

AI and Power Platform Governance

Freedom to Build, Confidence to Scale



Ray Rasmussen

Managing Principal, rGen Consulting

September 25, 2026



Table of Contents

Easier building, higher stakes	1
Business decides, technology enforces	2
Make the safe path the easy path	3
Start with what you already have	5
The governance profile	6
Governing the lifecycle: one example.....	8
Build: define exactly what will be released	8
Validate: prove what it does and what it must not do	9
Authorize: make the release decision explicit	9
Operate: measure quality, and be ready to stop	10
Retire: retire the workflow, not the history or shared services	10
Making it repeatable: the Center of Excellence	11
What leaders should do now	12
And in the end... ..	13
Sources.....	14

AI and Power Platform Governance: Freedom to Build, Confidence to Scale

Ray Rasmussen | rGen Consulting | September 25, 2026

Easier building, higher stakes

AI and low-code tools promise better work: faster decisions, less friction, and more capacity for what matters. Microsoft Copilot and Power Platform put that promise directly in the hands of business teams, often without IT in the room. That is both the opportunity and the exposure. When anyone can build an agent that reads customer data or updates a business record, the question shifts from *can we build it* to *should we build it, who owns it, and how will it be controlled* from the first prototype through final retirement.

Without governance, solutions can proliferate without adequate oversight, sensitive data can flow to unapproved destinations, apps can outlive their owners, and costs can become visible only after they have accumulated. None of these problems are solved by saying no. Instead, governance must decide in advance what is allowed, where, and under whose accountability.

Effective governance joins two disciplines. Business-led governance defines purpose, acceptable risk, and where people are accountable. Technology-led governance translates those decisions into operating controls including environments, data policies, release processes, and monitoring, throughout each solution's lifecycle. This paper shows how the two disciplines work together. It follows one example solution through five stages (Build, Validate, Authorize, Operate, and Retire) and explains how a Center of Excellence (CoE) makes those disciplines repeatable at scale.

This paper focuses on the governance of AI-enabled solutions built by business and technology teams using Copilot Studio and Power Platform. These solutions are one part of a broader enterprise AI governance program that also encompasses Microsoft 365 Copilot, AI embedded in purchased software, and other enterprise AI uses. Although specific controls vary by technology and use case, the disciplines described here—and reflected in the industry guidance cited throughout—are consistent with broader principles for responsible AI governance.

Most controls in this paper, including environments, data policies, and release discipline, will be familiar from low-code governance. They remain necessary, but are no longer sufficient. AI-enabled solutions differ from the apps and flows those controls were designed for in five ways:

- **Answers can be confidently wrong.** Generative AI can confidently present erroneous or false content.[10] Quality cannot be inferred from successful execution. Require answer-quality checks proportionate to the consequences of an error.
- **Behavior and exposure can change without a code change.** Instructions, knowledge sources, tools, and access configuration are part of the solution's operating design.[6][7] Review changes to them even when the app looks unchanged. A wider audience can also change the consequences of an error and should trigger reassessment.

- **Agents act with delegated authority.** When an agent can update a record or start a process, it acts on someone's behalf. Governance must decide whose authority it uses, how far that authority extends, and where a human must approve.
- **AI can amplify oversharing.** Content available through an agent's configured knowledge sources can become easier to discover. Access depends on the source and authentication configuration; uploaded knowledge and sources accessed in a user's context do not necessarily expose content in the same way.[7] Governance must verify what the intended audience can retrieve.
- **Content can carry malicious instructions.** Attacker-controlled text in documents, emails, web pages, or tool results may be misinterpreted as instructions, creating a risk of unintended actions or data exposure.[11] Treat retrieved content as information, not authority, and enforce permissions and action limits independently of model instructions.

These risks do not require a separate governance program. They require the existing program to address accuracy, change, authority, content access, and manipulation at every stage of a solution's life. The rest of this paper shows where those questions belong.

Business decides, technology enforces

Governance fails when it belongs to no one or to IT alone. The business decides why a solution exists, what risk is acceptable, and where human control must remain. Technology translates those decisions into permissions, approval steps, deployment controls, and monitoring. Each depends on the other: business expectations without technical controls leave too much to chance, while technical controls without business direction can protect the wrong things or obstruct useful work.

Assign clear ownership. Shared or consequential solutions have two named owners. The business owner is accountable for why the solution exists, the value it delivers, and the risks the organization accepts. The technical owner is responsible for keeping it secure, reliable, and supported. Together, they establish release, suspension, and restart authority, with designated deputies when an owner is unavailable. Personal-use solutions retain named maker accountability and lightweight documentation; broader use or higher risk triggers the two-owner requirement. The following roles form a recommended operating model, adapted to the organization's existing responsibilities:

Role	Accountable for	Key decisions
Executive sponsor	The governance framework and solution funding	Approves the framework and tier model; funds the CoE; resolves cross-functional escalations
Business owner	Purpose, value, and accepted risk	Confirms the risk tier; accepts residual risk and approves releases; sets quality thresholds; can suspend for quality or value reasons; jointly authorizes restart; decides retirement

Role	Accountable for	Key decisions
Technical owner	Security, reliability, and support	Co-approves releases; can suspend immediately for technical or safety reasons; jointly authorizes restart after controls are verified; carries out retirement
Platform team	Zones, data policies, and the release path	Enforces controls; grants environment and connector access
Security and compliance	Independent assurance	Independently reviews High-tier solutions and proposed reductions from High; advises on regulatory obligations; leads or supports incident response according to incident type and the organization's response plan
Center of Excellence	Enablement and reusable practice	Advises and enables; owns patterns and guidance, not approvals

These responsibilities belong within a leadership-approved governance framework, not solely with IT. That partnership makes a difference: McKinsey links CEO oversight of AI governance with stronger self-reported financial impact.[1]

Make the safe path the easy path

Good governance helps people get things done without giving every solution the same freedom. Leaders should insist on three mechanisms that make the safe path the default: environment zones, data policies, and review proportionate to risk.

Environment zones. Separate personal productivity, team collaboration, and enterprise-critical solutions into distinct environments with escalating controls. Makers experiment freely in a governed sandbox; solutions that serve a department or touch sensitive data move to team or enterprise environments with stronger oversight. Treat the default environment, where licensed users have the Environment Maker role,[12] as a personal-productivity zone, not the default home for shared or consequential solutions.

Data policies. Decide once which applications may access each type of data and which audiences may access it, rather than approving data access by app. Data loss prevention policies classify connectors as business, non-business, or blocked, and Microsoft now extends similar controls to agents, governing what they can connect to and use.[3] Assign stricter policies to zones that handle sensitive information. Done well, makers rarely have to ask whether a connection is allowed; the platform answers for them. Data policies do not replace content permissions or source-specific access checks. For each knowledge source and action, verify the identity used for access and the information exposed to the intended audience. Do not assume that every source preserves end-user permissions automatically.[7] Before grounding an agent, review sharing, permissions, sensitivity labels, and the configured access model, then test what representative users can retrieve.

Proportionate review. Not every solution needs the same scrutiny. Gartner recommends controls proportionate to an agent's autonomy and access;^[2] uniform controls fail in both directions, overburdening simple solutions and under-protecting powerful ones. In rGen's proposed framework, the solution team completes a standard assessment of data sensitivity, the consequence of an error, and the solution's authority to act without human approval. Deterministic rules assign the highest factor as the proposed initial tier (Low, Moderate, or High), and the technical owner validates the assessment inputs and resulting tier. The business owner then confirms the tier because deciding how much risk the organization accepts is a business decision, not a technical one. High-tier solutions and proposed reductions from High also receive independent review by security or compliance.

	Low Risk	Moderate Risk	High Risk
Data sensitivity	Public or non-sensitive internal	Confidential business or customer data	Regulated or highly sensitive data
Consequence of error	Negligible; personal use; easily caught	Rework or service impact; team or department reach	Material legal, financial, safety, or reputational harm; difficult-to-reverse consequences
Human agency	Human prompts, reviews, and acts	AI proposes; human approves before action	AI acts independently; assess the scope, consequences, and reversibility of its actions
Default zone	Personal productivity	Team	Enterprise
Review and release	Standard data policy; maker training	Documented testing; owner sign-off	Independent review; formal risk assessment; named risk acceptor
Operation	Spot checks	Periodic quality and access review	Continuous monitoring; tested suspension and fallback
Example	Personal meeting-summary agent	Customer-service assistant	Agent changing records without approval, or answering customers on open-ended or consequential topics

The highest factor sets the initial tier by default before additional safeguards are applied. Assess residual risk after those safeguards have been implemented and tested. Any reduction from High to Moderate requires documented evidence, business-owner acceptance, and independent concurrence from security or compliance. Record the initial tier, safeguards, residual risk, approved governance tier, approvers, and review date in the governance profile. Safeguards supporting a reduction remain mandatory; if they fail or are removed, the reduction

must be reassessed. Internal tier decisions do not alter applicable legal obligations. Reassess the tier whenever data, audience, tools, permissions, or autonomy change.

All tiers retain baseline data policies, access controls, accountable ownership, and documentation proportionate to risk. Autonomy warrants closer review, but the scope, consequences, and reversibility of an action determine whether a lower governance tier can be justified.

The default zone follows the tier but is not fixed by it. A Moderate solution that serves an entire organization may still belong in the enterprise zone, because zone reflects audience and criticality as well as risk.

These are internal governance tiers, not legal classifications. Assess applicable privacy, sector, and AI obligations separately. The EU AI Act uses a risk-based framework and includes transparency obligations for certain AI interactions, including chatbots.[9] For customer-facing assistants, identify applicable disclosure requirements and make clear when customers are interacting with AI. A governance profile supports documentation and accountability; it does not by itself establish compliance.

Start with what you already have

Most organizations are not starting from zero. They already have agents, apps, and flows that no one has documented, some built by people who have since changed roles or left, plus AI tools adopted outside approved channels. Governance that applies only to new solutions leaves existing exposure unaddressed.

Use platform inventory and usage reporting to identify apps, flows, owners, and usage within the platform.[13] Supplement that view with the organization's discovery process for AI used outside approved platforms, then sort the findings using the risk tiers:

- **Valuable and Low risk:** register it, confirm maker accountability, verify baseline safeguards, and complete documentation as needed.
- **Valuable and Moderate or High risk:** assign business and technical owners, create a governance profile, and move it to the right zone.
- **Unused, unowned, or duplicated:** investigate its purpose, users, and dependencies. Assign ownership if it remains valuable; retire it only after confirming that it is no longer needed or that an approved replacement is ready.

Treat shadow AI and Power Platform apps discovered in platform inventory and usage reporting as a potential signal of unmet demand, not automatically as misconduct. The goal is to bring it onto the safe path, not to punish it. Solutions found this way often show where the organization most needs approved patterns, and they make good early work for the Center of Excellence.

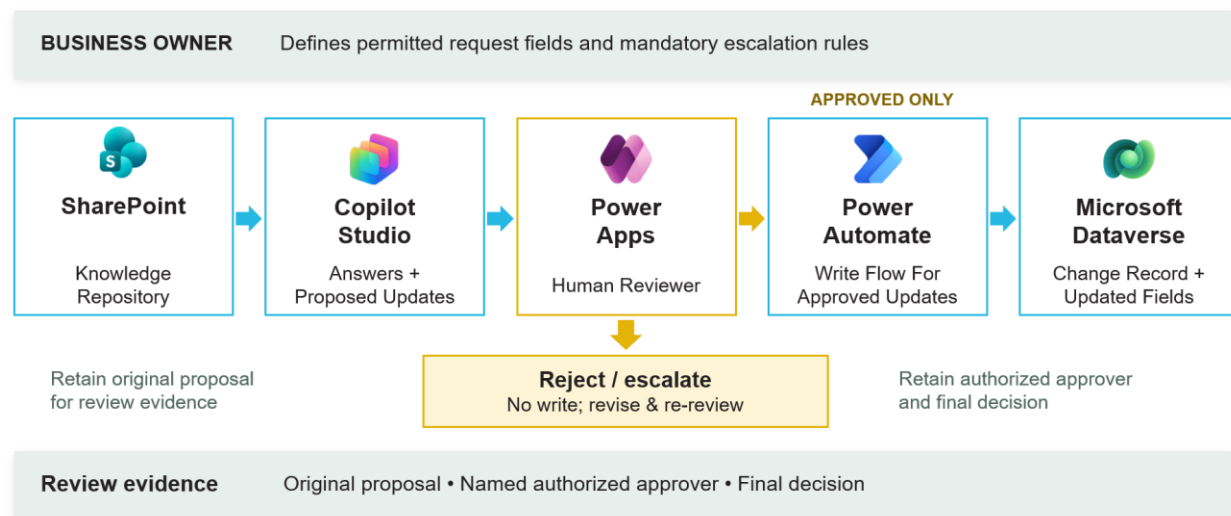
Over time, governance profiles for the solutions teams build become a core part of the organization's AI inventory, alongside records of Copilot and AI embedded in purchased software.

The governance profile

Consider a hypothetical customer-service assistant. It answers routine customer questions directly, within a defined set of topics and only from approved knowledge and hands anything outside that scope to a person. It also proposes updates to service requests. An authorized employee reviews each proposal, and **updates apply only after approval**. The business owner defines what employees may approve and what must be escalated. Each approved change is logged, including who approved it and what changed.

For this example, a **Moderate tier** is appropriate only if testing and review demonstrate that the assistant's direct answers remain within low-consequence topics, access to information is properly restricted, and out-of-scope or uncertain requests reach a person. Record changes require separate safeguards: permitted fields, authorized approval of the specific proposed change, and checks for incorrect updates. Customer data and customer-facing use do not automatically make a solution High; the sensitivity of the information and the consequences of its answers and actions determine the initial assessment. The governance profile records that assessment, the tested safeguards, residual risk, and the approved tier. A personal meeting-summary agent using non-sensitive information illustrates Low risk; broader authority or more consequential answers would require reassessment.

HYPOTHETICAL CUSTOMER-SERVICE REQUEST ASSISTANT • MICROSOFT TOOL EXAMPLE



[Figure 1. A customer question is answered within scope or handed to a person; a proposed service-request update changes the record only after human approval.]

Every AI-enabled solution needs a governance profile: a living record of what it does, which components and data it uses, who is accountable, which tier applies, and what the solution is authorized to do. The profile connects business decisions with technical controls. The business owner's purpose, tier, and accepted risk sit beside the technical owner's components, permissions, and release history. Keep the profile current from experimentation through retirement, with detail proportionate to risk, and organize it around six information groups.

Information group	What to document	Customer-service example
Purpose and accountability	Business problem, intended outcomes, and success measures; business and technical owners; risk tier; who approves releases and who can suspend or retire the solution.	Moderate tier, conditional on tested limits for direct answers and separately enforced controls for record changes. Records the initial assessment, safeguards, residual risk, approved tier, and any independent concurrence required for a reduction from High. Goal: faster, more accurate handling of service requests, measured by resolution time and update accuracy. The service manager owns the outcome; the platform lead owns technical operation.
Components and dependencies	Agents, apps, flows, data structures, and knowledge sources; what each does; connections and external or shared services it depends on.	SharePoint supplies approved knowledge; Copilot Studio answers questions and proposes changes; Power Apps supports human review; Power Automate applies approved updates; Dataverse stores service requests and change history.
Data access and permitted actions	Data sources and sensitivity; applicable data policy; identities used for access; authorized users; permitted reads and changes; human-approval requirements, escalation rules, and retention requirements.	Reads approved knowledge within the defined answer scope. Records the access identity for each knowledge source and action, including any user, application, or service account. Tests verify what customers and staff can retrieve under the configured access model. The update process restricts changes to permitted fields and requires authorized approval of the specific proposed change. Changes above the employee's limit are escalated. The zone's data policy blocks unapproved connectors.
Environments and deployment	Environment zone; where each component and data source resides in development, test, and production; environment-specific connections and settings; release version and deployment method.	Enterprise zone because the assistant serves the whole service organization, even if Moderate is the approved governance tier. Test uses a test request table and approved sample knowledge. Test connections and permissions are configured and verified to prevent writes to live service requests. Releases go through the approved deployment pipeline.
Testing and release approval	What changed in each version; functional, access, quality, and security test results; unresolved issues and accepted exceptions; release approver and date; recovery plan.	Evidence that unauthorized access is denied, rejected proposals leave records unchanged, out-of-scope questions are handed to a person, and answers to a standard question set meet the owner's quality bar. Records the named release approver and the rollback plan.

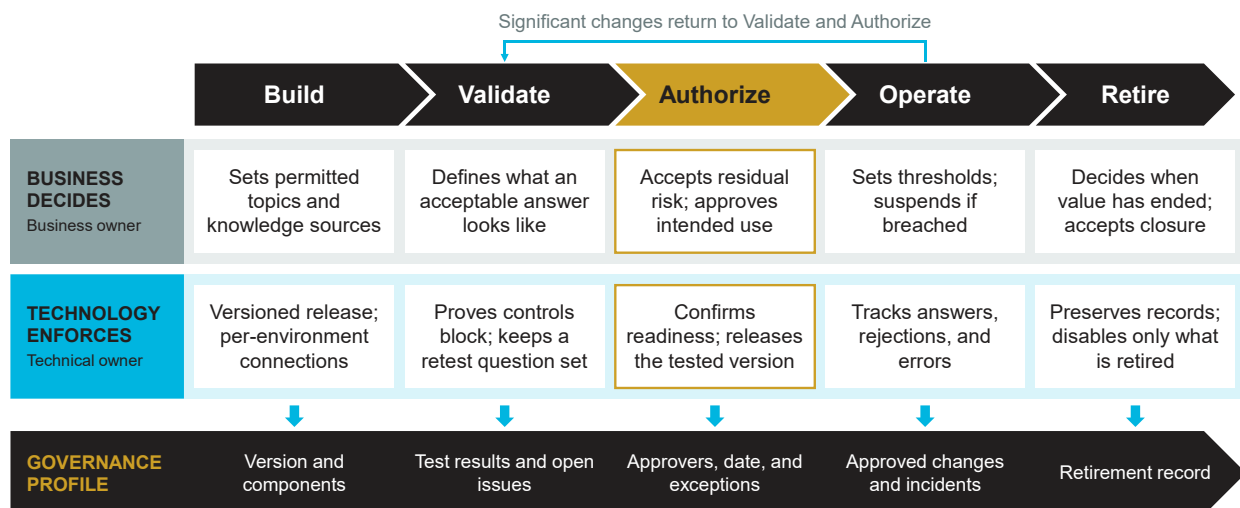
Information group	What to document	Customer-service example
Operation and retirement	Quality, usage, cost, and failure monitoring; thresholds and budget; support and incident responsibilities; review dates; retirement criteria, affected dependencies, user transition, and data retention or disposal.	Owners monitor answer quality, rejected proposals, failures, and cost against budget. Customers and staff have a clear way to report problems. A staged retirement plan protects shared knowledge and connections and keeps customer records for required retention periods.

All subsequent changes are added and stored within this profile. Release decisions, exceptions, incidents, and retirement are documented in one place rather than reconstructed later.

Governing the lifecycle: one example

Application lifecycle management (ALM) provides a controlled approach to developing and deploying a solution.[4] The principle is simple: build in development, prove the solution in test, and release it to production only through an approved, repeatable path. Do not edit production directly. Automated deployment pipelines can make the approved path easier to follow by guiding deployments and validating dependencies and target-environment settings.[14]

HYPOTHETICAL CUSTOMER-SERVICE ASSISTANT • GOVERNED RELEASE LIFECYCLE



[Figure 2. At every stage, the business decides and technology enforces; the governance profile carries each decision forward. Build → Validate → Authorize → Operate → Retire.]

Build: define exactly what will be released

Risk addressed: untested changes reaching users and test activity touching live records.

Give each release a name and version number, link it to the governance profile, and record which components are packaged and which dependencies must be prepared separately. Develop in unmanaged solutions and deploy managed solutions to downstream environments.[4] Managed properties control which managed components can be customized;

a managed solution is not automatically locked against all changes.[8] Confirm that your deployment approach supports the components you are using.[6]

The app, the data, and the connections are governed separately. Solutions and pipelines do not contain the data stored in Dataverse tables.[14] Any required data migration must be planned separately; deploying the application does not authorize copying customer records into development. Each environment must point to its own approved data sources and accounts, and Microsoft supports setting these values during deployment;[5] a successful deployment doesn't prove the connections are correct, so verify them.

Business decision: which topics the assistant may answer and which knowledge sources it may use. **Technical control:** a versioned release with environment-specific connections. **AI control:** knowledge sources are treated as governed components, not background content. Before the assistant is grounded in them, their permissions are reviewed and outdated articles are removed.

Validate: prove what it does and what it must not do

Risk addressed: a solution that works for an administrator but fails, leaks information, or can be manipulated in real use.

Test with representative users and sample data, not only an administrator's account. Verify that the update process enforces permitted fields and authorized approval of the specific proposed change, independently of the agent's instructions, and that a rejected proposal leaves the service request unchanged. Test incorrect answers, invalid updates, escalations, and failures. Include access tests and representative attempts to manipulate the assistant through instructions in retrieved content. Verify that out-of-scope or uncertain questions reach a person or receive a safe fallback when a person is unavailable. Testing provides evidence of control effectiveness, not a guarantee against every future error or attack.[10][11] Record the limitations and remaining risks.

Business decision: the business owner defines what an acceptable answer looks like. **Technical control:** confirm that the zone's data policies and controls actually block what they should. **AI control:** run a standard set of representative questions, including questions where an outdated article could produce a wrong answer, and keep it for retesting after every change. Link the results and unresolved issues to the version in the governance profile.

Authorize: make the release decision explicit

Risk addressed: risk accepted by default because no one was asked.

Before deployment, answer three questions:

- **Does it work as intended?** The release version matches the documented design and permissions, and testing supports its intended use.
- **Is the remaining risk acceptable?** Record who accepts any exceptions, the safeguards required, and when those exceptions are reviewed.

- **Is the team ready to operate it?** Production access, monitoring, support, and recovery arrangements are in place.

Business decision: the business owner accepts residual risk and approves the intended use.

Technical control: the technical owner confirms operational readiness. Release requires both approvals; High-tier solutions and proposed reductions from High also require independent review. **AI control:** confirm that the instructions, knowledge sources, and tool permissions being released are exactly those that were tested. As noted earlier, changing any of them can alter behavior without a code change. Record the version, changes, test results, approvers, and approval date. Keep that history in the governance profile for the life of the solution and beyond.

Operate: measure quality, and be ready to stop

Risk addressed: silent decline in quality, unnoticed cost growth, and slow response when something goes wrong.

Monitor the solution against the outcomes it was approved to deliver: quality, actions, failures, usage, and cost. AI quality requires its own signals. For the assistant, begin with three: a weekly review of sampled answers, the rate at which reviewers reject proposed updates, and errors reported by customers and staff. Adjust the measures and review frequency to the solution's risk and observed performance. **The business owner sets the thresholds;** the technical owner ensures they are measured.

Suppose an updated knowledge article introduces outdated policy terms. Sampled answers start to go wrong, and rejected proposals climb past the threshold. The business owner suspends the assistant, and customer questions return to staff while the article is corrected and the question set is rerun. When enforced by the update process, the approval requirement blocks changes that lack authorized approval. It does not guarantee that approved changes are correct or protect answers delivered directly to customers. Review affected answers and updates, correct any errors, and authorize restart only after the business and technical owners confirm that the issue has been resolved.

Changes to instructions, knowledge sources, tools, permissions, or audience can change behavior even when the app looks unchanged. Send significant changes back through Validate and Authorize, and reassess the tier whenever data, audience, tools, or autonomy change. Update the governance profile to reflect each approved change.

Retire: retire the workflow, not the history or shared services

Risk addressed: lost records, broken dependent solutions, and agents that keep acting after everyone believes they are gone.

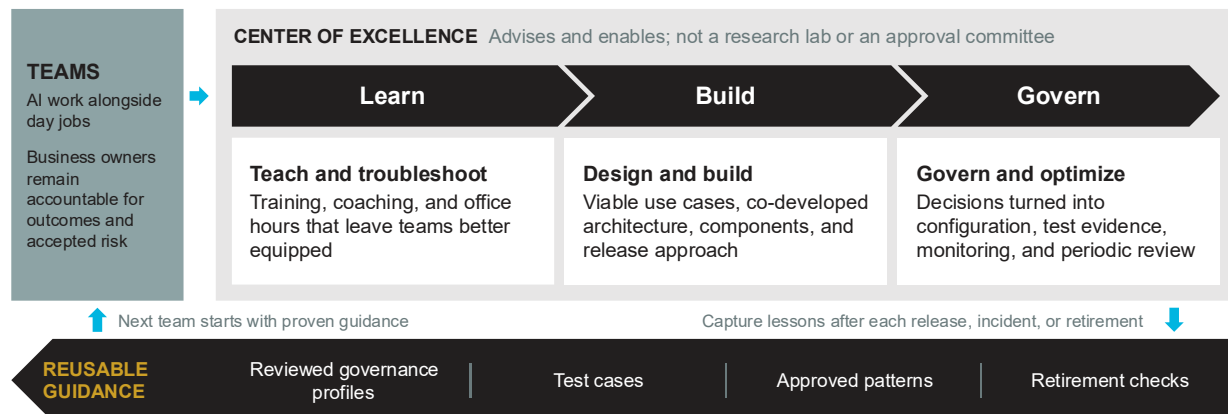
Retirement is a managed transition, not simply deleting an agent. When a new service workflow replaces the assistant, the **business owner** decides when its value has ended and accepts closure, and the **technical owner** performs the retirement using the governance profile as the map:

- **Transition the work.** Route new requests to the replacement or a manual process, and complete or transfer pending approvals and in-flight updates before withdrawing the old route.
- **Preserve what must be kept.** Customer records, approvals, conversation transcripts, audit records, and decision history stay accessible for required retention periods and legal holds. Stopping the assistant is not grounds to delete customer records.
- **Disable only what is retired.** Withdraw access to the old assistant and stop its update process, but leave shared knowledge and connections that other solutions still need. Do not assume withdrawal stops in-flight work: running and pending cloud-flow runs can continue after a flow is turned off.[15]
- **Verify and close.** Confirm that the old assistant can no longer act and that shared services still work. Then record retirement in the governance profile and inventory, and release only capacity or licenses that are proven to be unused.

Making it repeatable: the Center of Excellence

A Center of Excellence should provide practical and accessible expertise for teams whose AI work sits alongside their day jobs. It should not be a research lab or a central approval committee. At rGen, we see three complementary capabilities: teach and troubleshoot, design, and build, and govern and optimize.

MAKING IT REPEATABLE • HOW THE CENTER OF EXCELLENCE SUPPORTS TEAMS



Measure of success: the guidance and components teams reuse, not meetings held or approvals accumulated

[Figure 3. The CoE helps teams learn, build, and govern, then turns their experience into reusable guidance.]

Teach and troubleshoot. Offer hands-on training, coaching, and office hours covering platform tools and safe working practices, such as helping reviewers understand approval limits or helping a maker diagnose a connector issue. Support should solve the immediate problem and leave the team better equipped to handle the next one.

Design and build. Help teams select a viable use case and co-develop architecture, components, and a release approach. For the customer-service assistant, that means a

reusable review app, an approve-before-update pattern, and a standard question set, so each team does not rebuild the same safeguards.

Govern and optimize. Help owners translate access and autonomy decisions into configuration, test evidence, monitor, and conduct periodic reviews of permissions, licensing costs, and incidents. Bring business, security, platform, and operations specialists together when the tier warrants it.

Make the learning loop explicit. After a release, an incident, or a retirement, capture lessons in reviewed governance profiles, test cases, approved patterns, and retirement checks. Protect client and sensitive information before sharing. The next team should begin with proven guidance, rather than rediscovering the same problems.

Leadership should choose an internal, partner-supported, or blended CoE model based on available expertise and demand, then fund it as shared support rather than leaving each department to fend for itself. Judge its contribution by the guidance and components teams reuse, not by meetings held or approvals accumulated. The business owner remains accountable for outcomes and accepted risk; platform owners enforce controls. The CoE advises and enables.

What leaders should do now

Governance does not need to be complete before it is useful. Leaders can begin with six actions:

1. **Appoint accountable owners.** Name an executive sponsor and require business and technical owners for shared or consequential solutions. Keep named maker accountability for personal use, with clear triggers for stronger oversight.
2. **Approve the tier model.** Adopt Low, Moderate, and High tiers, along with the controls each requires, so teams know the rules before they build.
3. **Establish zones and data policies.** Set up personal, team, and enterprise zones, and decide once which data can flow where.
4. **Run discovery.** Find what already exists, register what is valuable, and retire what is not.
5. **Fix content permissions before grounding agents.** Review the sharing and labeling of the knowledge sources that agents will use.
6. **Fund the CoE.** Treat it as shared enablement, not a departmental cost or an approval queue.

Know whether it is working. Track a few measures that show whether governance is helping rather than slowing the organization down:

- **Inventory coverage:** the share of known solutions with an owner and a governance profile.
- **Time to approved release:** how long it takes to move from idea to production.

- **Reuse:** how many new solutions start from approved patterns.
- **Incidents:** how often problems are detected, and how quickly solutions are suspended and restored.

And in the end...

AI and Power Platform governance is a partnership. Business leaders decide what outcomes matter, what risks are acceptable, and where human accountability must remain. Technology teams translate those decisions into permissions, approval steps, deployment controls, monitoring, and retirement procedures. The governance profile keeps the two connected, while the Center of Excellence provides practical guardrails, support, and reusable guidance that help teams apply those decisions consistently without becoming a central approval committee.

Governance done this way aims to reduce avoidable friction while applying the controls each risk requires. It makes the safe path the easy path, so teams can build within clear boundaries instead of repeatedly seeking permission.

That is governance worth scaling: business-led decisions, implemented through technology, giving teams the freedom to build and the organization the confidence to scale.

Sources

- [1] McKinsey & Company. "The state of AI: How organizations are rewiring to capture value." March 12, 2025. <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-how-organizations-are-rewiring-to-capture-value>.
- [2] Gartner. "Gartner Says Applying Uniform Governance Across AI Agents Will Lead to Enterprise AI Agent Failure." May 26, 2026. <https://www.gartner.com/en/newsroom/press-releases/2026-05-26-gartner-says-applying-uniform-governance-across-ai-agents-will-lead-to-enterprise-ai-agent-failure>.
- [3] Microsoft Learn. "Configure data policies for agents." <https://learn.microsoft.com/en-us/microsoft-copilot-studio/admin-data-loss-prevention>.
- [4] Microsoft Learn. "Solution concepts." Updated January 29, 2025. <https://learn.microsoft.com/en-us/power-platform/alm/solution-concepts-alm>.
- [5] Microsoft Learn. "Pre-populate connection references and environment variables for automated deployments." Updated February 24, 2022. <https://learn.microsoft.com/en-us/power-platform/alm/conn-ref-env-variables-build-tools>.
- [6] Microsoft Learn. "Export and import agents using solutions." <https://learn.microsoft.com/en-us/microsoft-copilot-studio/authoring-solutions-import-export>.
- [7] Microsoft Power Platform Blog. "Knowledge in Microsoft Copilot Studio." March 27, 2025. <https://www.microsoft.com/en-us/power-platform/blog/2025/03/27/knowledge-in-microsoft-copilot-studio/>.
- [8] Microsoft Learn. "Managed properties." <https://learn.microsoft.com/en-us/power-platform/alm/managed-properties-alm>.
- [9] European Commission. "AI Act." <https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>.
- [10] National Institute of Standards and Technology. "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile." NIST AI 600-1. July 2024. <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf>.
- [11] Microsoft Security Response Center. "How Microsoft defends against indirect prompt injection attacks." July 29, 2025. <https://www.microsoft.com/en-us/msrc/blog/2025/07/how-microsoft-defends-against-indirect-prompt-injection-attacks>.
- [12] Microsoft Learn. "Power Platform environments overview." Updated August 31, 2026. <https://learn.microsoft.com/en-us/power-platform/admin/environments-overview>.
- [13] Microsoft Learn. "Manage and govern the default Power Platform environment." Updated June 23, 2026. <https://learn.microsoft.com/en-us/power-platform/guidance/adoption/manage-default-environment>.
- [14] Microsoft Learn. "Overview of pipelines in Power Platform." <https://learn.microsoft.com/en-us/power-platform/alm/pipelines>.
- [15] Microsoft Learn. "Turn a flow on or off, and delete a flow." Updated June 15, 2023. <https://learn.microsoft.com/en-us/power-automate/disable-flow>.