Microsoft

Administering and Governing the Power Platform for Enterprise

Contributors

Joe Unwin, Katie Liu, Marek Lutz

April 2026

Table of Contents

1. Introduction

Microsoft Power Platform is a productivity application development platform that delivers innovative business solutions across one integrated platform. Power Apps, Power Automate, Power Pages, Microsoft Copilot Studio, Dataverse and Connectors allow organizations to more quickly and easily build custom apps, workflows, websites, and agents driven by generative AI. Power BI allows any business to analyze and visualize real-time business performance.

Microsoft uses the platform to build their own first-party applications Dynamics 365 Sales, Service, Field Service and more.

These applications are built natively on the Power Platform. Enterprise customers can also build custom line of business applications using this same technology. Additionally, individual users and teams within your organization can build personal or team productivity applications with no or low code.

This whitepaper is targeted towards people responsible for planning, securing, deploying, and supporting environments and applications built on the Power Platform.

1.1. Purpose

This whitepaper is designed for those responsible for administering and governing Power Platform environments at scale. It provides practical guidance on gaining visibility into the platform, structuring environments intentionally, and embedding governance into the full lifecycle of apps, flows, and agents.

The document covers the core concepts, capabilities, and decisions required to operate Power Platform securely and efficiently, with an emphasis on managed platform features, proactive planning, and consistent day-to-day operations that support enterprise adoption.

1.2. Scope

Unless specifically noted, all features mentioned in this whitepaper are available as of April 2026.

The following topics are out of scope for this whitepaper:

- Power BI and other parts of the broader Power Platform
- Power Apps fundamentals for building applications
- ISV deployment scenarios, which are handled differently from enterprise deployment scenarios.
- Performance tuning of applications
- Full deployment and management of first-party Dynamics 365 applications
- While many of the concepts presented in this paper apply to Dynamics 365 Finance, Dynamics 365 Supply Chain Management, and Dynamics 365 Retail, they are not directly covered.
- Third party solutions which integrate with Power Apps.

Please visit https://docs.microsoft.com/en-us/power-platform/ to learn more about these topics.

1.3. What's new

Since this whitepaper was last published in 2024, the Power Platform has undergone major changes that include a variety of new offerings that organizations will want to take advantage of in administering their platform. This version of the whitepaper has major additions and modifications in the following key areas:

The approach to governance has shifted from loosely structured environment management toward a clear, enforced zoned model, built on Environment Groups, rules, and automated routing. The new approach emphasizes predictability and standardization rather than guidance alone.

Governance now begins the moment a maker first interacts with the platform, with routing automatically placing them into the correct environment based on identity and role, helping apply governance controls early in the agent creation lifecycle.

Zones themselves have evolved into operational entities rather than conceptual maturity levels, with each zone defined by specific connector access, data boundaries, sharing constraints, and lifecycle expectations enforced through environment-group rules rather than manual configuration.

This leads to the Managed Platform. The managed platform introduces a unified, standardized set of capabilities that strengthen governance, simplify operations, and provide visibility across the tenant. It shifts administration from manual oversight to an integrated platform experience where discovery, configuration, deployment, and monitoring all occurs in one place. The managed platform ensures that every environment, asset, and agent operates within consistent boundaries and that governance is embedded directly into the operational fabric rather than applied after the fact. It creates a predictable lifecycle for agents by providing clear ownership, visibility, and guardrails from the moment an agent is created through to its deployment and ongoing use.

Inventory provides a single, tenant-wide view of all apps, flows, agents, connectors, and environments, giving administrators complete visibility into what exists, where it sits, and who owns it. This consolidates what used to require multiple tools and manual discovery into a single governance experience.

Advanced connector governance builds on this by allowing organizations to centrally define which connectors are allowed or blocked and enforce these decisions consistently across zones. This replaces fragmented decision-making with predictable, rule-driven connector behavior aligned to security and compliance expectations.

Simplified deployment is supported through standardized pipelines and solution-based ALM, giving teams a repeatable way to package, validate, and promote agents between zones. Approvals, configuration consistency, and environment-specific policies are automatically woven into the deployment process, ensuring that promotion from personal to team to enterprise spaces happens safely and predictably.

Onboarding experiences such as welcome content and guided environment messaging now appear as part of the maker experience, helping creators understand rules, responsibilities, and boundaries before they begin building, which reinforces governance at the earliest possible stage.

Usage insights and analytics play a critical role in the managed platform by surfacing visibility into adoption, performance, consumption, and operational health. Administrators and business owners can quickly identify high-value agents, detect drift or anomalies, and track which parts of the organization are benefiting most. Integrated monitoring and reporting complement this by consolidating health signals, dependency issues, performance trends, and operational risks into a unified view that spans every environment and zone.

2. Overview

Power Platform is a product family that delivers innovative business solutions across one seamlessly integrated platform. Power BI, Power Apps, Power Automate, Power Pages and Microsoft Copilot Studio allow any business to analyze and visualize real-time business performance, quickly and easily build custom apps, automate workflows, deliver business websites, create agents and integrate AI capabilities.

Power Platform supplies a low code interface for any user to quickly create custom apps while simultaneously providing robust tools for pro developers. This makes it possible to develop integrated and innovative solutions across Azure, Microsoft 365, Dynamics 365, and standalone applications. For enterprises, the platform serves as a core "agility layer" that allows organizations to quickly build new applications and experiences while leveraging the data and services provided by core business systems like SAP or Salesforce. At the intersection of these products lies digital transformation – giving the customer the power to innovate anywhere, while enabling organizations to realize value across teams and departments.

Power Platform enables businesses to easily analyze data, create apps, automate workflows, build agents, and integrate AI.

2.1. Why do organizations use the Power Platform?

The Power Platform supplies a level of capability, security and supportability that allows organizations to quickly build, test and deploy applications and solutions, leveraging their existing data and systems. In this way, it provides a layer of capability that is agile, adaptable, and cost effective for developing and testing new ideas and services for an organization. As previously mentioned, the Power Platform functions as an "agility layer" that allows organizations to quickly build apps and deliver them out to their intended audience.

One of the key features of the Power Platform is its ease of use. The tools are low-code to no-code, meaning that makers do not need to write code to create working solutions. With appropriate governance in place, this means that solutions are produced quickly and in standard, supportable ways. They are likely to have less bugs with the testing tools provided and can be quickly deployed to end-users to test ideas and assumptions.

Another reason organizations use the Power Platform is that it is easy to integrate with their enterprise data. From building an order tracking app for front line staff that leverages data in SAP, or an employee feedback app that leverages your employees HR file sitting in Workday, the Power Platform is built to integrate with your enterprise data safely and seamlessly – wherever it is.

Addressing the shadow IT concern, many organizations are faced with the challenge of business units or individual users creating unofficial, often unsanctioned, IT solutions to address their specific needs. This phenomenon, known as shadow IT, can introduce risks related to data security, compliance, and system integration. The Power Platform offers a compelling solution to this challenge. By empowering users to develop their own solutions within a controlled and governed environment through the new onboarding and playgrounds, organizations can harness the innovation and agility of shadow IT while helping mitigate common risks related to security, compliance, and visibility. Users can solve their individual and business needs, while IT keeps oversight, ensuring that everything developed aligns with organizational standards and best practices. This balance between user empowerment and IT governance ensures that innovation can occur anywhere within the organization, enabling organizations to realize value across teams and departments.

Finally, organizations use the Power Platform because it is simple to use and scale reliably. Classic software development and operation typically involve many highly technical areas resulting in higher cost to operate and higher risk to maintain and update as business changes. By virtue of Power Platform being a standardized, robust, reliable offering, organizations no longer must worry about the undifferentiated heavy lifting of building, deploying, and operating coded applications. Power Platform makes it simple for an individual maker to create and deploy solutions to an audience.

2.2. Dataverse

Dataverse, is a fundamental component of Power Platform that provides a secure and scalable storage solution for data. Power Platform applications, such as Power Apps, Power Automate,

and Power BI, can seamlessly utilize Dataverse to store and manage data, making it a pivotal part of the unified ecosystem for business application development.

A significant feature of Dataverse is its ability to store data in a relational manner, allowing for complex relationships, hierarchies, and structures that are crucial for complex business applications. The data within Dataverse is stored in tables, which are analogous to tables in other relational databases such as SQL. Each table can contain multiple columns and can have relationships with other tables. Additionally, the platform allows for the definition of business rules and logic at the data layer, ensuring that data operations maintain business-specific constraints and conditions.

Another core feature is the rich set of security mechanisms Dataverse provides. Data integrity and protection are vital for business applications, and Dataverse facilitates this through its robust security model. This model supports fine-grained permissions, role-based access control, and field-level security, ensuring that only authorized users can access or modify specific sets of data. Moreover, it is integrated with Entra, which further bolsters its security capabilities by aligning with a widely accepted identity service.

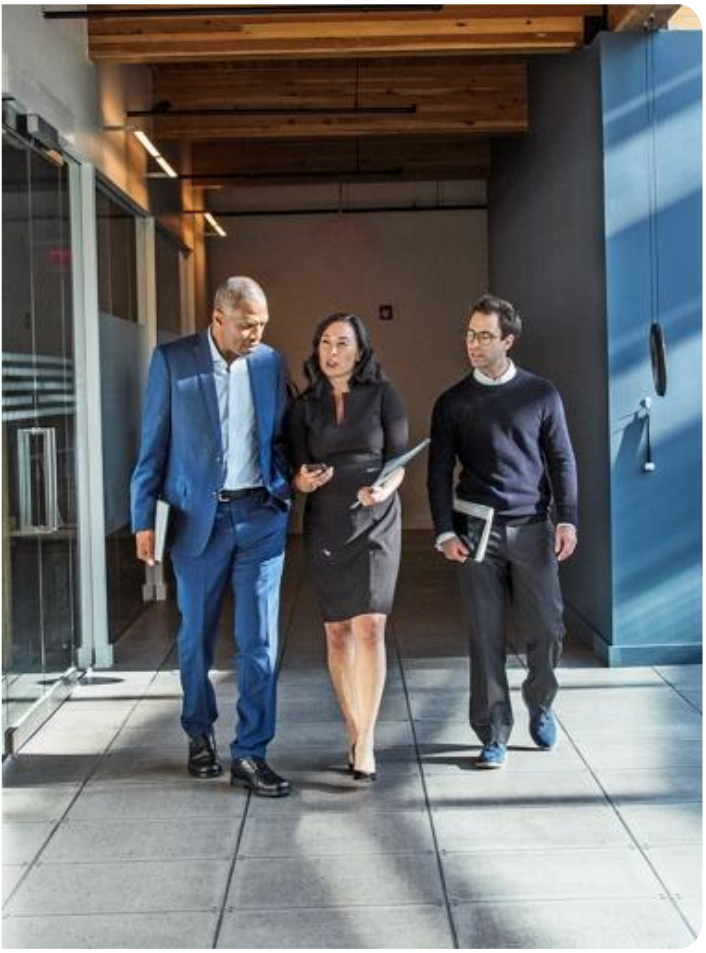
Lastly, Dataverse is known for its extensibility and integration capabilities. It provides a set of APIs that developers can use to interact with data, allowing for the creation of custom applications or integration with other systems. This feature is bolstered by a rich set of connectors provided by the Power Platform that facilitates integration with hundreds of applications and services, ranging from other Microsoft products like Dynamics 365, to SharePoint Online and Azure services, to third-party solutions.

In essence, Dataverse's core features of relational data storage, security, and extensibility make it an invaluable asset within the Power Platform's ecosystem.

> If you're looking at getting started with Dataverse you can review the training materials here: https://learn.microsoft.com/en-us/training/paths/get-started-cds/

2.3. Power Apps

Power Apps is a pivotal component of the Power Platform, providing a tool for rapid application development tailored for both business users and professional developers. With a focus on creating business applications quickly, Power Apps aims to bridge the gap between IT departments and other business units, reducing the time and complexity traditionally associated with app development.



One of the core features of Power Apps is its low-code approach to application development. Users can build fully functional apps with minimal coding, utilizing a user-friendly, drag-and-drop interface. This low-code environment empowers not just professional developers but also business users (often termed as "makers") to create bespoke applications tailored to their specific needs without waiting for extended development cycles.

Another defining characteristic of Power Apps is its low touch.

integration with other components of the Power Platform, especially Dataverse. This integration enables apps to store, retrieve, and interact with data effortlessly, ensuring consistency and reliability in the underlying data. Moreover, Power Apps can connect with a vast array of external data sources, from SharePoint Online and SQL Server to various third-party services, thanks to the myriad connectors available within the Power Platform ecosystem.

Furthermore, Power Apps is built with mobility in mind. In an era where business operations are increasingly shifting towards mobile platforms, Power Apps allows for the creation of responsive apps that work seamlessly across devices – be it a desktop, tablet, or smartphone. This ensures that business processes can continue uninterrupted regardless of the device or location, promoting flexibility and efficiency. In summary, Power Apps stands as a transformation tool in the Power Platform, offering rapid, low-code and vibe coding application development that integrates with diverse data sources while ensuring mobility and responsiveness.

If you would like to find out more about Power Apps then check out the learn documentation here: https://learn.microsoft.com/en-us/power-apps/

2.4. Power Automate

Power Automate is a service designed to automate workflows and tasks across a multitude of applications and services.

Its primary objective is to facilitate automation in business processes, enhancing efficiency and ensuring that repetitive tasks are handled automatically, reducing the manual overhead and potential for human error.

A core feature of Power Automate is its vast collection of pre-built connectors. These connectors allow users to establish automated workflows between different services, be it within Microsoft's ecosystem—like SharePoint, Dynamics 365, and Microsoft 365 or third-party applications such as Twitter, Dropbox, and Google Workspace. This extensive range of connectors ensures that Power Automate can be a central hub for automating workflows, irrespective of the disparate technologies a business might be using.

The versatility of Power Automate is further showcased through its ability to cater to both simple and complex automation scenarios. For straightforward tasks, users can leverage templates, which are predefined workflows tailored for common use cases. For instance, a user can quickly set up a flow that saves email attachments from Outlook to OneDrive.

However, for more intricate workflows, Power Automate provides advanced logic capabilities, like conditions, loops, and switches, enabling the crafting of nuanced automations tailored to specific business needs.

Security and compliance are also paramount in Power Automate. Given the potential sensitivity of data being processed and transferred across services, Power Automate emphasizes secure and compliant data handling. It integrates seamlessly with

Microsoft's security infrastructure, ensuring that data remains protected throughout its journey. Moreover, the platform provides detailed audit logs, allowing administrators to monitor and review all automation activities.

> If you would like to find out more about Power Automate then check out the learn documentation here: https://learn.microsoft.com/en-us/power-automate/

2.5. Power Pages

Power Pages provides organizations with a way to build externally facing business websites. Power Pages websites are highly secure and ready to scale with any enterprise. Creating sites through an easy-to-use low-code interfaces that will accelerate the design, build, and publishing workflow for both low-code makers and professional developers alike.

Power Pages, like other products in the Power Platform can take advantage of shared business data stored in Dataverse. This allows makers to build everything from apps, workflows, intelligent virtual agents, and analytics visualizations that are exposed through Power Pages.

Power Pages also provides enhanced security and governance at its core. This allows authors to ensure that business data is secure. Power Pages can take advantage of site authentication with a variety of providers which can, in turn, provides authorization scopes to business data. Power Pages supports modern Transport Layer Security (TLS) standards and is facilitated through the Azure App Service platform. The underlying services align with a variety of compliance accreditations, including International Organization for Standardization (ISO), System and Organization Controls (SOC), and Payment Card Industry Data Security Standard (PCI DSS).

For organizations servicing international, high-traffic sites, Power Pages can be set up to use content delivery networks, web application firewalls, and edge caching. By using Azure Front Door with Power Pages, organizations can offer global, low-latency business websites, taking advantage of the SaaS (software-as-a-service) platform.

Power Pages was formerly known as Power Apps Portals and Dynamics 365 Portals. The portals features previously available on these platforms, are now incorporated into Power Pages. This also means that tools such as the Power Pages Design Studio, Portals Management App, and the Power Platform Command-Line Interface (CLI) are compatible with Power Pages.

If you would like to find out more about Power Pages, then check out the learn documentation here: https://learn.microsoft.com/en-us/power-pages/

2.6. Microsoft Copilot Studio

Microsoft Copilot Studio is the enterprise AI agent platform from Microsoft, designed to help organizations build, connect, and govern intelligent agents across their business.

It provides a centralized environment for architecting agents that can act and operate over enterprise data, applications, and workflows, while remaining secure, observable, and compliant at scale.

As organizations enter the enterprise agent era, Microsoft Copilot Studio enables a shift from simple conversational experiences to agents designed to assist with and automate defined business tasks. These agents can support the automation and orchestration of multi-step processes through streamlined workflows that improve velocity, reduce costs, and increase productivity. Microsoft Copilot Studio is built to support this transformation by providing the tooling enterprises need to move from experimentation to production with confidence.

Copilot Studio offers an intuitive, natural-language–driven development experience that supports a wide spectrum of agent scenarios, from simple task assistance to sophisticated, multi-agent orchestration. Organizations can select from a broad range of models, ground agents in structured and unstructured enterprise knowledge regardless of where it resides, and extend agent capabilities using built-in tools, connectors, and automation flows.

A defining strength of Copilot Studio is its governance and security capabilities designed for enterprise environments foundation. The platform provides visibility and administrative controls across key stages of the agent lifecycle, including access management, data protection, capabilities that support compliance requirements and policy enforcement, and risk mitigation enabled by the Power Platform admin center (PPAC). Centralized controls help organizations manage and reduce the risk of uncontrolled agent sprawl, ensure sensitive data is handled appropriately, and confidently scale agent deployments across teams and departments. Measurement and reporting capabilities enable organizations to track usage, adoption, and impact, supporting continuous optimization and clear ROI justification.

By combining powerful agent orchestration, broad ecosystem connectivity, and robust governance, Microsoft Copilot Studio enables organizations to unlock the full potential of AI agents, transforming how work gets done while staying firmly in control.

If you would like to find out more about Microsoft Copilot Studio, then check out the learn documentation here: https://learn.microsoft.com/en-us/microsoft-copilot-studio/

2.7. Connectors

Connectors function as bridges between different services, enabling data to flow seamlessly across a diverse range of applications, databases, and other services. Whether you are looking to integrate data from cloud-based solutions, on-premises systems, or third-party platforms, connectors ensure that the Power Platform remains flexible and extensive in its data integration capabilities.

A key feature of connectors is their pre-built nature. Microsoft offers hundreds of ready-made connectors for the Power Platform, catering to popular services such as SharePoint, Dynamics 365, Azure SQL, Salesforce, Google Workspace, and many more. These connectors drastically simplify the process of data integration, allowing users to establish connections between the Power Platform and external systems with just a few clicks, without needing to dive deep into API intricacies.

In addition to the pre-built connectors, Power Platform also provides the capability to create custom connectors. Recognizing that businesses may have unique or niche systems that are not covered by the pre-built connectors. The platform provides tools for users to design their own connectors.

This ensures that even proprietary or less common systems can integrate with the Power Platform, offering businesses unparalleled flexibility.

The robustness of connectors is also worth noting. They are not just simple data pipes but come with capabilities to handle authentication, error handling, and even data transformation in some cases. This ensures that data integration is not just possible, but efficient and reliable, reducing potential friction points when bridging multiple systems.

Connectors are the unsung heroes of the Power Platform, ensuring that data and actions can move effortlessly between a vast array of services. They encapsulate the platform's philosophy of accessibility, flexibility, and integration, providing users with tools to make the most of their data, irrespective of where it exists.

If you would like to find out more about Connectors, then check out the learn documentation here: https://learn.microsoft.com/en-us/connectors/

2.8. Managed environments

Managed environments provide native capabilities for inventory, usage insight, controlled sharing, advanced connector governance, and standardized deployment and lifecycle management. By making governance and operations part of the platform experience, not an external process, managed environments enable organizations to scale confidently while keeping risk and complexity in check.

Enabling a Managed Environment is a simple admin action (for example, in the admin experience you can navigate to environments and use the Managed Environment toggle), and it unlocks controls that reduce deployment risk such as consistent policy application across environments and stronger governance posture.

> You can read more about Managed Environments throughout this whitepaper and on the learn documents website at: https://learn.microsoft.com/en-us/power-platform/admin/managed-environment-overview

2.9. AI features of the Power Platform

The Power Platform can also take advantage of recent advancements in Generative AI and machine learning. With Copilot, organizations can describe what they want their app, flow, or agent to do, in natural language, and Copilot will build it for them.

This capability allows makers to be incredibly effective in generating even complex apps quickly and safely.

In addition to Copilot, AI Builder empowers users to integrate artificial intelligence capabilities into their apps and workflows without needing deep AI expertise. Through a user-friendly interface, AI Builder offers pre-built models for common scenarios, like form processing, object detection, and prediction, while also allowing customization based on specific data. This tool democratizes AI, bridging the gap between complex AI processes and everyday business applications, enabling organizations to harness the power of AI-driven insights and automation seamlessly.

> You can read more about features across the platform here: https://learn.microsoft.com/en-us/power-platform/copilot

3. Planning

3.1. Understanding environments

In Power Platform, scaling safely isn't just about where solutions live, it's about ensuring every app, flow, and agent is created inside the right boundaries from day one. Just like traditional software development and deployment, the organization, management, and segregation of resources are pivotal to ensuring efficient workflows, security, and scalability. Power Platform addresses this need through the concept of "environments". These environments act as isolated containers, each hosting its own set of apps, data, and other components, ensuring that the different stages of the application lifecycle, from development to production, can be cleanly separated and managed.

Environments are the security boundary and the unit of management in the Power Platform. By delineating boundaries between different solution instances, environments prevent inadvertent overlaps, reduce the risk of errors, facilitate security management, and improve deployment efficiency. For instance, developers can work in an environment tailored for experimentation, such as their own personal playground environment, without the fear of disrupting live applications. At the same time, administrators can control access, allocate resources, and enforce policies differently for each environment, catering to its specific purpose and the needs of its users. With this foundational understanding, let us delve deeper into the primary types of environments within the Power Platform.

Each environment is tied to a single geographic location that is configured at the time the environment is created. By leveraging multiple environments in different regions, Power Platform supports multiple geographic deployments out of the box. Within each environment, the customer data for an environment does not leave the geography in which that environment is provisioned.

Environments can be used for individual users or user groups, for intended purposes, for stages in the lifecycle of a solution or app, or for different audiences. Each environment can have different Data Loss Prevention (DLP) policies applied to it, defining which connectors can or cannot be used in combination within the environment. Each environment has at the highest level a set of users that have the environment admin role. These users serve as the administrators of the environment.

Environments are no longer managed one-by-one as independent containers. Instead, organizations increasingly use Managed Environments and centralized governance to apply consistent standards at scale. Managed Environments unlock a suite of capabilities for operating Power Platform across the enterprise with more control, less effort, and better visibility, including environment routing, environment groups, and standardized deployment controls.

To reduce sprawl and enforce governance from the start, environment routing can automatically direct makers away from the shared default environment and into personal developer environments based on organizational rules. When used with environment groups, newly created environments can be automatically added to the appropriate group so that governance rules apply immediately, without manual setup.

Environment groups let tenant administrators cluster Managed Environments into logical zones and publish rules that lock key settings across every environment in the group. This prevents configuration drift and ensures that policies, such as ALM defaults and other governance settings stay consistent over time, even as new environments are created or moved between zones.

The default environment still exists and is required for certain Microsoft 365-integrated experiences, but it should be treated as a tightly controlled productivity space rather than a landing zone for scalable solutions.

3.2. Developing an environment strategy

A modern environment strategy focuses on minimizing long-lived solution development in the default environment, and instead using routing, groups, and managed controls to place makers into the right governed zone by design.

As an early step in establishing a well-governed Power Platform, organizations should formally define an environment strategy that reflects both today's footprint and the intended future state. At a high level, the strategy defines what environments you deploy, which workloads belong where, and how each environment is governed based on its role, business risk, and ALM path, so solutions can move from experimentation to production predictably and safely.

3.2.1. Manage the default environment

The first step in an enterprise environment strategy is to reposition the default environment. It must exist and it will continue to be used by Microsoft 365, integrated experiences, but it should not be the primary place where makers build long-lived apps, flows, or agents. Instead, treat it as a constrained personal productivity space, especially now individuals have their own personal development environments with environment routing, and use platform capabilities to move creation into governed developer workspaces by design.

For example, Power Apps created from SharePoint lists will be automatically provisioned in the default environment. In some instances, you can change the default environment routing for Power Platform solution created via a Microsoft 365 interface. For example, it is possible to set a custom default environment for SharePoint Online Power App forms. By using the Managed Environments environment routing, premium feature, you can route new makers to their own personal developer environment. Thereby simplifying solution practices and enforcing more control over user routing. Environment routing helps to limit sprawl and helps share best practices through onboarding content and experiences.

To limit the risk of data loss from the default environment, a policy should be invoked that limits the connectors available in the default environment to business approved connectors such as SharePoint, Teams, and Outlook. New connectors should be blocked by default in this environment as well as custom connectors. A default environment administrator should be assigned to regularly maintain and monitor this environment and ensure it is being used sensibly.

3.2.2. Determine major environment use cases

As a first step in building an environment strategy, define the major use cases (zones) your organization needs and the default path makers should follow. The goal isn't to only catalog environment "types", it's to standardize a small set of governed spaces and ensure makers land in the right one automatically. The recommendation is to now use environment routing so new makers are directed into their own personal developer environment (instead of building in the

default environment) and pairing that with environment groups so governance rules are applied consistently across every environment created for that use case.

Most organizations can cover many of their needs with four core use cases:

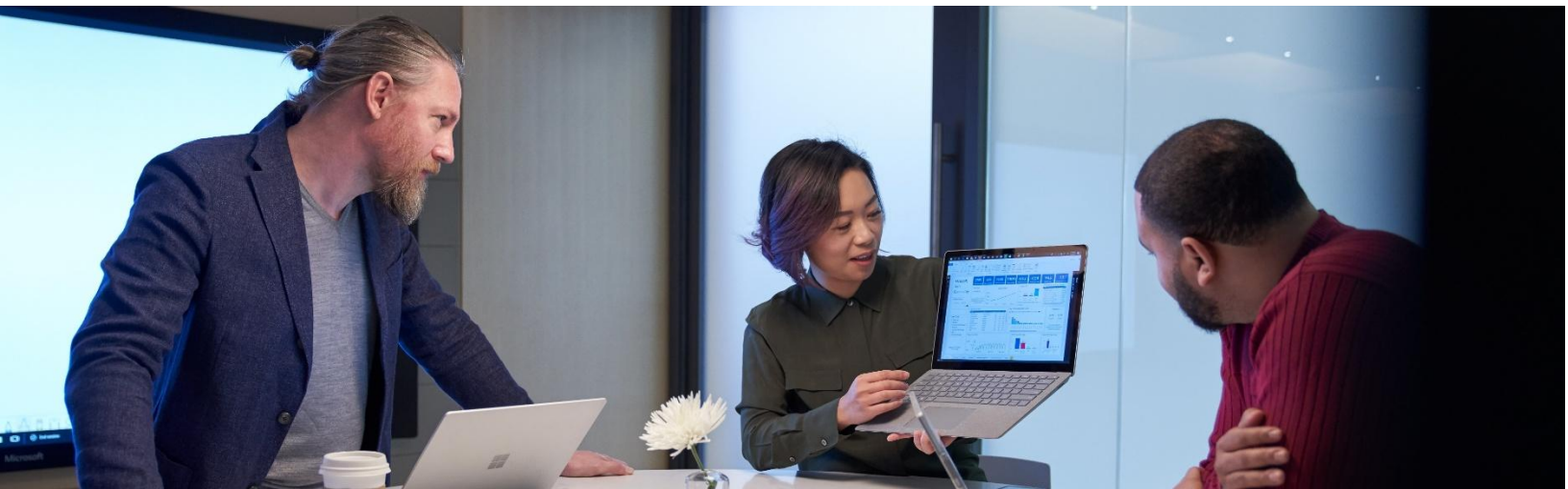
3.2.2.1 Personal developer (individual build / iteration): A private, auto-provisioned workspace for a single maker to iterate on small productivity solutions without impacting others. This should be the "default landing zone" for makers via environment routing, and it should be governed through an environment group so policies (for example, sharing limits, onboarding content, and connector constraints) are enforced without one-by-one configuration.

3.2.2.2 Advanced maker / departmental collaboration (power user): A governed team environment for advanced makers who need broader connector access, shared collaboration, or department-level solutions. Define how users are evaluated, trained, and granted entry, then enforce the right controls consistently via environment-group rules (rather than relying on manual setup).

3.2.2.3 Shared (multi-solution, medium complexity): A shared environment that intentionally hosts multiple solutions with common guardrails. This is often appropriate when solutions share components, data, or administrative overhead, and when teams need a faster path than full app-by-app dedicated isolation. The key is to treat "shared" as an explicit operating model: clear ownership, consistent policy enforcement, and a defined threshold for when a solution must graduate out of shared into a dedicated lifecycle.

3.2.2.4 Dedicated (business-critical workload with full lifecycle): A dedicated environment set (typically Dev/Test/Prod, and sometimes additional stages if needed) for a core solution used broadly or exposed to external users. This is where you standardize ALM expectations and governance, aligning environments to release stages, applying stricter controls in production, and using managed capabilities to keep lifecycle consistent at scale.

Most organizations will have these core use cases and should design an explicit process for how environments are created, grouped, governed, and administered.



3.2.3. Trial environments

Trial environments are designed for short-term evaluation and proof-of-concept work. Power Platform supports two trial models, standard and subscription-based, and they're governed differently depending on whether trials are user self-serve or admin-controlled in your tenant.

If your tenant allows standard trials for users, anyone with a suitable license can create a 30-day trial environment (typically one at a time, per the per-user trial entitlement). When the trial period ends, the environment is disabled and deleted, and its resources, including data, are removed unless it's converted to production before expiration (extensions can be available depending on the trial type and tenant policy).

For subscription-based trials, provisioning is admin-led: tenant admins can add a trial (subscription-based) environment to the tenant for broader, multiuser proof-of-concept scenarios.

To prevent sprawl, it's recommended to restrict trial environment creation using tenant environment creation controls (for example, limiting to Microsoft 365 Global Admins, Dynamics 365 Admins, and Power Platform Admins) and manage these settings from the Power Platform admin center.

For more details on who can create environments and more about trial environments review the docs here:
https://learn.microsoft.com/en-us/power-platform/admin/create-environment
https://learn.microsoft.com/en-us/power-platform/admin/trial-environments
https://learn.microsoft.com/en-us/power-platform/admin/control-environment-creation

3.2.4. Production environments

By default, production environments can be created by an administrator or by users with an appropriate Power Apps license, provided your tenant allows environment creation and at least 1 GB of Dataverse database capacity is available.

It is recommended to restrict production environment creation to only Microsoft 365 global admins, Dynamics 365 admins, and Power Platform admins from the Power Platform admin center.

3.2.5. Determine how environments are managed

Once environments are created, they require ongoing operational management. Adding and removing users, adjusting access, enabling or restricting connectors, and updating environment configuration are common day-to-day tasks that directly impact security and reliability. Define a clear operating process for how these requests are submitted, evaluated, approved, and executed so environment administrators can respond quickly without introducing inconsistent governance. At enterprise scale, the objective is to reduce case-by-case "exceptions" by standardizing what changes are allowed, who can approve them, and how approved changes are applied consistently across environments.

Use Environment groups to organize environments by operating zones and apply rules that enforce standardized settings across every environment in the group. When rules are published, the corresponding settings become locked at the environment level, keeping settings consistent and reducing administrative overhead. Using this with Environment routing, which can direct makers into personal managed developer environments by default, ensuring governance is applied before resources are created.

3.3. Planning your data with Dataverse

Dataverse planning is less about "where data lives" and more about ensuring your data model, naming standards, and storage posture scale as apps, flows, and agents multiply across environments. Each environment that uses Dataverse has its own Dataverse database, and solutions are deployed into that environment boundary.

3.3.1. Treat shared environments as shared schemas

Shared environments (for example, team or departmental environments where multiple solutions coexist) accelerate collaboration, but they also introduce a predictable risk, schema collision. When multiple solutions are deployed into the same Dataverse database, table and

column names must remain unique across all installed solutions. Without discipline, makers can unintentionally create duplicate or conflicting tables, fields, or relationships that are difficult to unwind later.

A scalable approach is to standardize publisher-based naming for all customizations:

- Use a distinct solution publisher prefix for each product team, department, or solution family so custom tables and columns remain clearly attributable and collision resistant.
- Require descriptive display names and consistent logical naming conventions so your table inventory remains supportable as adoption grows.

This becomes even more important as Copilot-assisted experiences reduce the friction of creating tables. If makers can generate tables quickly from prompts, governance must ensure the resulting schema still aligns to organizational naming and ownership standards.

3.3.2. Capacity is tenant wide

Dataverse storage is governed through a tenant-level entitlement model that tracks capacity across three storage categories:

- Database (relational table data and metadata)
- File (attachments and files stored with records)
- Log (logging/audit-related storage)

because these categories are tracked independently, capacity planning is should not take the "How big is my Dataverse?" approach but focus on which storage type will my solution primarily grow in, and do I have the right headroom for it?

3.3.3. Notification thresholds

If you're familiar with how notifications thresholds previously worked, the earlier 85% and 95% alert model should be updated. Notifications are now triggered based on how much capacity remains in any of the three storage types:

- When any storage category has less than 15% available, admins receive a "nearing limit" notification.
- When any storage category has less than 5% available, admins receive a stronger warning that administrative operations may be impacted.

These notifications are delivered on a recurring basis to tenant admins and Power Platform admins, and are tied to the capacity posture of the tenant

4. Governance

Effective governance of the Power Platform requires more than isolated policies or after-the-fact controls. As organizations scale their use of apps, flows, and agents across teams and business units, governance must be structured, enforceable, and embedded into the platform's operational fabric.

Microsoft recommends organizing governance around five core pillars that work together to provide security, consistency, visibility, and scalability across the full lifecycle of Power Platform solutions. These pillars align to how the platform is now administered—through Managed Environments, environment groups, automated routing, and centralized reporting—so that governance is applied by design rather than through manual oversight.

4.1. Fundamentals

To ensure success, it is recommended that fundamental governance principles are established. Conceptually, these principles can be summarized into three pillars that support the governance system.

4.1.1. Policy

There should be a comprehensive strategy outlining the system's objectives, accompanied by policies to provide high-level structure. For example, the overarching strategy may involve pilot agents to evaluate business benefits. To achieve this, policies may be required for data loss prevention, citizen development, and Application Lifecycle Management (ALM).

4.1.2. Process

The methodology for implementing this strategy is defined by processes. For instance, there should be a process for regular review and reporting of costs, as well as a process for ALM. Clearly defined issue escalation paths should also be established.

4.1.3. People

Governance is inherently complex, and requires management by individuals who are authorized, capable, and accountable.

No matter how many governance tools are available, true governance only happens when organizations define a clear strategy and actively implement supporting processes. Tools can help automate and enforce, but without strategic intent and disciplined execution, governance remains aspirational rather than operational. This is also why change management is important, and will be covered later in this whitepaper.

While no set of tools can replace the need for a clear governance strategy and disciplined processes, the right tools can make it much easier to operationalize and sustain those practices at scale.

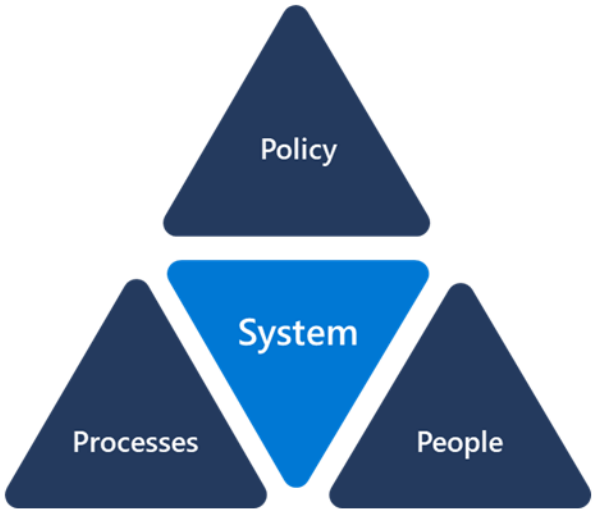


Figure 1 Governance pyramid

At a conceptual level, the pillars of governance can be presented as a pyramid. This pyramid emphasizes the importance of non-system factors in achieving a robust governance system.

Within the Microsoft governance ecosystem these governance pillars are enforced with the previously mentioned Environment Groups, Group Rules and Environment Routing.

4.2. Security controls

Security controls define the boundaries within which apps, flows, and agents are allowed to operate. These controls protect organizational data, enforce compliance requirements, and reduce the risk introduced by uncontrolled integration or exposure.

In the Power Platform, security controls span identity and access management, data protection, connector governance, and runtime enforcement. At a foundational level, environments act as the primary security boundary, allowing tenant administrators to isolate workloads, control data residency, and apply tailored policies based on risk and business purpose.

Key security capabilities include:

- Data Loss Prevention (DLP) policies that govern which connectors can be used together and restrict data movement across trust boundaries.
- Connector access and advanced connector policies that determine which services, including premium or custom connectors, are available in each environment.
- Role-based access and Dataverse security that control who can build, run, and share apps, flows, and agents, as well as what data they can access.
- AI-specific protections, including model selection, grounding boundaries, and integration with Microsoft Purview for sensitive data detection and compliance reporting.

Security controls are most effective when applied consistently at scale. Environment Groups and rules allow organizations to define these security baselines once and enforce them automatically across every environment in a given governance zone, preventing drift and reducing reliance on manual configuration.

4.3. Management controls

Management controls govern how Power Platform resources are created, operated, promoted, and retired over time. While security controls define what is permitted, management controls define ownership, accountability, and operational discipline.

These controls span the full lifecycle of apps, flows, and agents, including:

- Who is allowed to create environments and solutions

- How environments are configured and administered
- How sharing, publishing, and promotion between environments is handled
- How changes are reviewed, approved, and deployed

Modern Power Platform governance shifts management away from ad-hoc processes toward standardized, rule-driven operations. Managed Environments provide native capabilities for enforcing sharing constraints, onboarding experiences, solution deployment expectations, and environment-level settings that cannot be altered independently.

By combining environment groups, rules, and standardized pipelines, administrators can ensure:

- Consistent configuration across environments serving the same purpose
- Predictable promotion paths from development to production
- Clear separation between personal, team, and enterprise-critical workloads
- Reduced operational overhead by eliminating one-off exceptions

Management controls ensure that growth in apps, flows, and agents does not result in fragmentation, unclear ownership, or unsupported solutions.

4.4. Platform reporting and visibility

At scale, governance is unsustainable without comprehensive visibility. Platform reporting provides the insight administrators need to understand what exists, how it is used, and where risk or opportunity is emerging across the tenant.

Power Platform reporting is no longer limited to isolated usage metrics. The new reporting architecture delivers unified visibility across apps, flows, connectors, environments, and agents, enabling governance decisions to be based on real operational data rather than assumptions.

This visibility is delivered primarily through four experiences in the Power Platform admin center (PPAC): Inventory, Monitoring, Security, and Copilot, with Microsoft Purview providing cross-tenant data security and compliance reporting.

4.4.1. Inventory – Tenant-wide discovery

Inventory provides a comprehensive, cross-environment view of all Power Platform assets, including apps, flows, agents, and more. It establishes the authoritative catalog for governance by showing what exists, who owns it, where it lives, and how it is being used.

This inventory serves as the starting point for identifying sprawl, unmanaged solutions, and candidates for optimization or retirement.

4.4.2. Monitoring – Health and operational insight

The Monitoring experience surfaces health, reliability, and performance signals across Power Platform workloads. It enables administrators to identify failures, degradation trends, and abnormal behavior in apps, flows, and agents before they become business-impacting incidents.

Operational monitoring supports proactive governance by shifting administrators away from reactive troubleshooting toward continuous improvement.

4.4.3. Security – Centralized posture management

The Security experience consolidates security insights and recommendations across the Power Platform, including data access patterns, connector usage, and configuration risks.

It highlights misconfigurations and risky combinations that may increase exposure and provides guidance to remediate issues consistently across environments. This view connects platform-level governance with broader organizational security and compliance objectives.

By consolidating security signals and recommendations in one place, the Security experience reduces diagnostic overhead for IT teams, making it faster to identify misconfigurations, prioritize risk, and take corrective action across the platform.

4.4.4. Copilot – Adoption and value oversight

The Copilot experience consolidates governance, analytics, and business value metrics for Copilot and agent usage across Power Platform.

It provides usage, adoption, and value metrics that help organizations understand where AI-driven solutions are delivering impact, where governance controls are being applied, and how usage aligns with business outcomes. For many organizations, this becomes the executive-facing view that ties technical governance to return on investment.

However, when Copilot Studio agents are published to Microsoft Teams or Microsoft 365 Copilot, administration and visibility are surfaced through the Microsoft 365 Admin Center, making MAC the control point for these deployment surfaces.

4.5. Maker routing and environment auto-provisioning

One of the most important governance shifts in the Power Platform is that governance now begins before a maker creates their first app, flow, or agent.

Environment routing and environment groups ensure that makers are automatically placed into the correct governed space based on identity, role, or group membership. When a user first accesses Power Apps, Power Automate, or Copilot Studio, routing evaluates organizational rules and provisions them into an appropriate environment aligned to governance intent.

Typical routing patterns include:

- Personal developer environments for individual experimentation and iteration
- Team or departmental environments for shared collaboration
- Enterprise managed environments for high-risk or organization-wide solutions

Routing prevents accidental creation in the Default Environment and ensures that appropriate connector policies, sharing defaults, security baselines, and support expectations are enforced from the moment development begins.

By combining routing with environment group rules, organizations ensure that governance is applied automatically and consistently, reducing shadow IT and simplifying day-to-day administration.

4.6. Zones

Zones represent distinct levels or stages of governance maturity within the Microsoft ecosystem. Each zone is designed to address specific security, management, and operational needs while ensuring a scalable and adaptable framework for apps, workflows and agents. By conceptualizing governance through zones, organizations gain a clearer pathway to evolve their strategies, from foundational controls to advanced, fully integrated solutions.

The zones serve as a structured roadmap for IT practitioners and decision-makers to implement and manage agents effectively, aligning security and operational measures with their organization's maturity level. Each zone emphasizes key elements such as access management, capabilities that support compliance requirements and policy enforcement, resource optimization, and system integration. Organizations can progress through these zones by enhancing governance practices, expanding security measures, and refining management strategies. This tiered approach ensures adaptability to meet growing organizational demands while maintaining a robust governance structure.

At a practical level, Organizations implement zones through Environment Groups in the Power Platform Admin Center (PPAC). Environment rules prevent drift and apply consistently across grouped environments.

4.6.1. Zone 1 (green) – Personal productivity

Zone 1 represents the entry point for Power Platform governance and is designed to enable safe, individual innovation while maintaining clear security and operational guardrails. This zone supports personal productivity scenarios, where individual makers create lightweight apps, flows, and agents to improve their own efficiency or explore early ideas without impacting others or enterprise systems.

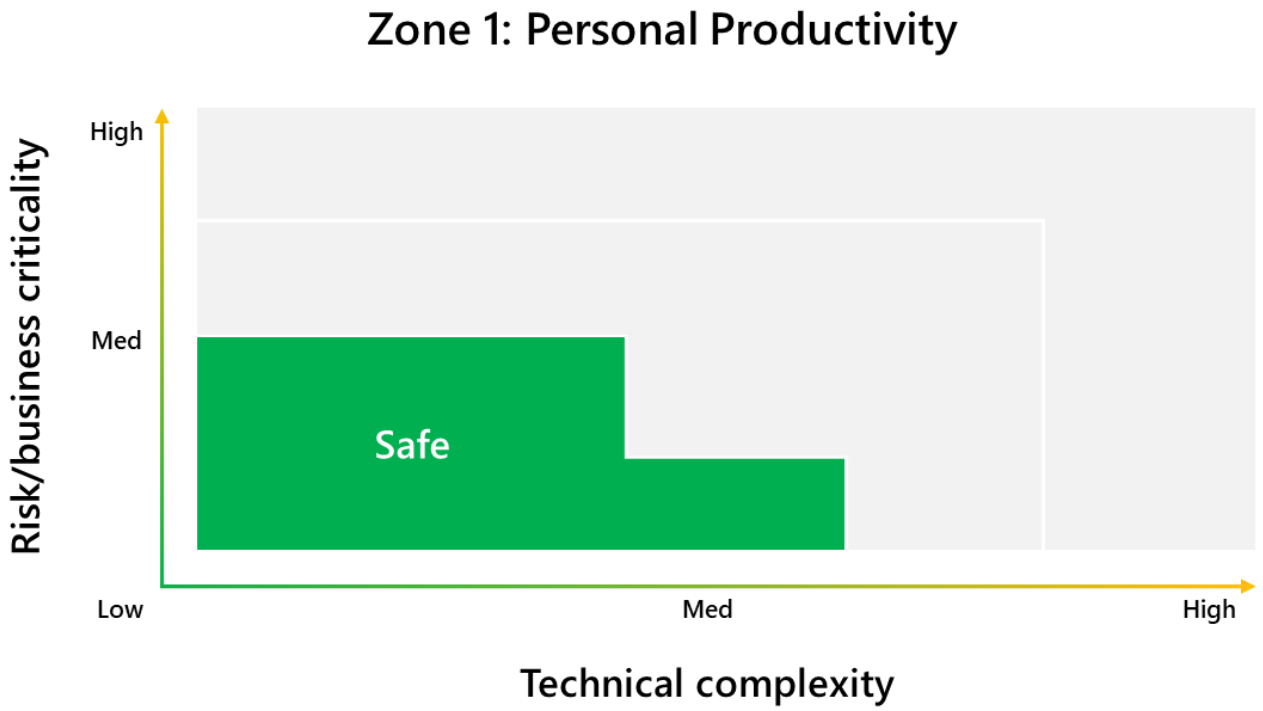


Figure 2 Personal productivity zone

In practice, Zone 1 is implemented using a dedicated environment group in the Power Platform admin center (PPAC) and typically consists of personal developer environments that are automatically provisioned and governed through managed platform capabilities. These environments provide isolation, reduce risk, and ensure baseline controls are consistently applied from the moment a maker begins building.

Zone 1 is intentionally optimized for experimentation and learning rather than broad sharing or production workloads. Assets created here are expected to remain personal, short-lived, or transitional, with a clear path to promotion into higher governance zones as scope, usage, or business impact increases.

4.6.1.1. Platform characteristics

Assets in the personal productivity zone are characterized by low business risk, limited data exposure, and individual ownership.

Within Zone 1 environments:

- Power Apps are typically small, task-focused applications built for individual use, often leveraging Microsoft 365 data sources such as SharePoint or Dataverse with limited scope.
- Power Automate flows support personal automation scenarios, such as notifications, approvals, or data synchronization related to the maker's own workload.
- Agents created with Copilot Studio assist individual users with productivity tasks or information retrieval and are not designed for team or organization-wide consumption.

These environments are governed to ensure users can innovate safely without introducing unmanaged sprawl, data leakage, or dependency risk. All resources remain owned by the individual maker and operate within boundaries defined by platform policy rather than ad-hoc guidance.

4.6.1.2. Security controls

Zone 1 applies a baseline security posture that protects organizational data while minimizing friction for individual makers.

The Power Platform admin center provides a control plane for managing Power Platform environments, connectors, and runtime behavior across apps, flows, and agents.

For Zone 1 environment groups, administrators typically configure:

- Restricted connector availability using advanced connector policies, allowing only low-risk, tenant-approved services.
- Identity-based access, where assets execute under the maker's own identity and permissions.
- Environment-level settings that prevent elevation of privileges or bypassing of platform safeguards.

These controls ensure that personal productivity assets cannot access data or services beyond what the maker is already authorized to use.

4.6.1.3. Management controls

Management controls in Zone 1 focus on containment, clarity of ownership, and lifecycle visibility, rather than scale or enterprise reliability.

Zone 1 environments are grouped using environment groups and configured as managed environments, allowing administrators to enforce consistent settings across all personal developer environments. This approach prevents configuration drift and eliminates the need for one-off environment tuning.

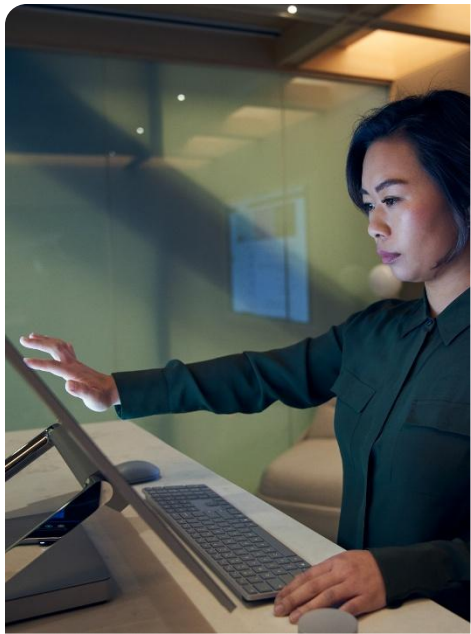
Common management controls include:

- Automatic ownership assignment to the individual user.
- Limited or disabled external sharing.
- Welcome content and in-product guidance that clearly communicates usage boundaries and promotion expectations.

By embedding these rules into the environment group, governance is applied by default rather than enforced retroactively.

Assets created in Zone 1 are expected to remain personal. Sharing is typically limited to the individual maker only, or a very small, explicitly approved review audience (for example, viewer-only access).

If an app, flow, or agent needs to be shared more broadly, integrated into a business process, or maintained over time, it should be promoted into a Zone 2 environment using solutions and managed ALM processes. Zone 1 is not designed to scale production workloads.



4.6.1.4. Reporting and visibility

Effective governance requires visibility, even at the personal productivity level. Zone 1 leverages platform-wide reporting to ensure assets remain discoverable, auditable, and manageable.

The Inventory experience in PPAC provides a tenant-wide view of all Power Platform assets, including apps, flows, and agents, regardless of environment. This inventory allows administrators to:

- Identify assets created in personal environments.
- Understand ownership and usage patterns.
- Detect inactive, orphaned, or high-growth resources that may require attention.

Inventory establishes the foundation for governance by making personal productivity assets visible without requiring manual discovery or user reporting. As solutions mature, this visibility helps organizations identify candidates for promotion into higher governance zones with appropriate controls and operational ownership.

4.6.2. Zone 2 (yellow) – Team collaboration

Zone 2 builds on the personal productivity foundation of Zone 1 and introduces shared, governed collaboration across teams and departments. This zone is designed for apps, flows, and agents that support team-level processes, require shared ownership, or interact with departmental data, while still remaining within clearly defined governance boundaries.

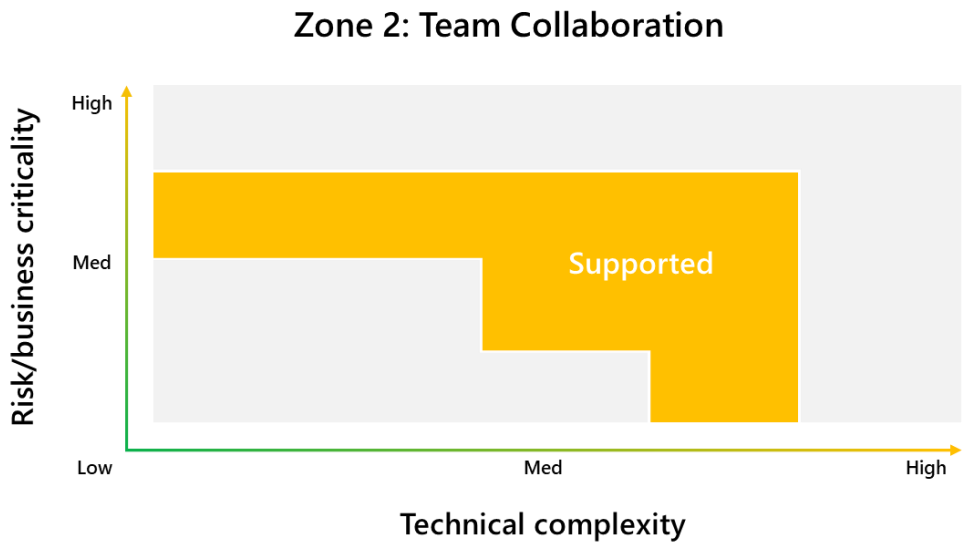


Figure 3 Team collaboration zone

Zone 2 environments are implemented as managed environments grouped under a dedicated environment group in the Power Platform admin center (PPAC). Governance in this zone shifts from individual containment toward controlled collaboration, balancing increased capability with stronger enforcement, visibility, and lifecycle discipline.

Assets in Zone 2 are expected to be intentional, reusable, and supportable, with a clearer promotion path toward enterprise-managed deployment when business criticality increases.

4.6.2.1. Platform characteristics

Zone 2 supports collaborative solutions that are used by multiple users within a team, function, or department.

These include:

- Power Apps used by teams to manage shared processes (for example, intake, approvals, or operational tracking).
- Power Automate flows that orchestrate departmental workflows, integrations, or system interactions.
- Agents built with Copilot Studio that assist teams with task execution, information retrieval, or process automation, operating within defined connector and data boundaries.

Unlike Zone 1, assets in Zone 2 are intentionally shared, may have multiple contributors, and are expected to follow a lightweight but consistent application lifecycle model. Ownership is defined at the solution level rather than the individual level, and assets are treated as team resources rather than personal experiments.

4.6.2.2. Security controls

Security controls in Zone 2 are more restrictive than Zone 1, reflecting the increased data access, sharing scope, and operational impact of collaborative solutions.

Power Platform admin center remains the central control plane for Zone 2 governance, enforcing security consistently across all apps, flows, and agents within the environment group.

For Zone 2 environment groups, administrators typically configure:

- Advanced connector policies that allow a broader but still curated set of connectors suited to departmental use, while blocking high-risk or external services.
- Explicit access scoping using Entra ID security groups, ensuring only approved makers and users can create, modify, or run solutions.
- Enforcement of environment-level security baselines that cannot be reduced by local environment administrators.

These controls ensure collaboration does not come at the expense of data protection or compliance, and that solutions operate only within approved integration boundaries.

4.6.2.3. Management controls

Management controls in Zone 2 formalize how collaborative solutions are created, changed, promoted, and operated, without introducing production-grade overhead.

All Zone 2 environments are configured as managed environments and grouped into a Zone 2 environment group. This allows administrators to apply rules that standardize:

- Sharing defaults and limitations
- Connector availability
- Solution and pipeline requirements
- Maker onboarding experiences

By enforcing these settings at the environment-group level, organizations prevent configuration drift and ensure every collaborative workspace behaves consistently.

Unlike Zone 1, assets in Zone 2 are expected to follow a defined lifecycle:

- Assets are packaged using solutions
- Changes are tracked and promoted intentionally
- Environments may be aligned to a typical development process of dev, test and prod.

While full enterprise ALM is not always required, Zone 2 establishes the discipline necessary to safely scale collaboration and prepares solutions for promotion into Zone 3 when appropriate.

4.6.2.4. Reporting and visibility

As collaboration increases, visibility becomes essential for effective governance.

Zone 2 relies on platform level reporting in PPAC to ensure administrators and governance teams have insight into usage, risk, and operational health across shared environments.

The Inventory experience in PPAC provides a consolidated view of all apps, flows and agents across Zone 2 environments. This allows administrators to understand:

- What shared solutions exist
- Who owns and maintains them
- How broadly they are being used

Inventory data helps identify high-value solutions, detect unmanaged growth, and determine when assets should move toward enterprise management or retirement.

There's also the Security experience in PPAC aggregates security insights across Zone 2 environments, highlighting connector usage patterns, configuration risks, and policy misalignments. By surfacing actionable recommendations, the Security experience supports proactive governance rather than reactive cleanup.

Finally, Monitoring experience in PPAC provides operational insight into the health and performance of collaborative apps, flows, and agents. This visibility enables early detection of issues that may impact teams and ensures shared solutions remain reliable as adoption grows.

4.6.3. Zone 3 (red) – Enterprise managed

Zone 3 represents the highest level of governance maturity for the Power Platform and is intended for business-critical, high-impact, or organization-wide solutions and builds upon both Zone 1 and 2. This zone is designed for apps, flows, and agents that operate at scale, integrate with sensitive or regulated data, and require strong guarantees around security, reliability, compliance, and supportability.

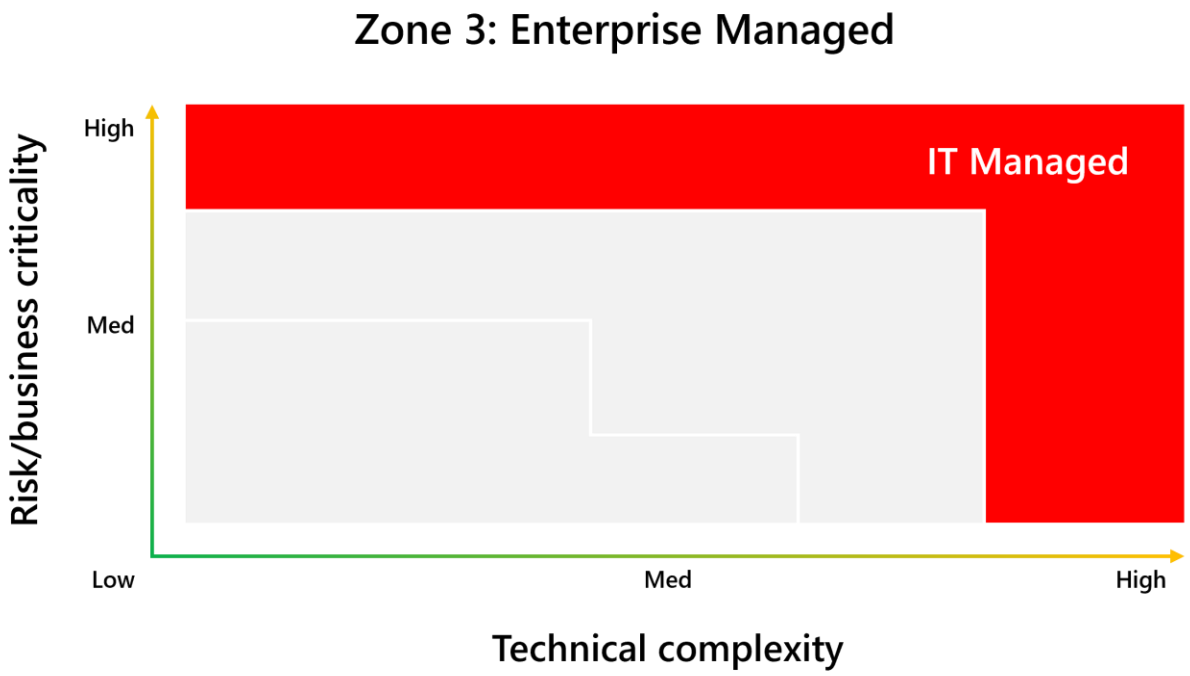


Figure 4 Enterprise managed zone

Environments in Zone 3 are centrally owned and operated typically by IT alongside a Center of Excellence (CoE) or a dedicated platform team and are governed through managed environments and environment groups in the PPAC. Governance in this zone shifts from collaboration enablement toward enterprise assurance, ensuring that solutions can be trusted as part of the organization's core digital ecosystem.

4.6.3.1. Platform characteristics

Zone 3 moves to enterprise-grade workloads that are:

- Used by large audiences or across multiple business units
- Dependent on critical systems or sensitive data
- Expected to meet formal requirements for availability, performance, and compliance

The workloads in Zone 3 focus on Power Apps that function as line-of-business applications or external-facing experiences, Power Automate flows that orchestrate critical processes, system integrations, or operational automation and Agents built with Copilot Studio that provide shared AI-powered capabilities across teams or the entire organization.

These assets are treated as products rather than projects. Ownership is explicit, operational responsibility is defined, and changes are introduced in a controlled and auditable manner.

Zone 3 solutions are expected to follow standardized lifecycle practices and align to enterprise support and risk management models.

4.6.3.2. Security controls

Security in Zone 3 is intentionally strict, reflecting the elevated risk and business impact of enterprise-managed solutions.

PPAC is the primary enforcement point for Zone 3 security, providing centralized management for environments, connectors, runtime behavior, and cost controls across the Power Platform.

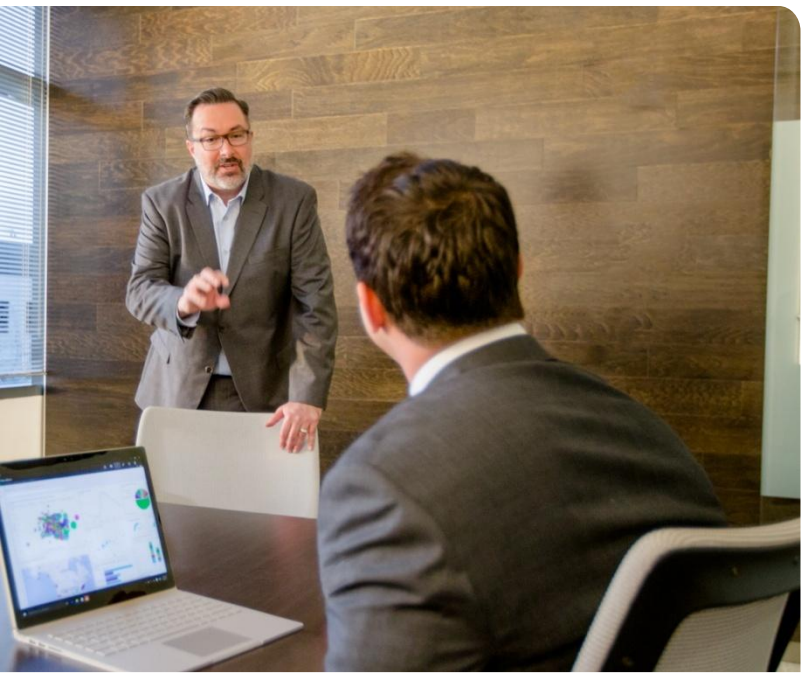
Key security characteristics similar to Zone 2 include:

- Advanced connector policies that allow only explicitly approved connectors and, where supported, restrict specific connector actions.
- Identity-based access controls scoped through Entra ID security groups, with clear separation between makers, operators, and consumers.
- Enforcement of platform-wide security baselines that cannot be overridden locally.

These controls ensure enterprise solutions cannot introduce unmanaged integration paths or expand their data access beyond what has been formally approved.

Zone 3 environments are explicitly designed to align with organizational security and compliance programs. Power Platform governance in this zone works together with Microsoft Purview to honor data classification, retention, audit, and compliance policies that apply across the broader Microsoft estate.

This ensures that apps, flows, and agents in Zone 3 operate within the same trust boundaries as other enterprise systems.



4.6.3.3. Management controls

Management controls in Zone 3 are focused on consistency, predictability, and operational resilience.

The Zone 3 environments are configured as managed environments, assigned to a Zone 3 environment group and governed by environment group rules that lock key settings across all environments in the zone.

These rules enforce standards such as connector availability, sharing defaults, pipeline requirements, and environment configuration, ensuring that governance remains intact as solutions evolve.

Zone 3 solutions are expected to follow a full application lifecycle management practice, including:

- Solution-based development
- Versioned promotion through development, test, and production environments
- Controlled deployment using Power Platform pipelines

Pipelines provide a repeatable and auditable mechanism for promoting apps, flows, and agents between environments, incorporating approvals, environment-specific configuration, and operational checks. This reduces risk while enabling teams to deliver changes at enterprise scale.

4.6.3.4. Reporting and visibility

At enterprise scale, governance depends on deep, continuous visibility into usage, risk, and operational health. Zone 3 relies on tenant-wide reporting experiences in PPAC to provide that insight. Similarly to Zone 2, the Inventory experience in PPAC serves as the authoritative catalog for all enterprise-managed Power Platform assets.

The security experience consolidates information across Zone 3 environments, surfacing configuration risks, connector exposure, and policy misalignments. Then the monitoring experience provides health and performance signals across enterprise workloads, including failure patterns, degradation trends, and usage anomalies.

4.6.4. Zone Example

	"Green" Zone →	"Yellow" Zone →	"Red" Zone
Purpose	Citizen Dev agent creation (DIY) for personal use and experimentation with safe defaults	Team or Department Agents in the partnered DIY zone require formal assistance and oversight from a DIY coach, but are built by trained citizen developers	Large, potentially risky agents in the professional development zone are reserved for pro dev & IT-led development only
Secure	Only M365 and Power Platform Connectors Agents run in user's context only	Zone specific Advanced Connector policies in Power Platform Admin Center Teams share access to approved Data sources Scale with Environment groups + rules	Zone specific Advanced Connector policies in Power Platform Admin Center + Purview
Govern	Personal use agents in Developer Environments. Environment routing keeps agents isolated to maker. Sharing limited and scoped to just Maker use	Admin approved environments provisioning Scoped roles and sharing policies ALM Pipelines for agent versioning IT-admin approval to publish agents	Manage sharing via Integrated Apps in Microsoft Admin Center
Monitor	Review agent usage in Copilot Hub in Power Platform	Track agent usage and security posture in Microsoft Admin Center, Microsoft Purview, and Power Platform Admin Center	

Figure 5 Zone Example

5. Security

Security in Power Platform is implemented through layered controls that span tenant governance, environment boundaries, connector and data controls, and Dataverse's authorization model. As adoption scales across apps, flows, and agents, security considerations should be focused on continuously assessing risk, enforcing guardrails at scale, and ensuring access aligns to business intent.

5.1. Security in the Power Platform admin center

The Power Platform admin center (PPAC) is the central place to manage security posture, apply baseline protections, and operationalize security recommendations across environments. The Security posture management experience provides a comprehensive view of your security position, includes preconfigured security defaults, and guides administrators through a focused action center for completing required security tasks. It also supports proactive governance by enabling admins to customize security settings at scale to fit business needs.

5.1.1. Data policies

For most organizations, the most common security risk is accidental data movement across trust boundaries. Data policies (Data Loss Prevention (DLP)) address this by governing which connectors can be used together, and which connectors are blocked entirely. Data policies define the consumer connectors that business data can be shared with and help prevent business data from being inadvertently published to consumer services.

5.1.2. Managed environments

When an environment is enabled as a Managed Environment, additional security capabilities become available to help reduce risk and support enterprise controls. Managed Environments include (among other capabilities) data policies, IP firewall, IP cookie binding, customer-managed key (CMK), Lockbox, auditing configuration, and masking rules.

5.2. Dataverse security

Dataverse provides a security model designed for enterprise applications and is only in effect when an environment includes a Dataverse database. IT & Administrators are typically responsible for establishing the security model foundations, ensuring users have the correct configuration, and troubleshooting access issues as solutions scale.

5.2.1. Role-based security and business units

Dataverse uses role-based security to group privileges into security roles, which can be assigned directly to users or associated with Dataverse teams and business units. A critical concept is that privilege grants are cumulative—the broadest access granted across a user's roles prevails.

Business units are a core security modeling building block that defines a security boundary. Every Dataverse database includes a single root business unit, and organizations can create child business units to further segment users and data access.

5.2.2. Security roles and privileges

Security roles define how different users access different types of records, and users can have multiple roles where privileges accumulate. PPAC provides a direct path to view and manage security roles for a given environment.

5.2.3. Column-level security

Many enterprise solutions require protecting specific sensitive fields (for example, identifiers or regulated attributes) even when a user can access the record. Dataverse supports column-level security, allowing administrators to control who can view or update specific columns. This is configured by enabling column security on a column and granting access through column security profiles (assigned to users or teams).

6. Change management

Power Platform enables rapid creation of apps, automations, and agents across teams, but that speed also increases the likelihood of inconsistent practices, unmanaged sprawl, and fragile solutions unless change is intentionally managed.

Change management in Power Platform must account for two factors:

- The platform itself brings continuous updates as Microsoft delivers new capabilities, security improvements, and admin experiences through cloud updates.
- Your organization's assets change continuously as makers and teams iterate on apps, flows, sites, and agents, often daily, based on business needs and user feedback.

A modern change management approach therefore combines not just process and platform controls but people too, so that innovation can move quickly while remaining safe, supportable, and aligned to business risk.

6.1. What "change" means in Power Platform

Building on the previous section, for most enterprises, Power Platform change management includes three categories of change:

6.1.1. Platform change (Microsoft-driven)

These are changes that arrive through service updates, new features, deprecations, admin center experiences, security enhancements, and behavior changes. In Power Platform, some updates can be influenced through environment-level settings such as refresh cadence (for example, choosing how frequently an environment receives updates and features for certain services).

6.1.2. Tenant change (admin-driven)

These are changes your admins intentionally make to govern the platform: environment strategy updates, new environment groups and rules, routing changes, connector governance decisions, advanced connector policy updates, onboarding messaging, or standardizing deployment processes. The shift toward managed governance means these decisions are increasingly expressed as rules applied at scale, not one-off configuration.

6.1.3. Solution change (maker/team-driven)

These are changes to the business assets built on the platform: new apps, updated flows, revised agents, connector additions, schema changes, or new integrations. Solution change must be managed through lifecycle practices (solutions, pipelines, approvals, testing, and rollback planning) that match the business criticality of the workload.

6.1.4. User change (people-driven)

These are user experience changes, Power Platform can move quickly and that speed is a feature, but it also means small changes in behavior (where makers build, what they connect to, how they share, and how they deploy) can create outsized operational and risk impact if the organization is not brought along intentionally. This is especially true as apps, flows, and agents expand beyond experimentation into real business processes.

6.1.4.1. Skilling is role-based

"Users" in Power Platform is not one audience. Your change plan should explicitly skill the groups that introduce and experience change differently:

- Admins / platform operators who manage tenant and environment governance (environment strategy, routing, environment groups/rules, connector governance, monitoring and lifecycle controls).
- Makers (new and occasional) who need safe defaults, clear guardrails, and quick-start guidance so they don't accidentally create sprawl or fragile solutions.
- Advanced makers / solution owners who move solutions across environments and need lightweight ALM discipline (solutions, pipelines, validation) that matches business risk.
- End users who consume apps, flows, and agents and need trust, clarity, and feedback loops so adoption is intentional and measurable.

This aligns directly to the governance zones described earlier in the whitepaper: Zone 1 requires skilling focused on safe personal productivity and "graduation" expectations; Zone 2 requires collaboration patterns and solution ownership; Zone 3 requires operational rigor, support readiness, and controlled release practices.

6.1.4.2. Skilling pattern

A practical way to structure this is to follow a pattern that focuses on a tailored, role-based skilling plan backed by expert-led training and scalable delivery. In practice, that pattern typically looks like:

- Building a tailored organizational skilling plan aligned to roles and business goals
- Delivering live, instructor-led training with real-time Q&A
- Scaling enablement through high-impact webinars (and smaller on-demand content) for broad audiences.

6.1.4.3. Anchor skilling to the "moments that matter"

Instead of treating training as a one-time event, design skilling around the points where users make decisions that create risk or operational cost:

- Building: Where should I build? What's allowed in this environment/zone?
- Connecting: Which connectors/data sources are appropriate? What is restricted by policy?
- Sharing: Who can I share with, and what ownership/support expectations change when I do?
- Promotions: How do I move from dev/test to production safely (solutions, pipelines, validation)?
- Incidents: How do I troubleshoot responsibly and escalate when something fails?

6.1.4.4. Build a community within your organization using a 90 Day Plan

Building champions within your organization helps scale up skilling initiatives and helps unblock employees as they become makers and help to become more productive. Visit the link below to help create a community and have champions who in turn unblock makers for organizational growth because platform, tenant, and solution change are continuous, skilling must be continuous too, refreshed as policies evolve and as solutions mature from personal productivity to shared and enterprise-managed use.

Please visit https://learn.microsoft.com/en-us/microsoft-copilot-studio/guidance/community-framework-overview/ to learn more about making a plan.

6.2. Establish decision rights and a change operating model

Before implementing tooling, organizations should define who is allowed to introduce which types of change, and what approvals are required. This is most successful when structured as a lightweight operating model rather than an overly centralized bottleneck.

An example of an enterprise model includes:

- Platform owners (Power Platform admins) who own tenant-level configuration, environment strategy & governance enforcement.
- Zone owners (or environment group owners) who own the standards for a governed zone and manage drift prevention through environment group rules.
- Solution owners who are accountable for business outcomes, supportability, and lifecycle decisions for apps, flows, and agents.
- A change review cadence (weekly or biweekly) that reviews upcoming platform changes, high-impact solution releases, policy changes, and operational issues.

This model should stay risk-based, not every change requires heavy review, but high-impact changes should be visible and handled efficiently.

6.3. Use release rings to roll out change safely

One of the most effective patterns for cloud services is a "release ring" approach: validate change in smaller scopes first, then broaden rollout as confidence increases. Microsoft guidance for cloud change emphasizes making risk-based decisions rather than disabling everything by default, this is the same approach that should be used for Power Platform.

In Power Platform, release rings can align to your environment strategy and zones previously discussed:

- Ring 0 (pilot / platform team), validate new features, governance settings, and admin workflows in a controlled internal environment group

- Ring 1 (advanced makers / departmental), expanded to selected teams and governed collaborative environments where feedback is fast and impact is bounded.
- Ring 2 (enterprise), roll out broadly to managed environments that host business-critical apps, flows, and agents with standard deployment and monitoring expectations.

Where applicable, a refresh cadence can be implemented to be part of your ring strategy, giving you a controlled way to manage how quickly updates are introduced for certain experiences.

	Release Rings (CM)	Zones (Governance)
Core purpose	Manage *change* over time	Manage *risk* at rest
Primary question	"When and to whom do we roll this out?"	"Where is this allowed to run?"
Control type	Social and process control	Technical and policy control
Success signal	Confidence, readiness, adoption	Compliance, isolation, safety

6.4. Operationalize change with managed platform controls

The most sustainable way to manage change is to convert change management from a manual process into a platform-enforced process. So, what can we use that's mentioned in this whitepaper to achieve this?

6.4.1. Environment groups and rules

Environment groups allow administrators to apply standardized settings across many environments, and published rules lock those settings so they can't be overridden locally. This is one of the strongest mechanisms for preventing "configuration drift" over time as environments multiply.

6.4.2. Inventory

Tenant wide visibility is the starting point for reliable change. Inventory provides a cross-environment view of assets (apps, flows, agents) so administrators can identify what exists, who owns it, and where risk may be emerging, especially before making tenant-wide changes.

6.4.3. Monitoring and health signals

Operational monitoring reduces the time between change introduction and issue detection. It enables proactive response to failures, degradation trends and anomalies across platform workloads.

6.4.4. Pipelines

Pipelines provide a repeatable mechanism for promoting solutions across environments and are designed to support safe, governed change processes at scale. They bring ALM automation into the service and help admins implement secure, custom-tailored change processes.

6.5. Communication and enablement

Even the best admin controls fail if makers and teams don't understand what has changed, why it has changed, and what is expected of them. Successful Power Platform adoption requires both technology and culture, users need an environment where they can thrive with guardrails (Personal Development Environments).

Some example best practices may include:

- Maintain a clear "what's changing" channel (internal site or Teams channel) that summarizes tenant policy changes, new standards, and rollout timelines.
- Publish zone-specific expectations (what makers can build, share, and connect to in each zone) and reinforce them through onboarding experiences and admin communications.
- Use training and office hours to reduce friction when introducing new lifecycle requirements (solutions, pipelines, approvals, or connector governance etc).

The goal is that the documentation makes governed behavior easy to understand by having access to documentation describing what's happening and changing as well as having training to ensure everyone is on the same page.

6.6. Using agents to accelerate change management

As organizations adopt AI agents, they can also use agents to support the change management process itself but not as a replacement for governance, but as a way to reduce manual coordination and increase consistency.

What type of agents can help with this?

- An agent that answers “what changed?” using approved governance documentation and internal standards.
- An agent that guides makers through release steps (solutions, pipelines, environment variables) and routes exceptions to the right admin queue.
- An agent that summarizes tenant recommendations and governance actions surfaced through admin experiences, then turns them into a prioritized change backlog.

When using agents for change management, the key governance principle remains that agents should help execute your process, not redefine it. They must operate within the same boundaries you enforce across environments, data, and connectors.

7. Deployment

Deployment is where change management becomes operational: turning planned, reviewed change into a controlled, repeatable release motion that can scale across teams without introducing unmanaged drift or production risk. A modern deployment approach for Power Platform focuses on repeatability, environment consistency, and traceability, so apps, flows, and agents can move through the lifecycle with the same enterprise guardrails applied.

7.1. Planning

Deployment depends on decisions made earlier in platform planning. There’s foundational considerations such as environment discovery, single sign-on, security policies, and administrative roles to plan for.

As part of this planning, organizations typically define:

- An environment strategy for adopting Power Platform at scale (including routing and the use of managed environments).

- The environments used across the lifecycle such as dev, test and production, and the types of environments such as developer, trial, sandbox or production, aligned to intended use and risk tolerance.

These decisions create the structure that deployment relies on, so releases consistently land in the right place, under the right policies, with clear operational ownership.

7.2. Application Lifecycle Management fundamentals

Application lifecycle management (ALM) is a set of practices, tools, and processes that help teams build, test, release, and maintain solutions reliably as they evolve. In Power Platform, ALM is implemented by treating environments and solutions as the core building blocks for controlled change. Environments serve as containers to store, manage, and share business data, apps, and processes, and they help separate work based on security requirements or target audiences.

A healthy ALM approach also depends on separating "what the solution is" from "how it's configured" in each environment. Power Platform supports this by using environment variables and connection references to handle environment-specific settings (such as tables, connections, keys, endpoints, or site/list parameters) without hard-coding those values directly into apps, flows and agents.

This reduces deployment friction, improves consistency across teams, and creates the preconditions for automated promotion patterns through pipelines.

7.3. Pipelines

Power Platform Pipelines provide a Continuous Integration and Continuous Deployment (CI/CD) mechanism that automates solution deployments across environments, streamlining ALM for makers, admins, and developers. Admins can set up governed deployment pipelines in minutes, makers can deploy solutions with just a few clicks, no manual exports/imports, and pro developers can extend or run pipelines through the Power Platform Command-Line Interface (CLI) for more advanced scenarios.

7.3.1. Core setup requirements

To use pipelines, each participating environment must have a Dataverse database, and all target environments must be enabled as Managed Environments. It's recommended to designate a dedicated "pipeline host" environment (a production Dataverse environment) to

store pipeline configurations, run history, and security roles. This host serves as the central management plane for your pipelines. You should plan at least 3-4 environments for a healthy ALM process (e.g. Dev, Test/UAT, and Prod, plus a host environment). The pipeline feature itself is delivered via a Power Platform Pipelines managed solution that an admin installs in the host environment using the Power Platform Admin Center (PPAC).

Once installed, an admin configures the pipeline by registering each environment (Dev or Target) in the host's Deployment Pipeline Configuration app and defining the stages (deployment steps) of the pipeline (e.g. Dev → Test → Prod).

Security is controlled with two roles provided by the pipeline's solution; Deployment Pipeline Administrator (full control to configure pipelines) and Deployment Pipeline User (to run shared pipelines). Which are assigned in the host environment to the appropriate users or groups. Makers who receive the Deployment Pipeline User role (and have sufficient rights in the source and target environments) can then run pipeline deployments for solutions.

7.3.2. Platform host vs. custom host

Platform Host (Personal Pipelines) is a default, tenant-wide pipeline host that makers can use to create personal pipelines directly, without upfront admin configuration. This allows an individual maker to deploy from a dev environment to up to two target environments (for example, Dev → Test → Prod) on their own. The platform host is provisioned automatically the first time a user opens the Pipelines page in an environment that isn't already linked to a custom host. Personal pipelines are quick to set up and require no code or DevOps tools, makers simply click "Create pipeline" in their solution and follow the prompts.

However, this mode has limitations to help IT admins: personal pipelines support at most 3 environments (one dev & two stages), cannot be shared with other users (the pipeline is only visible to its creator), and cannot be extended with custom automation or connectors. The platform host is ideal for simple, individual ALM needs but is not suited for multi-team scenarios or complex workflows. Admins can prevent makers from inadvertently using personal pipelines by ensuring those dev environments are linked to a custom host, if an environment is already tied to a custom pipeline host, makers won't be able to create a personal pipeline in it.

The custom host (Managed Pipelines) is a pipeline host environment that an admin sets up to centrally govern ALM in PowerPlatform. This approach is recommended for enterprise scenarios because it supports the full range of the pipeline capabilities. You can include multiple stages (up to 7), involve multiple makers, and enforce governance policies. Custom-hosted pipelines are shareable, once the admin configures the pipeline in the host environment

and shares it, any user with the Deployment Pipeline User role can run deployments (with proper environment permissions).

Custom pipelines also allow advanced extensibility: for example, admins can integrate approval workflows, use the Power Platform CLI to script pipeline runs, or even trigger pipeline deployments from Azure DevOps or GitHub actions. All pipeline run data and configuration are stored in the Dataverse of the host environment, giving admins full visibility.

7.4. Managing environments

A common cause of deployment failures isn't the solution itself, it's the configuration around it. The same app, flow, or agent often needs to point to different resources in each environment (different SharePoint sites, endpoints, keys, connections, or databases), especially when going through the dev and test stages.

Power Platform's ALM model addresses this directly with environment variables and connection references, so teams do not hard-code environment-specific values into components that then break as the solution moves. Environment variables store parameter keys and values separately from the sections that use them, which makes it straightforward to keep the solution consistent while changing only what must differ between development, test, and production. Because environment variables are solution components, teams can transport the references (keys) and set the appropriate values as the solution is promoted.

When teams later move toward automation, Microsoft provides deployment settings files (JSON) so connection references and environment variables can be prepopulated for the target environment, removing the need to enter values interactively after importing a solution and making CI/CD scenarios more reliable.

Managed Environments strengthen this model by making the target environment "release ready." In practice, even if your configuration is perfect, deployments still fail (or create governance gaps) when target environments don't have consistent guardrails. Managed Environments are designed to activate governance capabilities so admins can apply policies consistently at scale. Enabling a Managed Environment is a simple admin action (for example, in the admin experience you can navigate to environments and use the Managed Environment toggle), and it unlocks controls that reduce deployment risk such as consistent policy application across environments and stronger governance posture.

This becomes especially important once you introduce pipelines. This is due to pipeline target environments must be enabled as Managed Environment when using the custom host approach.

7.5. Extended deployment

While native pipelines cover many deployment needs, it also provides options if you require deeper DevOps integration:

- GitHub Actions guidance describes automated workflows that can build, test, package, and deploy Power Platform solutions, including importing/exporting solutions and deploying to downstream environments.
- Azure DevOps build tools are positioned for more complex enterprise workflows and integration with broader CI/CD practices.

These automation paths can also support activities such as environment provisioning and static analysis as part of an automated lifecycle.

This provides an escalation path: teams can adopt service-native deployment patterns first and introduce deeper DevOps automation where engineering standards or complexity may need it.

7.6. Validation

For deployments, quality is not a one-time activity, it's a repeatable process that reduces risk every time a solution moves forward, especially when working in enterprise.

In Power Platform, validation typically combines three layers: static quality checks, automated testing where applicable, and human approvals for risk-based releases. A structured testing and validation phase helps teams catch reliability, performance, and security issues early, so they are caught before they become production incidents.

Start with static analysis as the baseline quality gate. Solution checker performs a comprehensive static analysis of solution objects and produces a report that highlights issues, affected components and guidance on how to resolve each issue. It can be run quickly and consistently and is a great first step.

Managed Environments can also enforce solution checker validations during the solution import, turning quality from a recommendation into an operational control. This approach allows organizations to scale safely by lower-risk environments can start with a warn approach while production can adopt stricter enforcement once teams are ready.

Automation testing allows repetitive integration testing ensuring that when you make changes to your apps, flows and agents, that you're not having negative consequences in unchanged areas. This is something to consider when implementing your test strategy.

Then finally focus on human approvals, with approval-based deployments which add a governance and compliance checkpoint before production promotion. This is especially important when releases affect business-critical processes, regulated data etc.

Together, these approaches establish a consistent pattern: static analysis, enforcement for critical environments, automated tests where supported and approvals for risk-based releases.

7.6.1. Agent evaluations

AI agents can drift over time as models, grounding data, instructions, tools, and context change. Agent evaluations (evals) provide a repeatable, evidence-based way to validate agent behavior beyond a few static manual spot checks, helping teams build confidence before a larger rollout.

In Copilot Studio, evals use test cases (a user question that can be paired with an expected answer) grouped into test sets that can be run repeatedly as the agent evolves. Teams can create test sets from realistic prompts (including manually authored scenarios, imported datasets, AI-assisted generation, or past conversations) and score results using configurable graders (for example, quality/completeness, classification of expected behavior, or whether the agent used the right capability at the right time). This gives you a more realistic test scenario than just static box checking, mimicking what happens in real scenarios.

For production readiness, run evals under an explicit user identity context so results reflect the same access boundaries users will experience, and review outcomes at two levels: aggregate trends across the set and case-level explainability to understand why specific tests pass or fail. Re-run and compare evaluation results after changes to confirm improvement and catch regressions early.

Agent evaluation complements (but does not replace) other validation controls and should be used together.

8. Resources

Agent Governance and Security whitepaper