection, prompts, retrieval and evaluation	Known model limitations
Risk and control lead	Privacy, security, compliance and control design	Independent challenge of residual risk
Operations lead	Human supervision, incident handling and service readiness	Ability to stop or restrict operation
Assurance lead	Gate evidence and traceability	Independence of release recommendation

The outcome owner approves the business delegation. The process owner defines what decisions exist. The product and engineering roles design how the agent works. Risk and assurance challenge the evidence. Operations owns what happens at 02:00 when the workflow behaves unexpectedly.

No architecture can compensate for a missing process owner.

4. Stage One — Frame the Outcome and Test Agent Fit

4.1 Purpose

Begin with an operational outcome whose decision burden is understood. Do not begin with a general ambition to “deploy agents” or with a list of available tools.

Agentic design is appropriate when the work contains variable sequences, requires information from several sources and benefits from context-sensitive judgement. It is less appropriate when a fixed rule, ordinary workflow automation or a single model call can deliver the result more predictably.

4.2 Inputs

- strategic or operational outcome

- current process measures
- volume, variation and exception data
- existing automation and system constraints
- consequence of delay and error

4.3 Activities

Define the task in the form: For this class of request, reduce or improve this operational measure by enabling these decisions.

Then compare four solution patterns:

Pattern	Use when	Warning
Rule or workflow	Steps and conditions are stable	Do not add a model for novelty
Model-assisted task	One prediction or draft improves a human decision	Keep action with the user
Human-led agent	Sequence varies but material actions need approval	Design approval latency into benefits
Bounded autonomous agent	Repeated decisions are measurable, reversible and controllable	Autonomy must be evidenced per action

Estimate value after supervision, exception handling, control and maintenance. If the agent saves ten minutes but creates six minutes of review, model the net four.

4.4 Artefact: Outcome and Agent-Fit Brief

The brief contains:

- outcome, baseline and target
- task boundary and excluded cases
- decision owners and affected parties
- options considered and why an agent is justified
- expected volume, benefit and continuing cost
- highest-consequence failure
- proposed starting autonomy level

4.5 Gate: Agent-Fit Gate

Proceed only if the process owner can name the decision the agent improves, a simpler option has been considered, and value remains plausible after control costs. Otherwise

redesign the use case.

5. Stage Two — Map the Decision Chain

5.1 Purpose

Make the hidden decisions inside the workflow visible. “Resolve an invoice exception” sounds like one task. It may contain identity matching, policy interpretation, evidence assessment, supplier communication, financial approval and system posting.

The decision chain is the true architecture of the agent.

5.2 Activities

Walk representative cases from trigger to closure. For every step, record:

- information required and its source
- judgement or rule applied
- action available
- owner of the current authority
- consequence if wrong
- reversibility and time to detect
- known exceptions
- hand-off to a person or system

Include normal, boundary and hostile cases. A boundary case is legitimate but unusual. A hostile case includes misleading input, instruction embedded in a document or a user attempting to exceed authority.

5.3 Artefact: Decision and Consequence Map

Use one row per decision.

Decision	Evidence	Allowed action	Consequence	Reversible	Current owner
Match invoice to order	Supplier, amount, order reference	Propose match	Wrong liability	Yes	Accounts payable

Decision	Evidence	Allowed action	Consequence	Reversible	Current owner
Accept quantity variance	Receipt and tolerance policy	Approve below threshold	Financial loss	Limited	Budget holder
Request missing evidence	Missing-field rule	Send approved message	Delay or disclosure	Yes	Case handler
Post final status	Completed checks	Update finance system	Accounting record	Controlled reversal	Finance operations

The map should expose decisions that should never be delegated, decisions suitable for recommendation, and low-consequence actions that may be autonomous.

5.4 Gate: Decision-Clarity Gate

Proceed only when every material action has an owner, consequence and recovery route. If the team cannot identify who currently owns a decision, the programme has found an operating-model gap. Resolve it before encoding authority.

6. Stage Three — Bound Authority and Design the Agent Contract

6.1 Purpose

Translate organisational authority into executable boundaries. The agent contract is not a legal document. It is the agreed specification of what the agent may pursue, know, decide, do and communicate.

6.2 Authority Levels

Assign an authority level to each action:

1. Observe. Read approved data and produce no external effect.
2. Recommend. Propose an action for human decision.
3. Prepare. Create a transaction or communication that requires approval.
4. Act with conditions. Execute inside thresholds, with logging and recovery.
5. Act and supervise exceptions. Complete the workflow autonomously, escalating defined cases.

Do not assign one level to the whole agent. An agent may autonomously request a missing reference, prepare a financial adjustment and only recommend the final approval.

6.3 The Agent Contract

The contract records:

- Purpose: the outcome and permitted task class
- Scope: included and prohibited requests
- Identity: service account, user context and permission source
- Knowledge: approved systems, retrieval sources and freshness requirements
- Tools: callable actions, parameters, limits and timeouts
- Decision policy: thresholds, rules and escalation conditions
- Communication policy: recipients, approved channels and disclosure limits
- Memory: what may persist, for how long and for what purpose
- Evidence: logs needed to reconstruct a run
- Stop conditions: triggers that halt one task or the service
- Recovery: reversal, correction and notification process
- Owner: person accountable for changing the contract

Credentials must enforce the contract. A prompt saying “do not approve payments above £500” is not a control if the tool credential can approve £50,000.

A policy stated only in the prompt is guidance; a policy enforced in tools, identity and workflow is a boundary.

6.4 Artefacts

- signed agent contract
- tool and permission register
- authority matrix
- data and memory specification
- escalation catalogue
- abuse and misuse scenarios

6.5 Gate: Authority Gate

The outcome owner, process owner, engineering lead and risk lead jointly approve the contract. No tool is connected until its permission and recovery path are agreed.

7. Stage Four — Build the Workflow and Its Evidence

7.1 Purpose

Build the agent and the means to know whether it works. Evaluation is not a final test phase; it is part of the design.

7.2 Architecture Components

A practical enterprise agent usually contains:

- an instruction layer defining purpose and behaviour
- approved context retrieval
- a planner or orchestration loop
- a tool gateway enforcing permissions and validation
- state for the current task
- human approval and escalation points
- event logging and traceability
- monitoring for quality, cost, latency and incidents

Keep deterministic logic outside the model where possible. Thresholds, mandatory fields, access control and transaction validation should be enforced by ordinary software. Use the model where language or contextual judgement is needed.

7.3 Evaluation Pack

Create a versioned set of cases covering:

- common successful paths
- missing and contradictory information
- rare but legitimate exceptions
- attempts to exceed authority
- instructions embedded in retrieved content
- unavailable or slow tools
- repeated failure and retry behaviour

- changes in policy or source data
- hand-off quality
- recovery after an incorrect action

Measure more than task completion:

Measure	Question
Decision quality	Was the judgement supported by the evidence?
Policy compliance	Did every action remain inside the contract?
Tool accuracy	Were parameters and targets correct?
Escalation quality	Did the agent stop for the right cases with useful context?
Recovery	Could the action be reversed and the affected person informed?
Efficiency	What time and cost remained after supervision and exceptions?
Traceability	Can an independent reviewer reconstruct the run?

7.4 Worked Example: Invoice Exceptions

The team selects 600 historical invoice exceptions: 350 common, 150 boundary, 70 policy exceptions and 30 deliberately hostile or corrupted cases.

The first agent resolves 78 per cent of common cases but attempts to continue after contradictory receipt data in 11 cases. The team does not merely adjust the prompt. It changes the orchestration: conflicting receipt records force an escalation, the tool gateway blocks posting, and the escalation includes the two source values.

A second evaluation resolves 74 per cent of common cases—slightly fewer—but produces zero unauthorised postings and reduces incomplete escalations from 22 per cent to 4 per cent. The programme accepts the lower automation rate because the workflow is safer and easier to operate.

This is the discipline required. Optimise the service, not the demonstration.

7.5 Artefacts

- versioned evaluation pack and results
- architecture and data-flow record
- prompt, retrieval and tool versions

- exception and failure analysis
- control implementation evidence
- proposed operating thresholds

7.6 Gate: Evidence Gate

Release preparation begins only when material decisions meet agreed thresholds, no prohibited action can be reached in testing, traceability is complete and operations accepts the residual exception load.

8. Stage Five — Release Through Controlled Autonomy

8.1 Purpose

Move from controlled evaluation to live consequence without granting maximum autonomy at once.

8.2 Release Ladder

1. Shadow. The agent observes real cases and proposes actions without influencing work.
2. Assist. Users request recommendations and decide every action.
3. Approve. The agent prepares actions; authorised users approve.
4. Bounded act. The agent executes low-consequence actions within thresholds.
5. Expanded act. Authority increases only for decisions supported by live evidence.

Set entry and exit criteria for each level. Time alone is not evidence. A four-week pilot with twelve relevant exceptions may tell less than a two-week pilot with eighty.

8.3 Operational Readiness

Before live action, operations needs:

- a runbook with stop, restrict and recovery procedures
- named on-call and business escalation contacts
- dashboards for outcomes, exceptions, tool failures, cost and latency
- user guidance describing capability and limitations
- incident categories and notification rules
- change control for models, prompts, tools, data and policy

- a manual fallback that has been tested
- a communications plan for affected staff and users

Use canary release where practical: a limited case type, team, geography or time window. Avoid simultaneous expansion of users, tools and authority; otherwise evidence cannot show which change caused an effect.

8.4 Artefact: Release Dossier

The dossier joins the business case, agent contract, evaluation results, operating model, residual risks and release limits. It should state clearly what the agent cannot do.

8.5 Gate: Release Gate

The outcome owner accepts value and residual consequence. Operations confirms readiness and stop authority. Assurance confirms that evidence matches the contract. The release decision specifies the authority level, population, duration and conditions for expansion or rollback.

9. Stage Six — Operate, Learn and Expand Deliberately

9.1 Purpose

Manage the agent as a changing service. Model behaviour, tools, source data, policies and user practices will evolve. Approval at launch is not permanent evidence.

9.2 Monitoring Rhythm

Monitor three horizons.

- Per run: prohibited action, tool error, unusual sequence, escalation and recovery
- Weekly: success, override, exception, cost, latency and user feedback
- Monthly or after material change: decision quality by case group, drift, emerging misuse, realised benefit and authority fitness

Review both false action and false restraint. An agent that acts wrongly creates risk; an agent that escalates everything creates hidden cost and may encourage users to bypass it.

9.3 Expansion Rules

Expand authority only when:

- live cases cover the decision types being delegated
- error and recovery remain inside tolerance
- operations can support the new exception volume
- realised value remains positive after control cost
- no material user group carries disproportionate harm
- the new action has a technical enforcement boundary
- the process owner explicitly accepts the change

Any change to tools, identity, material instructions, model, retrieval source or decision policy triggers impact assessment and targeted re-evaluation. A supplier model update is a service change, not background maintenance.

9.4 Artefacts

- live evidence dashboard
- decision-quality review
- incident and recovery log
- realised-benefit report
- change and re-evaluation register
- current agent contract
- authority expansion or restriction decision

9.5 Gate: Continuation Gate

At an agreed interval, decide to continue, restrict, redesign or retire. An agent that no longer creates net value should not survive because it is technically interesting.

10. Governance Cadence

The method uses three forums, each with a different job.

Forum	Cadence	Decision
Delivery review	Weekly	Workflow defects, evaluation and backlog
Operational control review	Fortnightly during release, then monthly	Incidents, exceptions, thresholds and restrictions
Outcome and authority board	At gates and material changes	Business delegation, release, expansion or retirement

Do not send every issue to the authority board. Do not allow the delivery review to expand authority. Clear forum boundaries prevent both paralysis and informal delegation.

11. The Minimum Viable Artefact Set

A small but material agentic workflow should not need hundreds of pages. It does need a coherent record.

The minimum set is:

- Outcome and Agent-Fit Brief
- Decision and Consequence Map
- Agent Contract
- Tool and Permission Register
- Evaluation Pack and Results
- Operations Runbook
- Release Dossier
- Live Evidence Dashboard
- Change and Re-evaluation Register

These artefacts should reference one another. A tool in the register supports an action in the decision map, is bounded by the contract, tested in the evaluation pack and monitored in operation. Traceability is more valuable than document volume.

12. Final Practitioner Test

Before approving an agentic workflow, ask a person outside the delivery team to answer ten questions from the evidence:

1. What outcome is the agent pursuing?
2. Which decisions can it make?
3. Which actions can it execute?
4. What is it prohibited from doing?
5. Whose authority is being delegated?
6. How is that authority technically enforced?
7. When must it stop and ask for help?

8. Can a material action be reconstructed and reversed?
9. Who can restrict the service immediately?
10. What live evidence would justify more autonomy?

If the answers are vague, the agent is not governed, however sophisticated its architecture.

Agentic workflows can create real enterprise value in 2024, but autonomy should be treated as a sequence of explicit delegations. Start with the decision, encode the boundary, build the evidence, release gradually and keep the authority revocable.

The aim is not an agent that can do everything. It is an agent that can do the right things, inside a system that remains accountable when it does not.