Microsoft Power Platform

Importante!

- Material elaborado pelo time de especialistas em Power Platform da Microsoft Brasil para ser utilizado em **demonstrações online para seus clientes**.
- O uso deste material é permitido **apenas para clientes** Microsoft que participaram da sua apresentação em ambiente online e para **fins de referência e/ou autoestudo**.
- Este conteúdo é de **propriedade intelectual da Microsoft**. Todos os direitos reservados.
- Este material utilizou como referência as funcionalidades da **Power Platform** disponíveis conforme a **documentação oficial** e previstas para lançamento em seu **guia de lançamento** na data de sua elaboração (ver slide de abertura). Como o conteúdo destas fontes é dinâmico, sempre consulta-las em caso de dúvidas.

Em caso de dúvidas ou questões acesse nosso site (Power Platform Connect)
https://microsoft.github.io/powerplatformconnect/

Microsoft Power Platform

Fusion development with Power Platform

Version 1.0 from Sep 12, 2024

Agenda

Why fusion development? (12 mins)

ALM basics with Power Platform (27 mins)
Core platform concepts that enables ALM

Power Platform tools for fusion development (41 mins)
Tools available for admins, makers and pro-devs

Creating and sharing custom components (15 mins)
How to create and share canvas and code components

Before you go (10 mins)
Final tips and recommendations

Closing (05 mins)

Why fusion development?

Custom code isn’t keeping up with business demand

1 billion
apps will be built 2028.[1]

Tech talent limitations

Ground up software development takes time

Surging digital demand

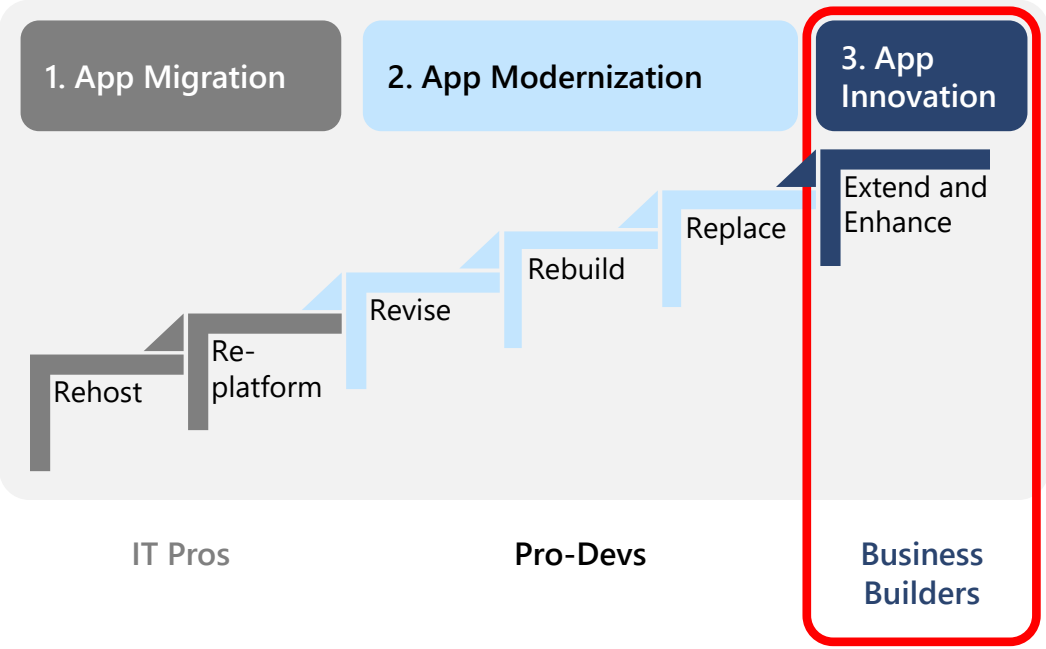
Security & compliance risks

70% **of new applications developed by enterprises will use low code or no-code technologies by 2025. – Gartner[2]**

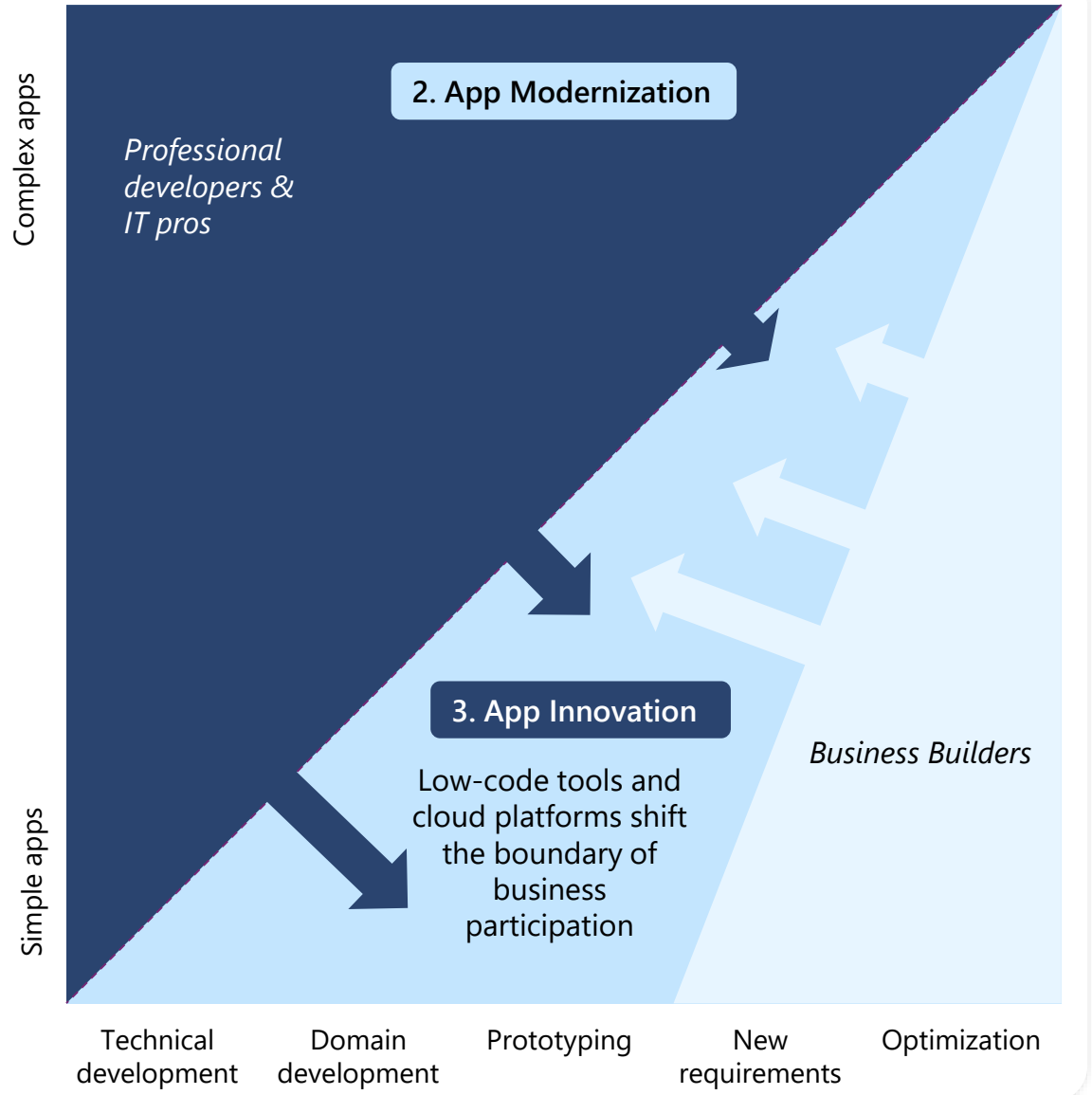
Source: IDC, 1 Billion New Logical Applications: More Background, Doc #US51953724, April 2024
2. Gartner, Forecast Analysis: Low Code development Technologies. January, 2021.

IDC Predicts:

By 2027, the share of non-technology-focused people in companies who will spend 10 hours or more a week contributing to digital innovation will grow from 5% today to 45%.



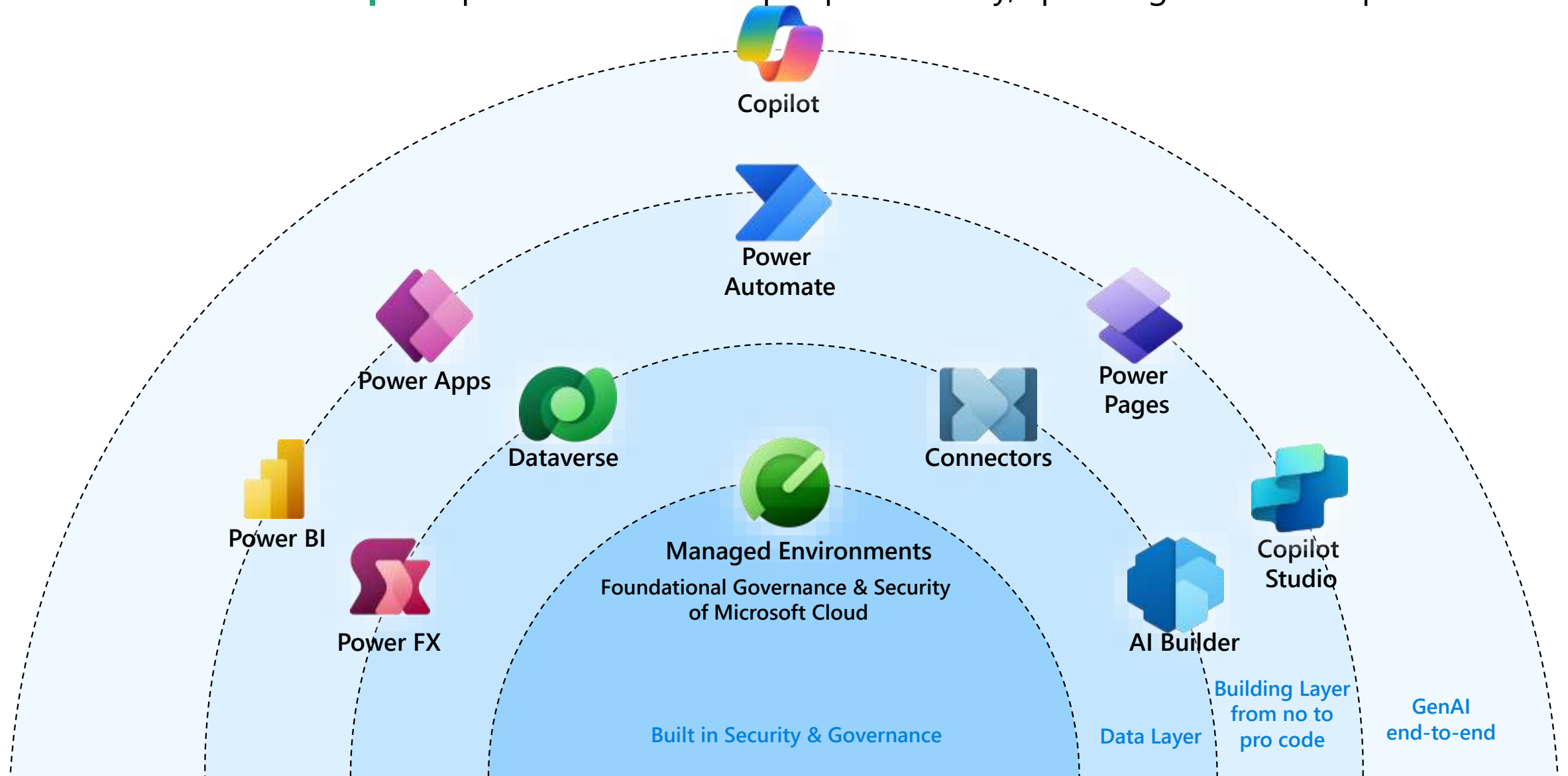
App Modernization Becomes Super-Charged via low code extensions into the business



Accelerate developers and empower new makers

Microsoft Power Platform

The world's **most complete** platform for developer productivity, spanning no-code to pro code



AI & Low-code empower everyone

IT Professional

Manage thousands of apps at scale

Code-First Developer

Use familiar tools, accelerate progress and focus on business logic

One platform for collaborative development

Maker

Contribute to building apps and help bridge the talent gap

AI & Low-code enables fusion teams

IT Professional

Manage thousands of apps at scale

Fusion team

Code-First Developer

Use familiar tools, accelerate progress and focus on business logic

Maker

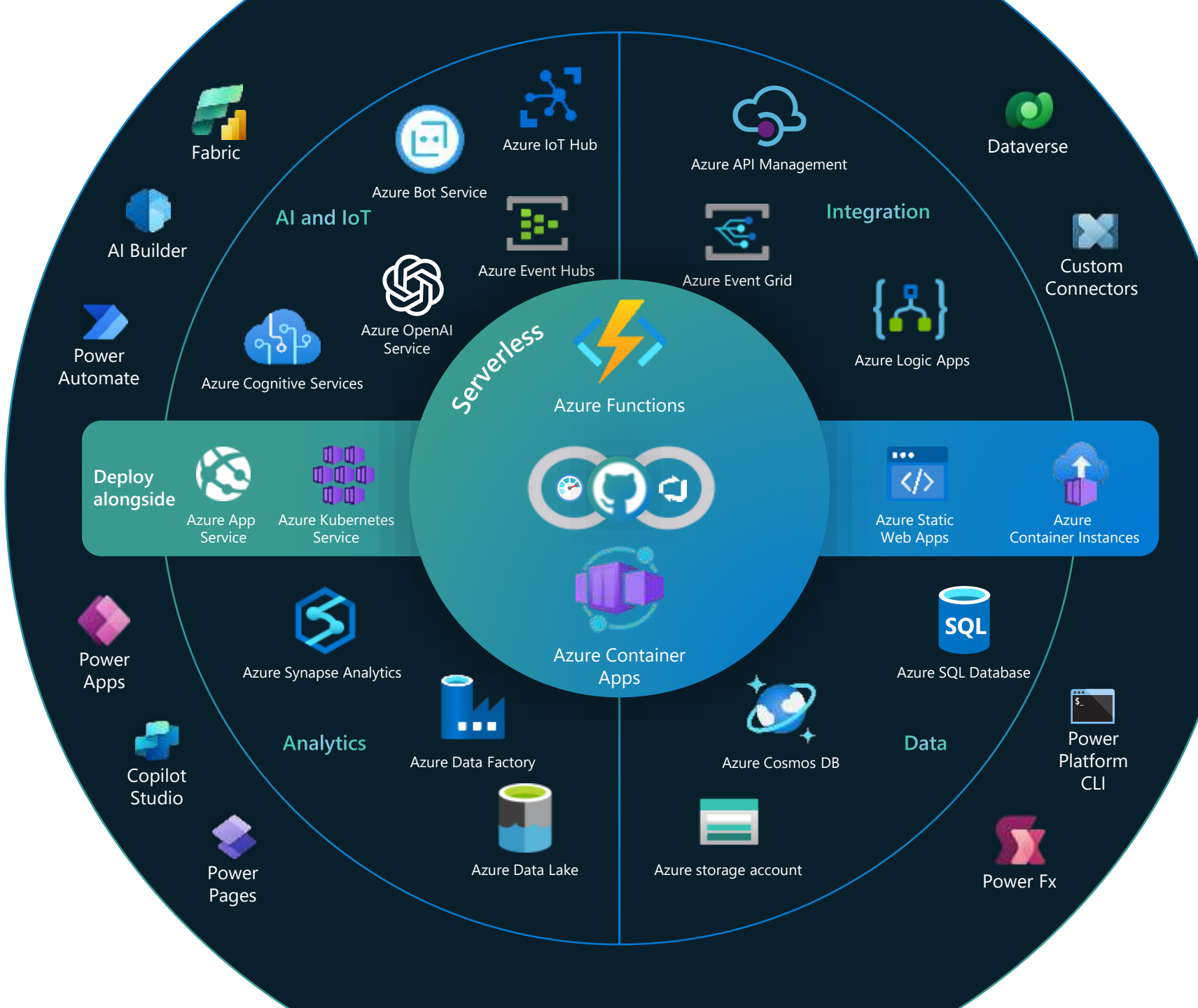
Contribute to building apps and help bridge the talent gap

AI & Low-code enables fusion teams

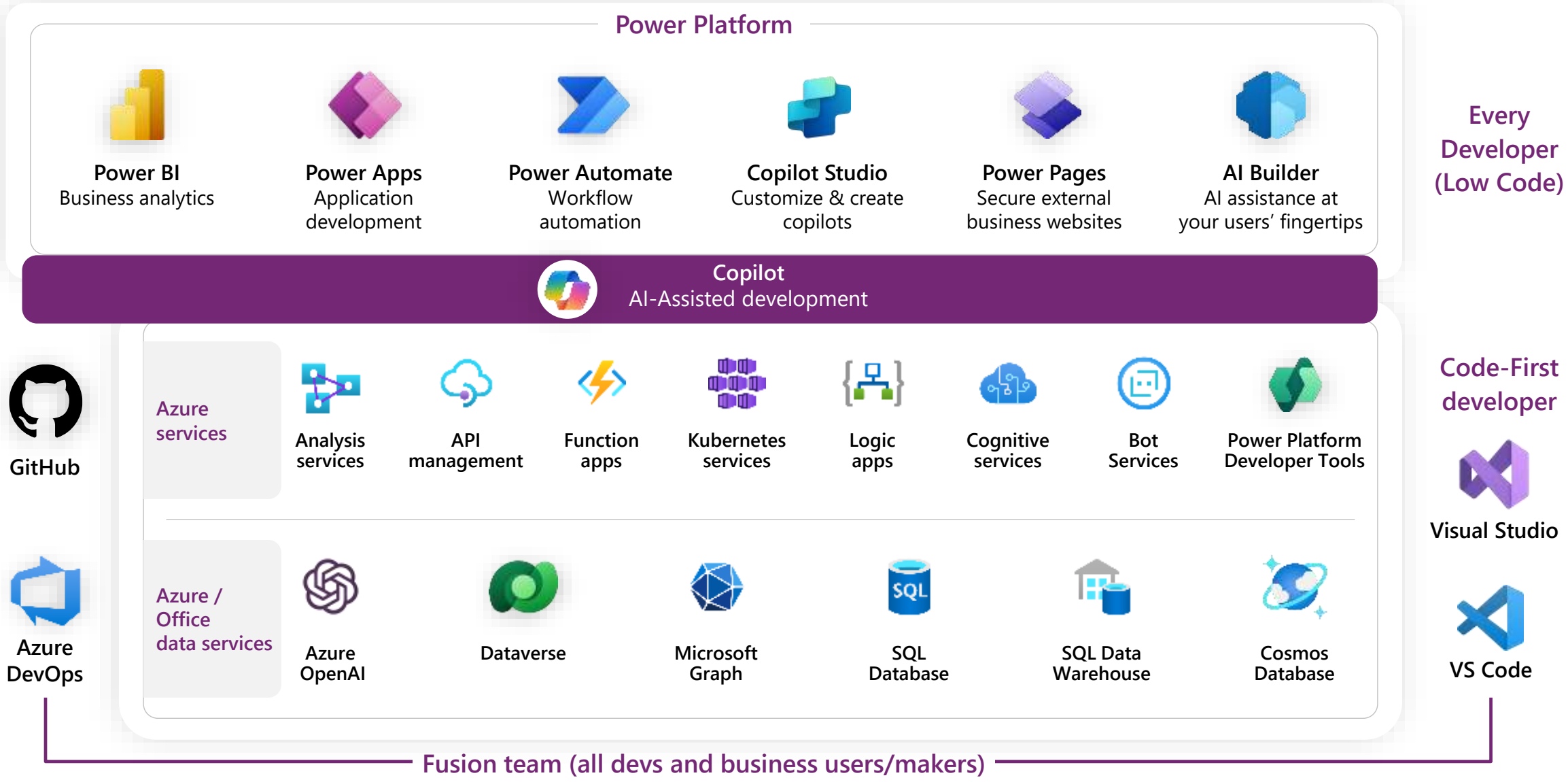
Digital "*fusion teams*" are distributed and **multidisciplinary** digital business teams that **blend technology** and other types of domain **expertise**.

**Gartner study: Fusion Teams: A New Model for Digital Delivery. Originally published August 2, 2019; refreshed February 4, 2021. Study ID G00710746*

Runs on Azure
Extends Azure



Develop Faster than Ever Before



Leverage existing Skills and Code Investments

Use the tools you're familiar with

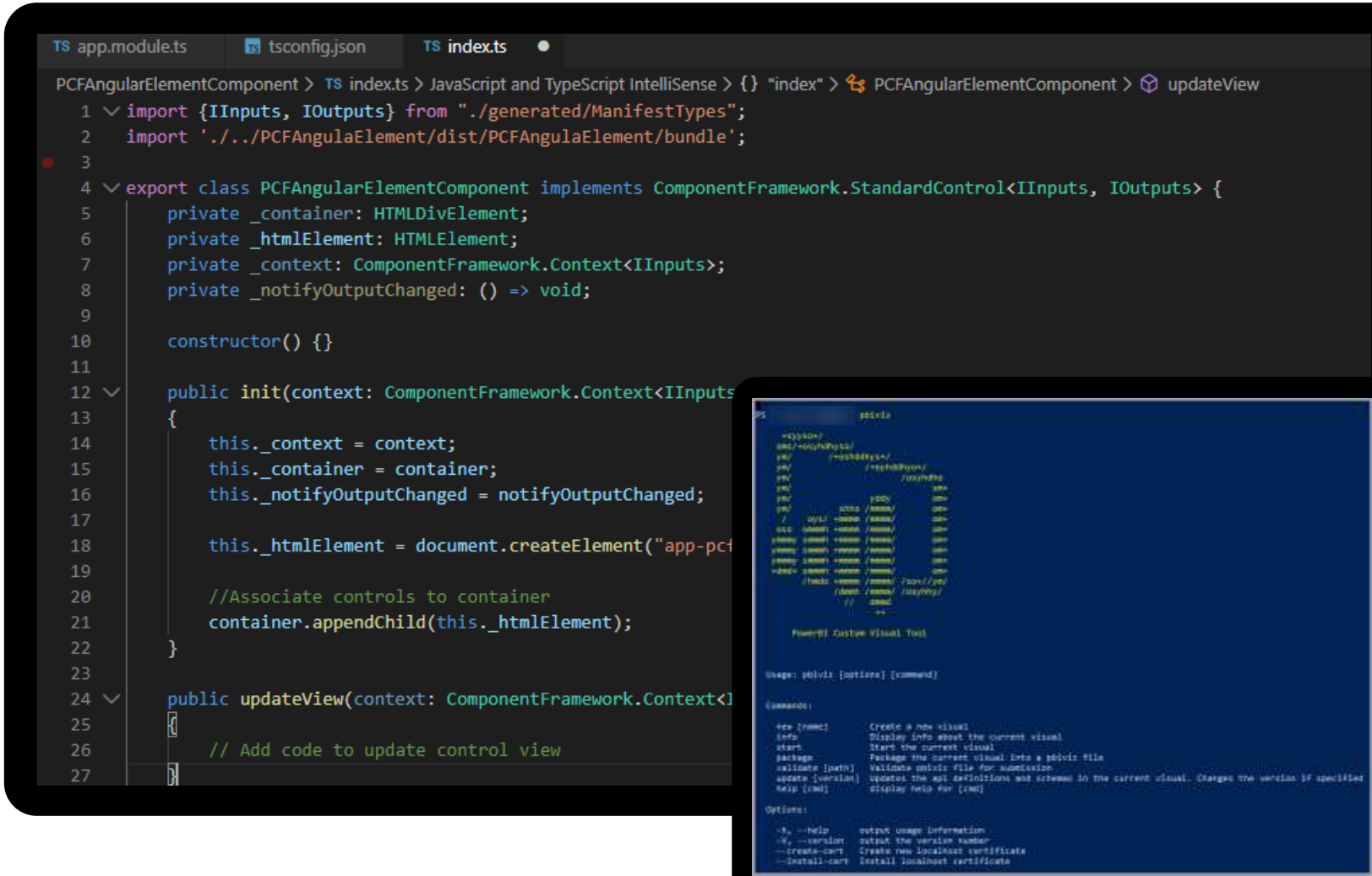
Extension for visuals in the Power Platform is based on TypeScript

Developers can use Visual Studio or Visual Studio Code

APIs with many helper functions

Use common frameworks like React

Command-line tools for building and testing



Composable Technology Architecture

Innovative Platforms

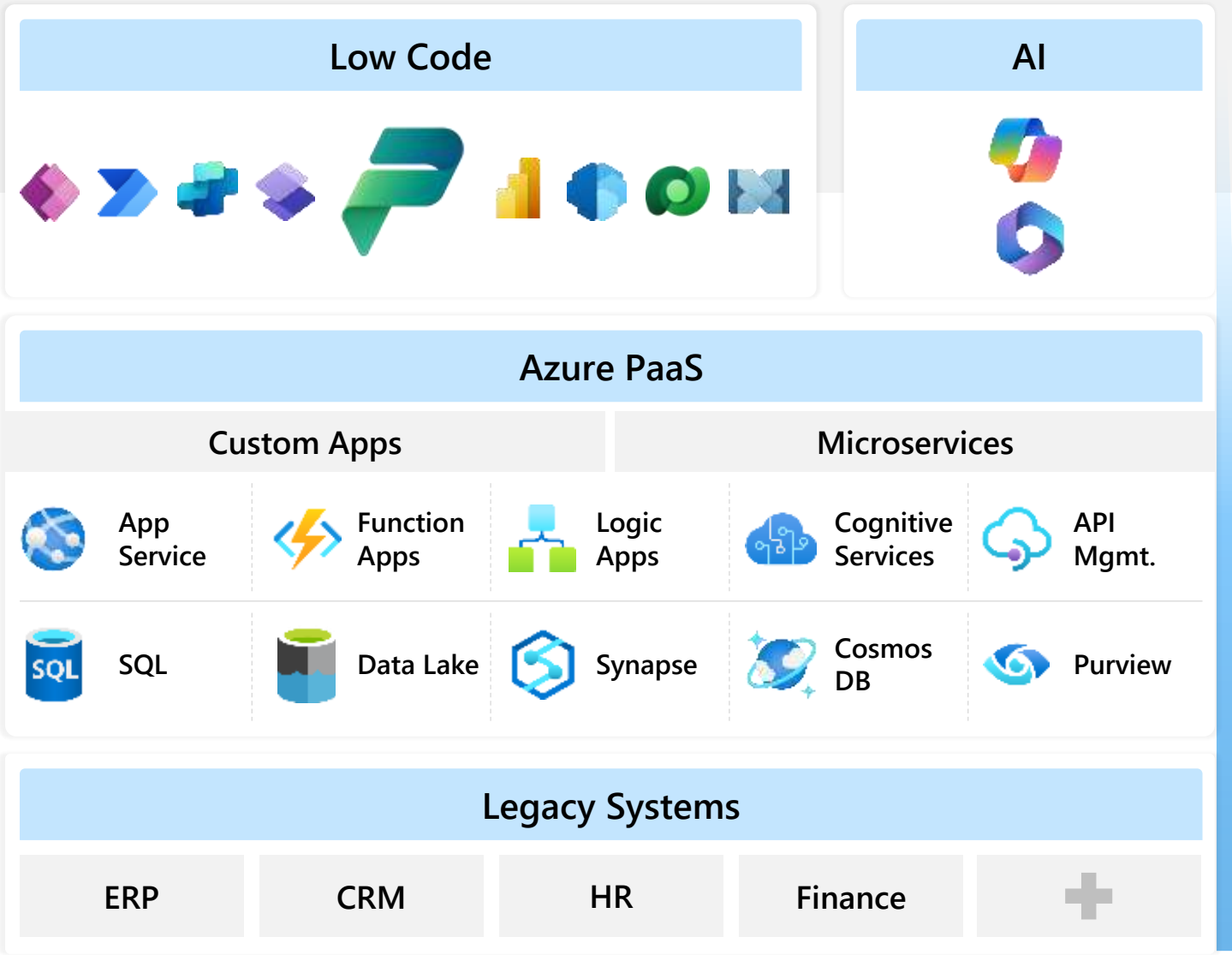
- Net new ways of solving problems
- Greatest level of experimentation
- Cultivate curiosity at the edge

Differentiated Platforms

- Represents the organization's uniqueness
- Expresses the values of the business

Legacy Systems

- Rigid "pillars" of the IT landscape
- Require high degree of process & rigor
- Change at this level incurs high risk



Building the Flagship App

Experienced developers and makers working together

Code-First Developer

1.  The code-first developer starts building the service
2.  Interact with legacy data
3.  Builds API and publishes as an Azure function to API Management
4.  Exports as API Management Connector, a first-class component to Power Platform
5.  Handover to a citizen dev

Maker

1.  Citizen developers build UX
2.  Citizen developers work together within the same app
3.  Collaboration leads to progress on plugging in the API Management Connector
4.  Citizen developer gets help from a code-first developer
5.  Uses open-source tools to add logic
6.  Release to app stores for iOS and Android

The fusion team maturity model

The 5 phases of full realization of benefits for fusion development

Initiation

Plan and launch a fusion program

- Fusion opportunity identification
- CIO-led charter and approach to structuring
- Application assessment
- Champions of process identified
- Cross IT executive buy-in

Key driver:

CIOs

Preparation

Set up governance and security strategy

- Creation of controlled environments and proper governance
- Formal COE established
- Establishment of roles and responsibilities
- Skilling distribution across fusion team

Key driver:

IT pros (Sec, DevOps, etc.)

Collaboration

Collaborate on app development

- Initial application built
- Fusion team focus on areas of expertise
- Achieve rapid deployment and real-time iterations

Key driver:

Pro devs, Business Users/Makers

Acceleration

Maintain quality with DevOps and best practices

- Best practices are curated and scaled
- Showcase fusion-built apps
- Field ideas for new apps
- Apply standard DevOps processes to fusion apps

Key driver:

CIOs, IT pros

Innovation

Enjoy the benefits of fusion development

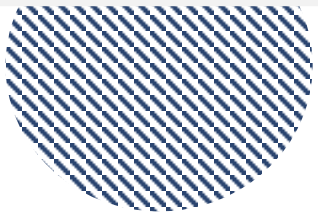
- Fusion applications flourish around the entire enterprise, empowering teams to ideate and execute
- Pro devs and IT pros have confidently secured environments for creation of mission-critical apps
- Competitive advantage realized

Key driver:

Pro devs, IT pros

ALM basics with Power Platform

Environments



Power Platform environments

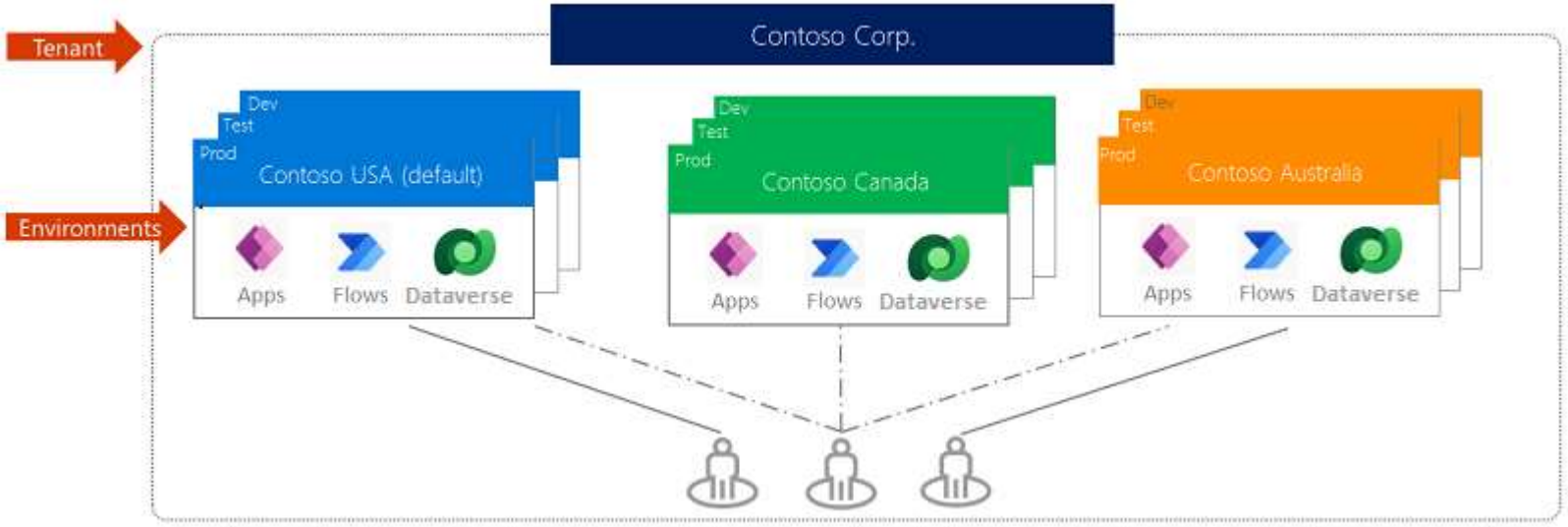
Environment overview

A space to **store**, **manage**, and **share** your organization's business data, apps, chatbots, and flows.

It also serves as a **container** to separate apps that might have **different roles**, **security** requirements, or **target** audiences.

How you choose to use environments depends on **your organization** and the apps you're trying to build.

You can also **move resources** between environments through **solutions**.



Learn more here.

Power Platform environments

Environment strategy

To follow **ALM principles**, are required dedicated environments for at least **development**, **testing** and **production**

Some organizations might also need **more environments**, for example for UAT and SIT testing

Consider separate development environments to **help isolate changes** from one work effort being checked in before it's complete

Power Platform updates are **applied by regions**, so additional consideration must be taken in **global organizations**

The level of **complexity**, how **critical** the app is, and **users impacted** by the application are all important measures of how to **provision environments** to support all the scenarios.

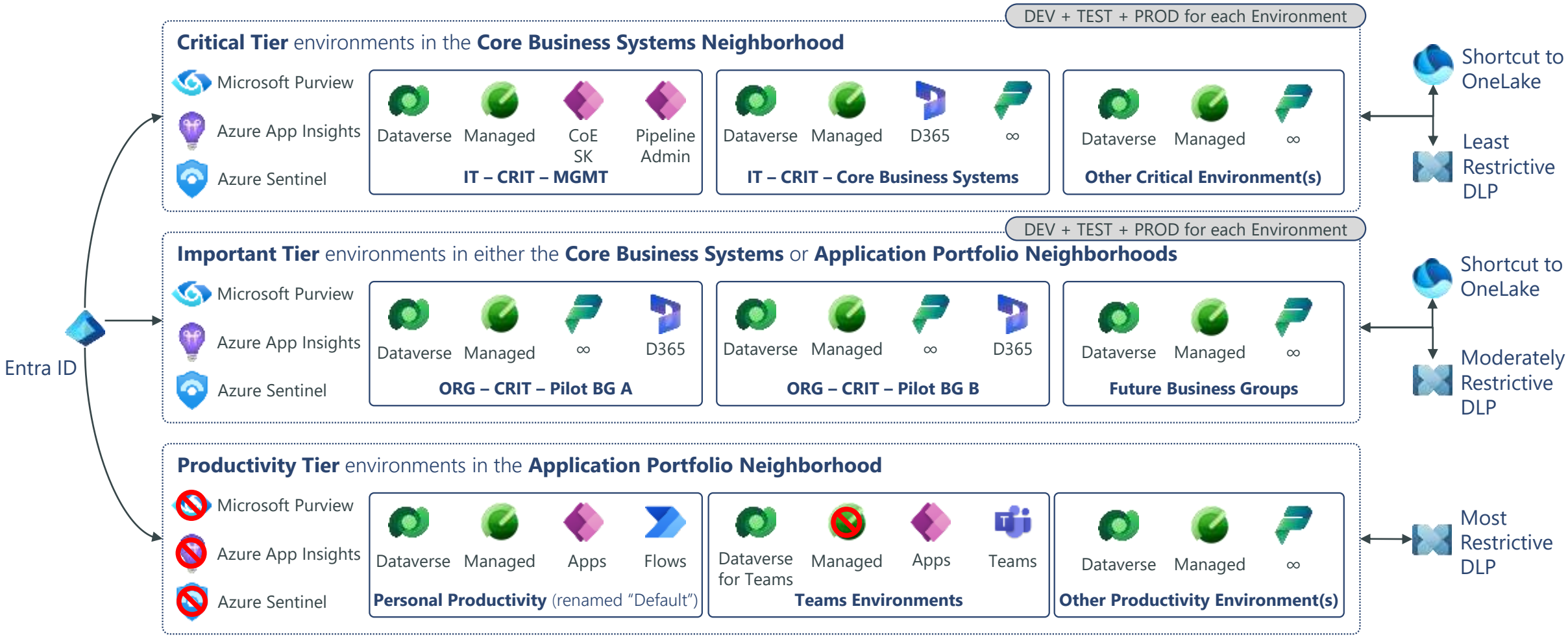
The environment strategy will **shape** and **direct** the **DLP strategy**

Each environment consumes **1 GB of data capacity**, so manage custom environments wisely.

Learn more here.

Power Platform environments

Environment strategy



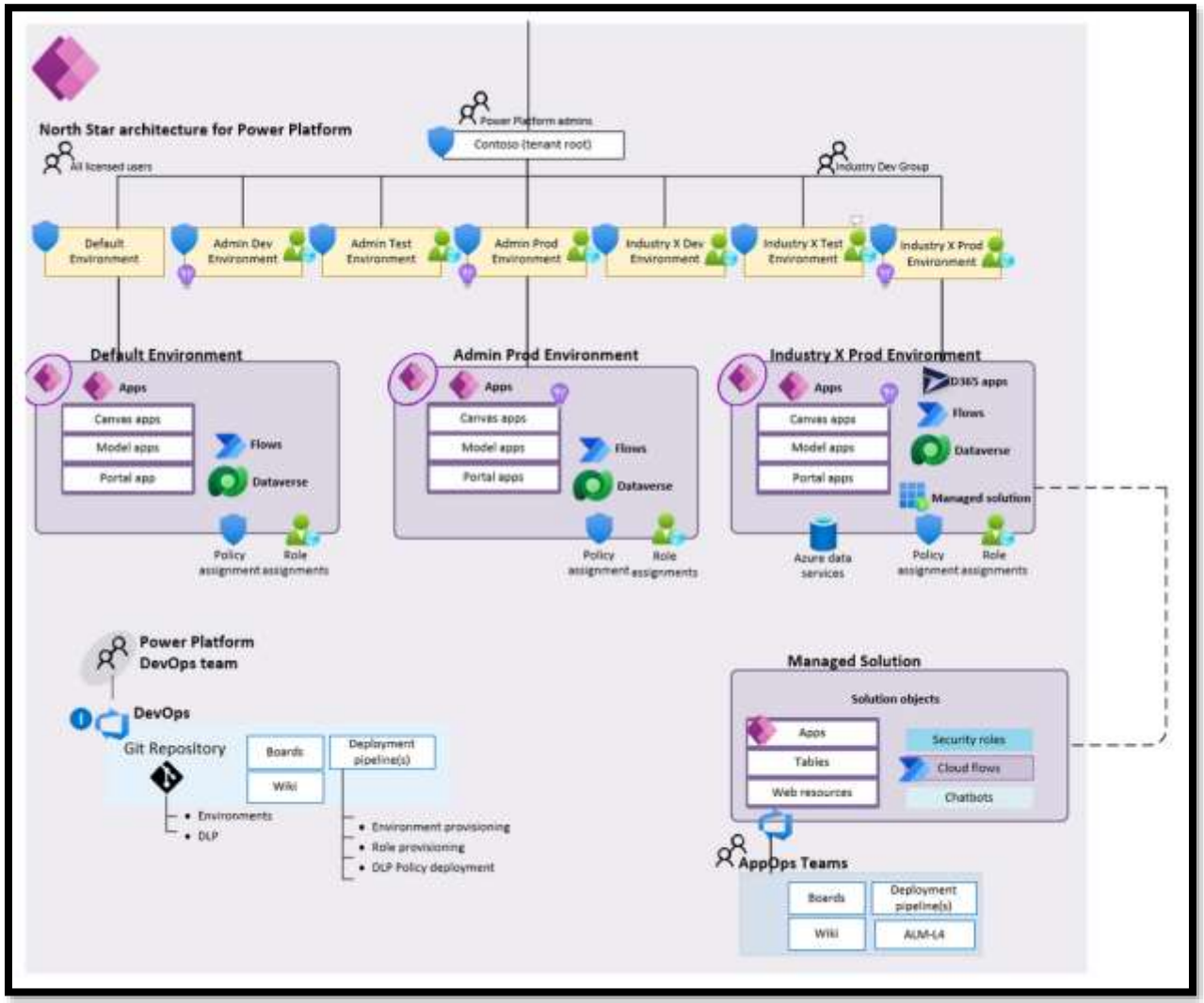
Data integration capabilities between Power Platform and Azure (or third party) integration services such as Azure Data Factory, Event Grid, Logic and Function apps, and a Dataverse Data Landing Zone are intentionally omitted from this diagram for simplicity. They should be included in a more detailed architecture if intended for deployment alongside the Power Platform Landing Zone.

Power Platform environments

Environment strategy

North Star Landing Zones for Power Platform

A reference implementation is an optimized, and roadmap aligned implementation that enables organizations to create **landing zones at scale** into their existing Power Platform tenant, ideally instantiated using the North Star Architecture implementation to ensure the landing zones are secure and well-governed.



https://github.com/microsoft/industry/tree/main/foundations/powerPlatform

Phil Topness, Kristian Nese, and Ken Auguillard talks about Nort Star

Power Platform environments

Features that support an enterprise-scale, environment strategy

Types of environments

Describes the different uses of environments as part of your strategy.

Managed Environments

Provides a set of premium capabilities that make environments easier to manage at scale.

License autoclaim

Simplifies license assignment by allowing users to claim Power Apps per user licenses when they're needed, instead of requiring an admin to identify users who need licenses in advance.

Environment groups and rules

Explains how to manage environments as groups and apply rules to groups to automate consistent governance policies.

Default environment routing

Automatically moves makers away from creating resources in the default environment to their own, personal environment.

Microsoft Dataverse

Provides enhanced security and ALM.

Preferred solutions

Helps makers ensure that all the assets they build are in a Dataverse solution, making it easier to promote them to other environments.

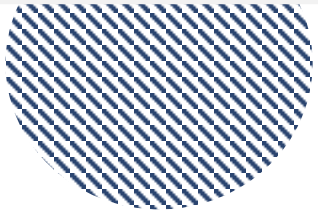
Pipelines in Power Platform

Provides a simplified process for promoting assets from development to test and production environments, making continuous integration and deployment (CI/CD) available to all makers.

Catalog in Power Platform

Allows makers to share components, like apps and flows, and more advanced starting points, such as templates.

Solutions

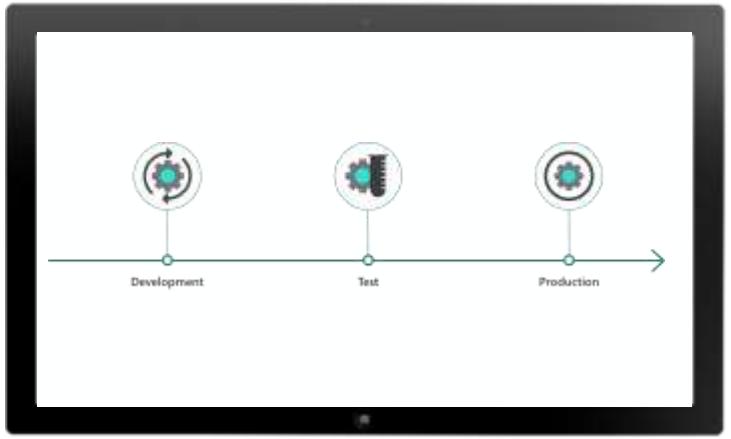


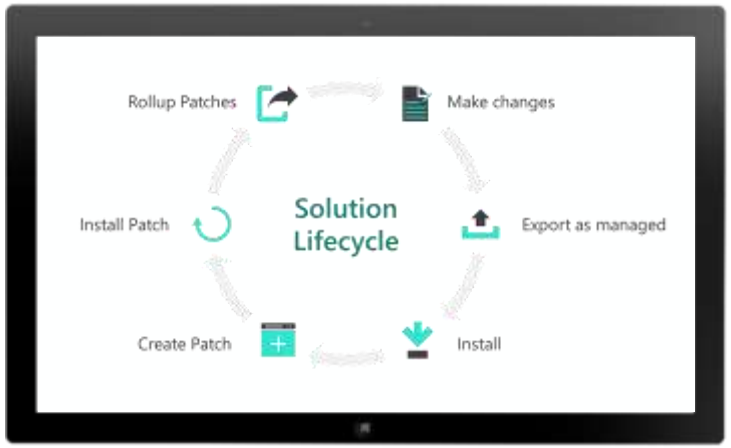
ALM basics with Power Platform

Microsoft Power Platform

Solutions

Solutions enables aplication liflecycle management in Power Platform





Author solutions	Deploy solutions	Manage solution lifecycle
Solutions are containers to track the changes and customizations you make	Solutions are how you transport and install changes to target environments	Solutions enable lifecycle management for your customizations and can be fully integrated into your DevOps processes

https://docs.microsoft.com/en-us/power-platform/alm/solution-concepts-alm

ALM basics with Power Platform

Microsoft Power Platform

Solutions types

Unmanaged

- Used in **development environments** while you make changes to your application.
- Can be **exported** either as **unmanaged** or **managed**.
- **Exported unmanaged** versions of solutions should be **checked** into **source control system**.
- Unmanaged solution, not environments, must be considered the **source of true** for Microsoft Power Platform assets
- When **deleted**, its components remains and belongs to **Default Solution**.

Managed

- Used in all any environment **that isn't a development** environment for that solution.
- Must be considered a **build artifact**.
- **You can't edit components** directly within a managed solution. Buy you can define some **properties** that can be changed at destination environment
- **Altering** a **managed component** at destination environment will create a **dependent unmanaged solution**.
- When a managed solution is **deleted** (uninstalled), **all the customizations** and **extensions** included with it are **removed**.

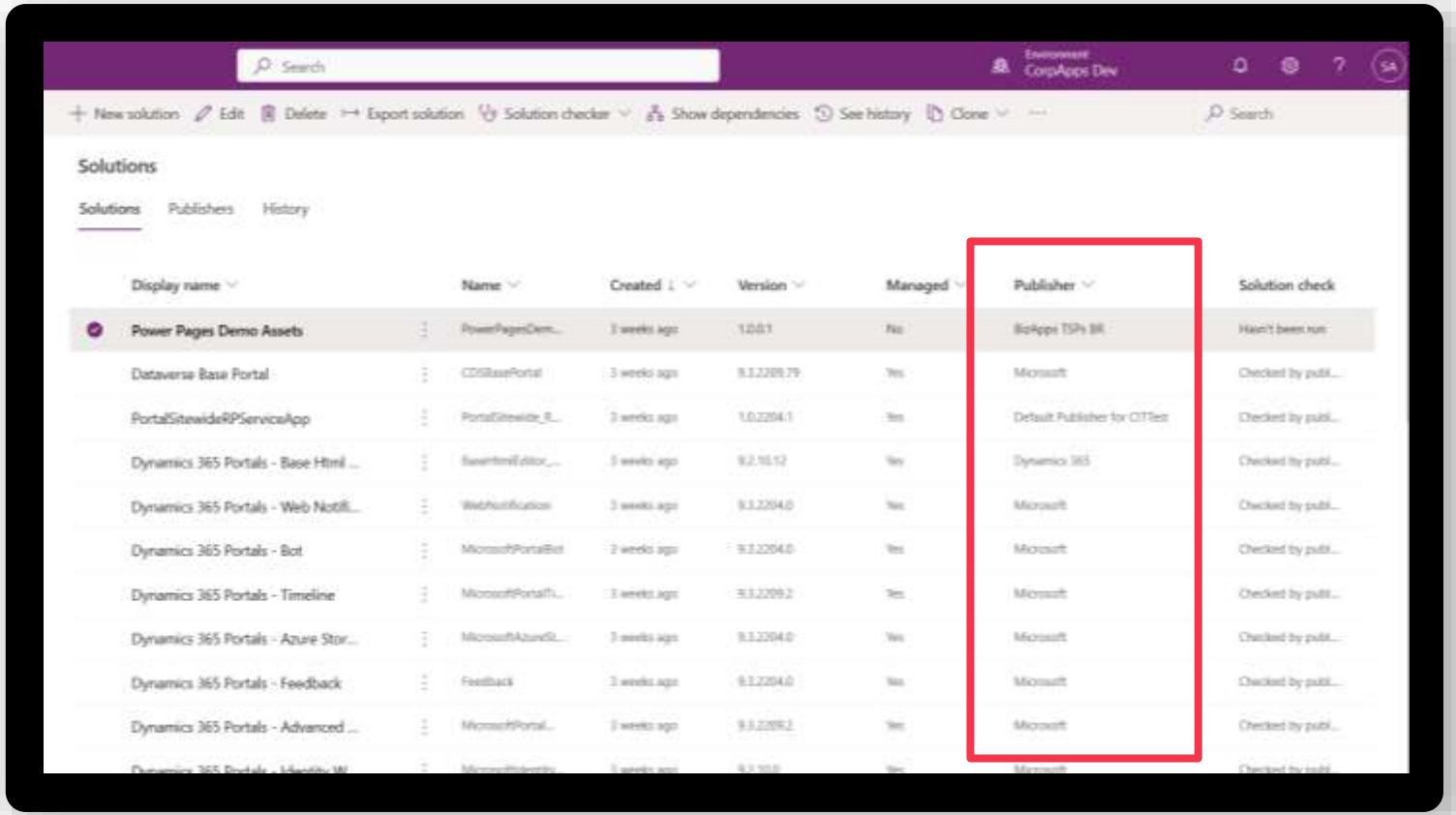
https://docs.microsoft.com/en-us/power-platform/alm/solution-concepts-alm
https://learn.microsoft.com/en-us/power-platform/alm/managed-properties-alm

ALM basics with Power Platform

Microsoft Power Platform

Solution publisher

- Solutions **require** a **publisher**
- Publisher can be associated with multiple solutions
- **Two default publishers** are included
  - CDS Default Publisher
  - Default Publisher for “your org name”
- **Create your own publisher** - do not use one of the default publishers
- Use the **same publisher in all your solutions**
- The publisher dictates the **customizations prefix and choice values**
- **Important** - When importing new assets through a managed solution, the **publisher “owns” those assets**



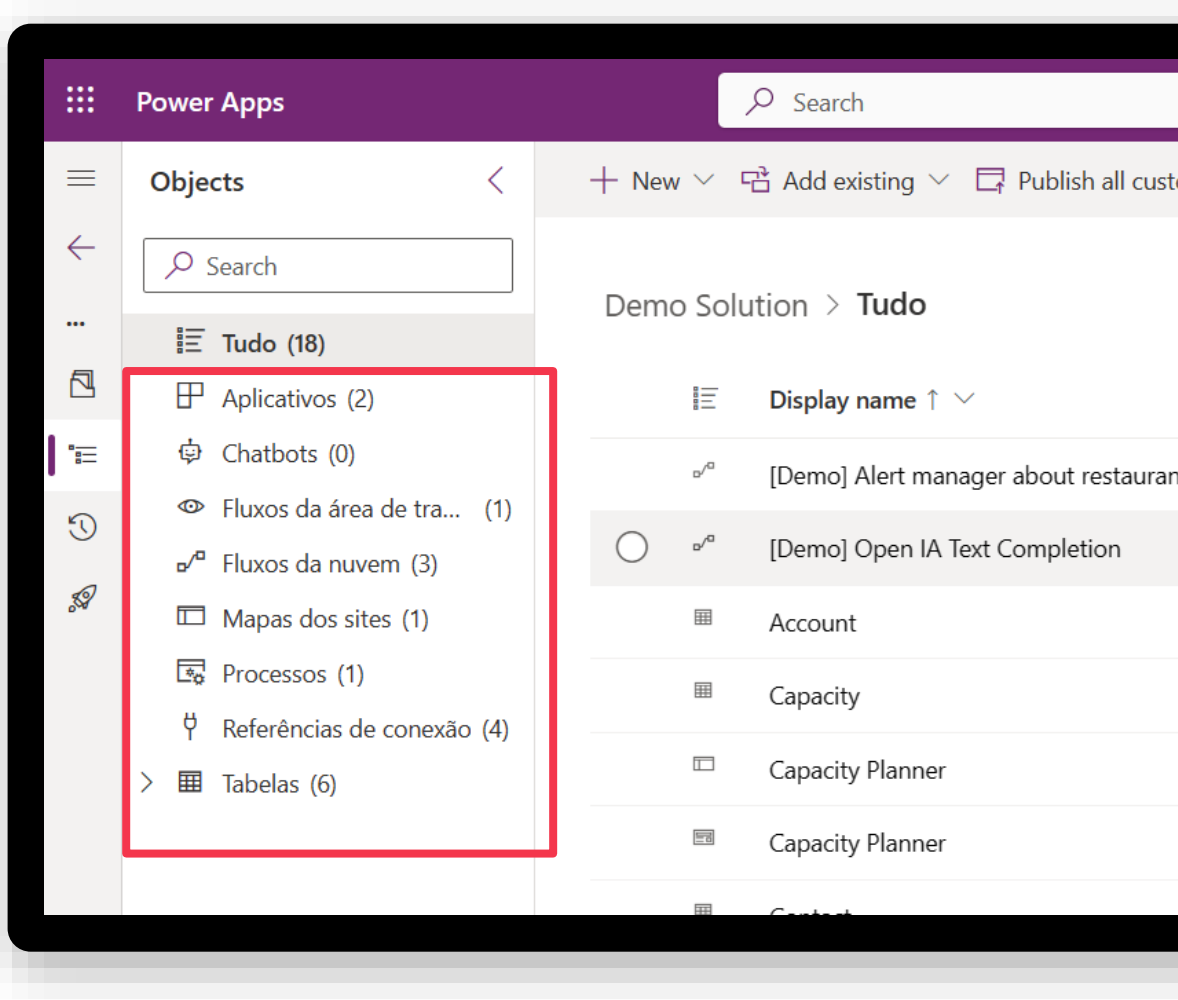
https://docs.microsoft.com/en-us/power-platform/alm/solution-concepts-alm

ALM basics with Power Platform

Microsoft Power Platform

Solution components

A component represents something that you **can potentially customize**. Anything that can be included in a solution is a component. To view the components included in a solution, open the solution you want. The components are listed in the **Components** list.



https://docs.microsoft.com/en-us/power-platform/alm/solution-concepts-alm

ALM basics with Power Platform

Microsoft Power Platform

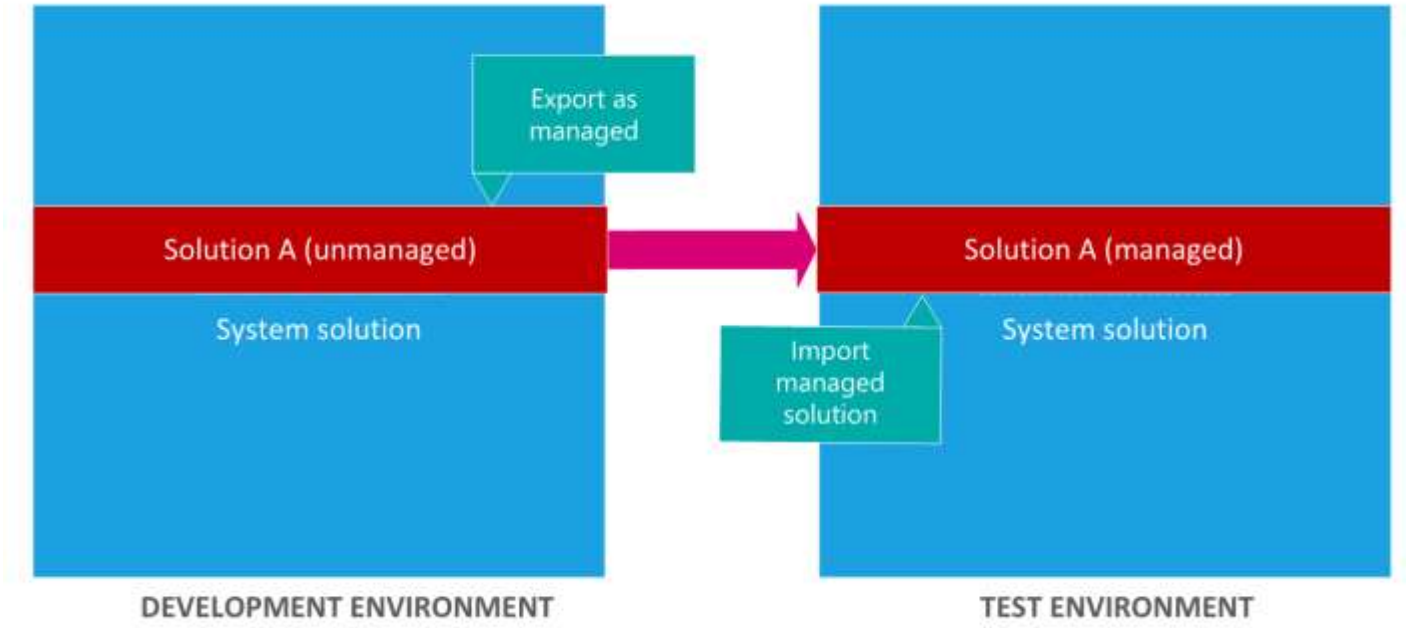
Solution lifecycle

Create Author and export unmanaged solutions.

Update: Create updates to a managed solution that are deployed to the parent managed solution. You **can't delete components** with an update.

Patch*: A patch contains **only the changes for a parent managed solution**. Use patches when making **small updates** (similar to a hotfix).

Upgrade: removes unused components, implements upgrade logic and merges all patches to the solution into a new version of the solution. Solution upgrades will **delete components** that existed but are **no longer included in the upgraded version**.



Solution lifecycle

* Patching isn't recommended because it limits team development and increases complexity when storing your solution in a source control. Use **Update** instead.

https://docs.microsoft.com/en-us/power-platform/alm/solution-concepts-alm
https://learn.microsoft.com/en-us/power-platform/alm/update-solutions-alm

ALM basics with Power Platform

Microsoft Power Platform

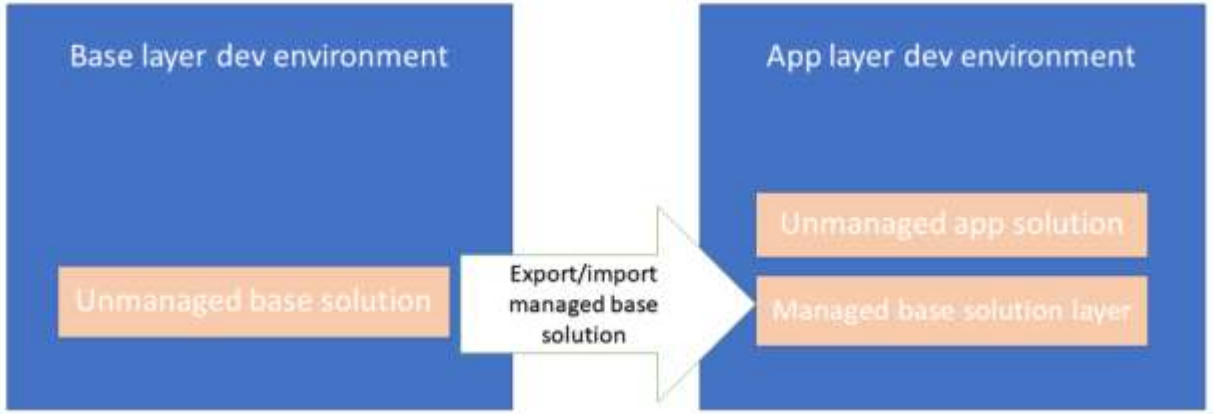
Organize solutions

Take some time do plan!

- Think about how many solutions you **want to release** and whether the solutions will **share components**.
- How it will be the **size** of your solution?
- How your solution will be **delivered** and **updated**?
- What will be the **purpose** of your solution? (a template/base solution, departmental, mission critic)

Tips

- For most of the cases, **one solution** will be the **best option**.
- However, use **multiple solution** could be **beneficial**. For example, when you have a large application and need to **segment development** and **deployment**.
- Segment your solutions by component type when there are no **cross-dependency risks**.
- Solution dependency can be **hard** to solve.
- Avoid develop multiple apps with **different purposes** on the same environment because this can lead to **solution dependency** between them.



A recommended solution segmentation approach

https://learn.microsoft.com/en-us/power-platform/alm/organize-solutions
https://learn.microsoft.com/en-us/power-platform/alm/removing-dependencies

ALM basics with Power Platform

Microsoft Power Platform

Solution layers

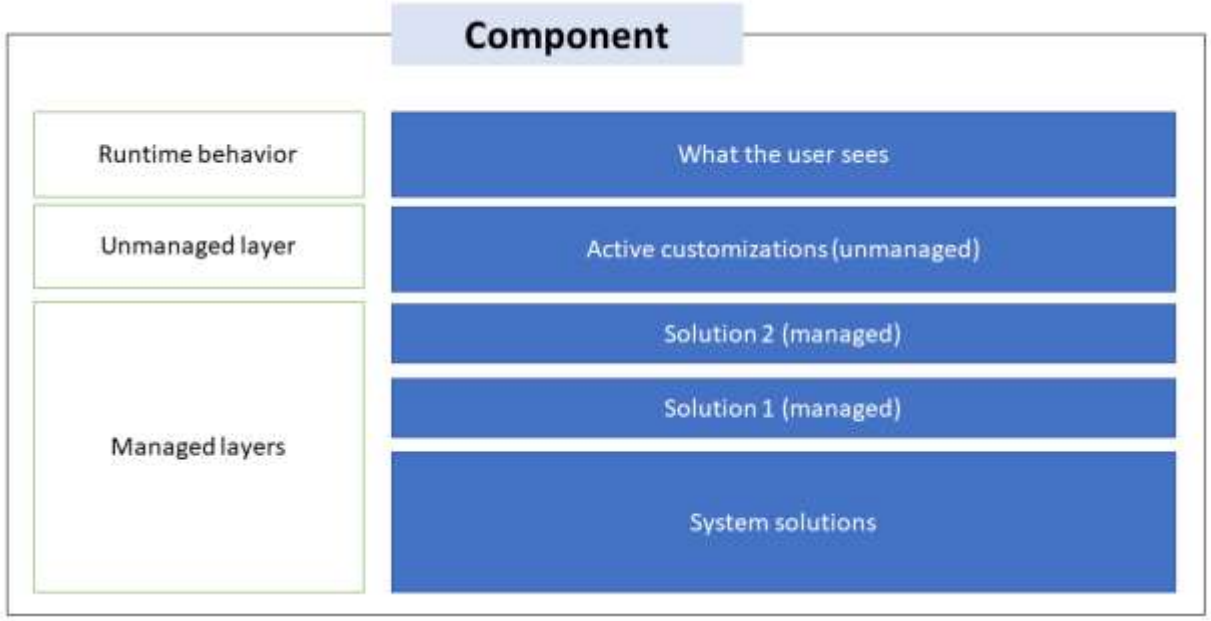
When you have **multiple solutions installed**, there will be in layers. There are **two** distinct layers:

- **Unmanaged**: all imported unmanaged solutions and ad-hoc customizations
- **Managed**: All imported, managed solutions and the system solution

Key facts about layers

- Solution layering is implemented at a **component level**.
- When multiple managed solutions are installed, the **last one installed** is above the managed solution installed previously.
- If you **uninstall** a managed solution, the managed solution **below** it takes effect.
- If you **uninstall all managed solutions**, the default behavior defined within the **system solution** is applied
- The are two possible runtime behaviors for a component:

  Model Driven Apps, forms and site map **will be merged**
  All other components will be that on **top layer**



Solution layers

Avoid put components on multiple solutions

https://learn.microsoft.com/en-us/power-platform/alm/solution-layers-alm
https://learn.microsoft.com/en-us/power-platform/alm/how-managed-solutions-merged

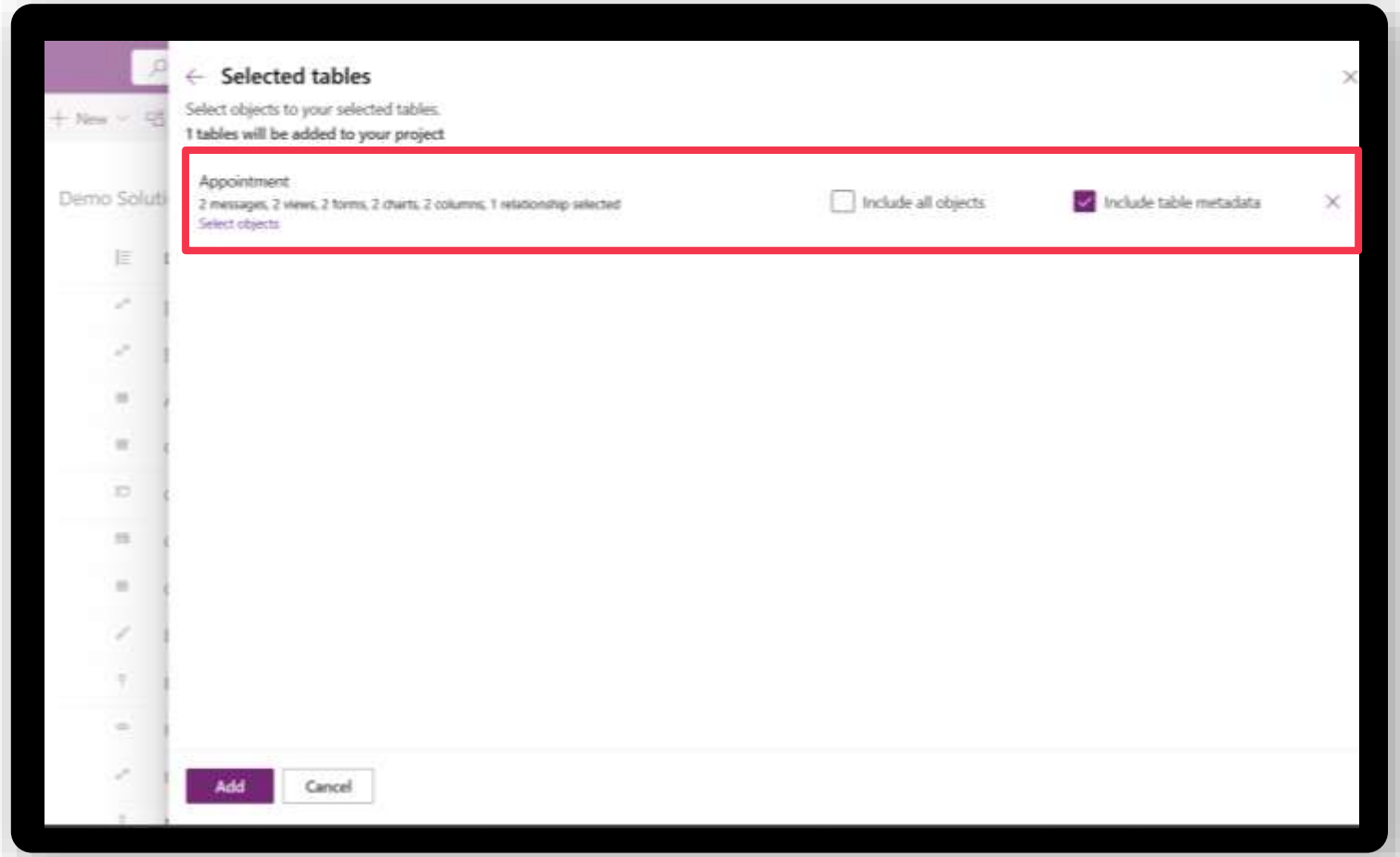
ALM basics with Power Platform

Microsoft Power Platform

Use segmented solutions

- Only include in a solution tables components that you **created** or **modified**.
- A solution with **unintended components** can cause **unexpected behavior** at destination environment, since it will create a **new solution layer** for them if them already exists.
- The following options are available when including a table component in a solution:

Include no components or metadata
Select components
Include entity metadata
Include all components



If the component exists in target environment and you haven't change it, don't include it in the solution

https://learn.microsoft.com/en-us/power-platform/alm/segmented-solutions-alm

Demonstration

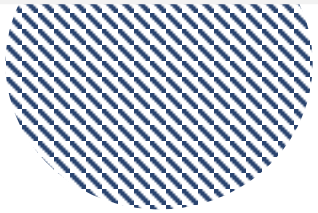
- Using solutions to transfer components between environments (6:36 min)

Microsoft Power Platform

How to automate ALM process

Microsoft Power Automate

Source control and team development



ALM basics with Power Platform

Microsoft Power Platform

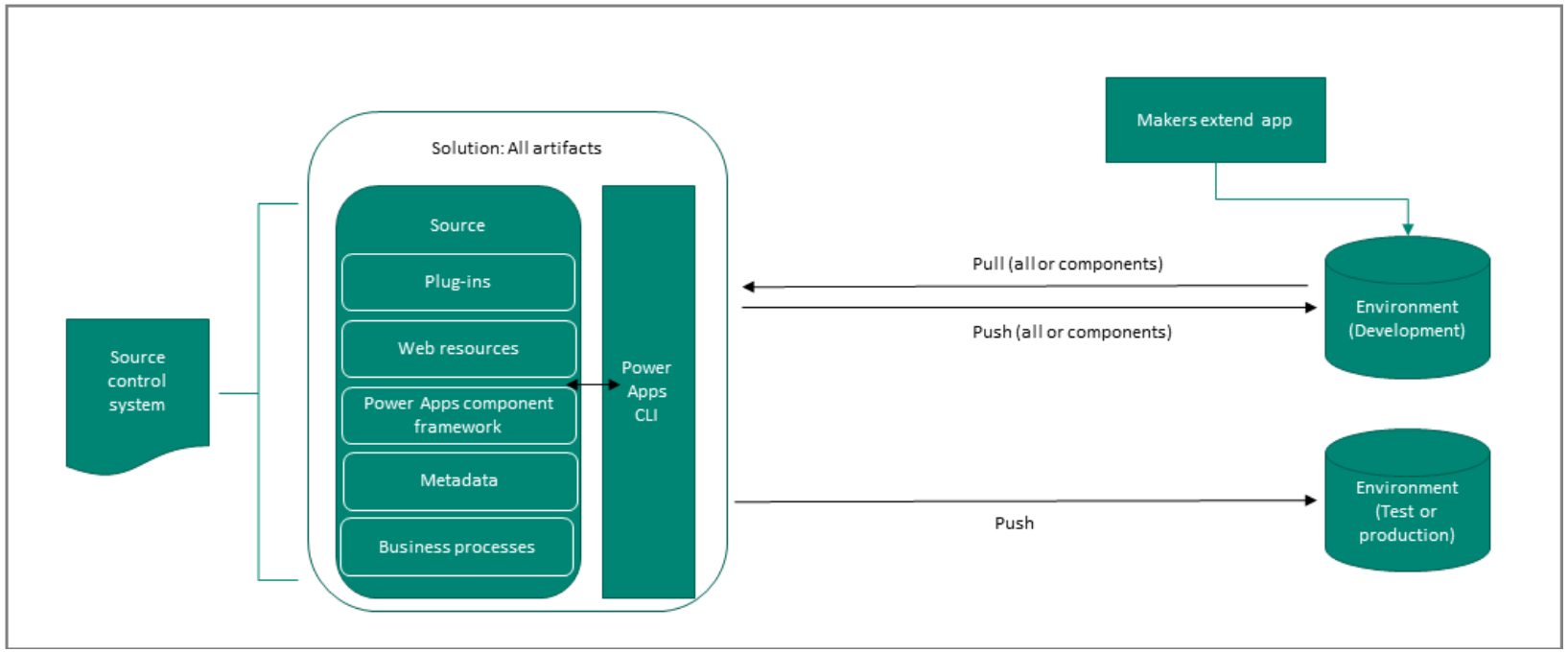
Source Control

Use when **multiple app makers** and **developers** are working on the **same set of files**.

When used, **must be the single source of truth**, not environments

Gives you the ability to **roll back changes** or restore deleted files

Use a **Git repository** and adopt a **branch strategy** to enable team development and **Azure DevOps** or **GitHub** to start your **DevOps journey**



https://learn.microsoft.com/en-us/power-platform/alm/basics-alm#source-control

ALM basics with Power Platform

Microsoft Power Platform

Source Control

Team development is **always challenging**, so keep in mind:

- **Automation is a journey**. The focus must me on foster communication and gradually improve efficiency.

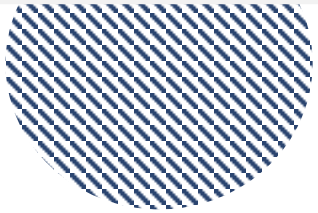
- Because source control systems **have limitations** on how merges occur, we recommend that you avoid situations where multiple people make changes on this complex components at the same time: **forms**, **flows**, and **canvas apps**.

- A **recommended approach** is:
  - Define who in the development team **will be in charge of each type of customization** (Dataverse, apps, flows and custom development)
  - Use **source control** and **build tools** on **Azure DevOps** and **GitHub** to allow pro-devs to get access to the recent versions of the customizations and to integrate and their code automatically.

https://learn.microsoft.com/en-us/power-platform/alm/basics-alm#source-control

Power Platform tools for fusion development

For admins and makers

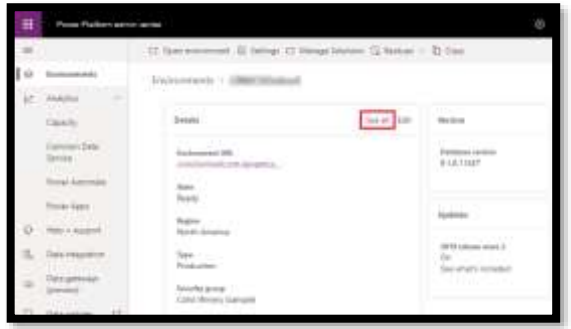


Tools for admins and makers

Microsoft Power Platform

Tools used

Power Platform Admin Center



Manage environments, data integration, gateways and data policies

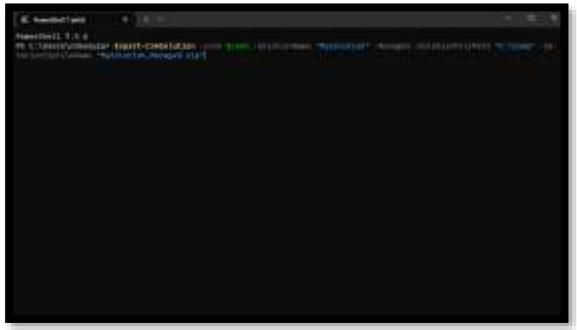
Learn more

Configuration Migration Tool

Move configuration and/or reference data across environments

Learn more

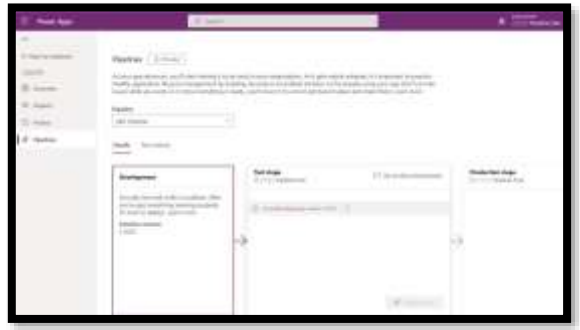
Power Shell Modules



Automate many of the monitoring, management, and quality assurance tasks that are only possible manually

Learn more

Pipelines



Deploy solutions to environments with a single click

Learn more

Apps and Solution Checker

Find and resolve programmatic errors and performance issues faster

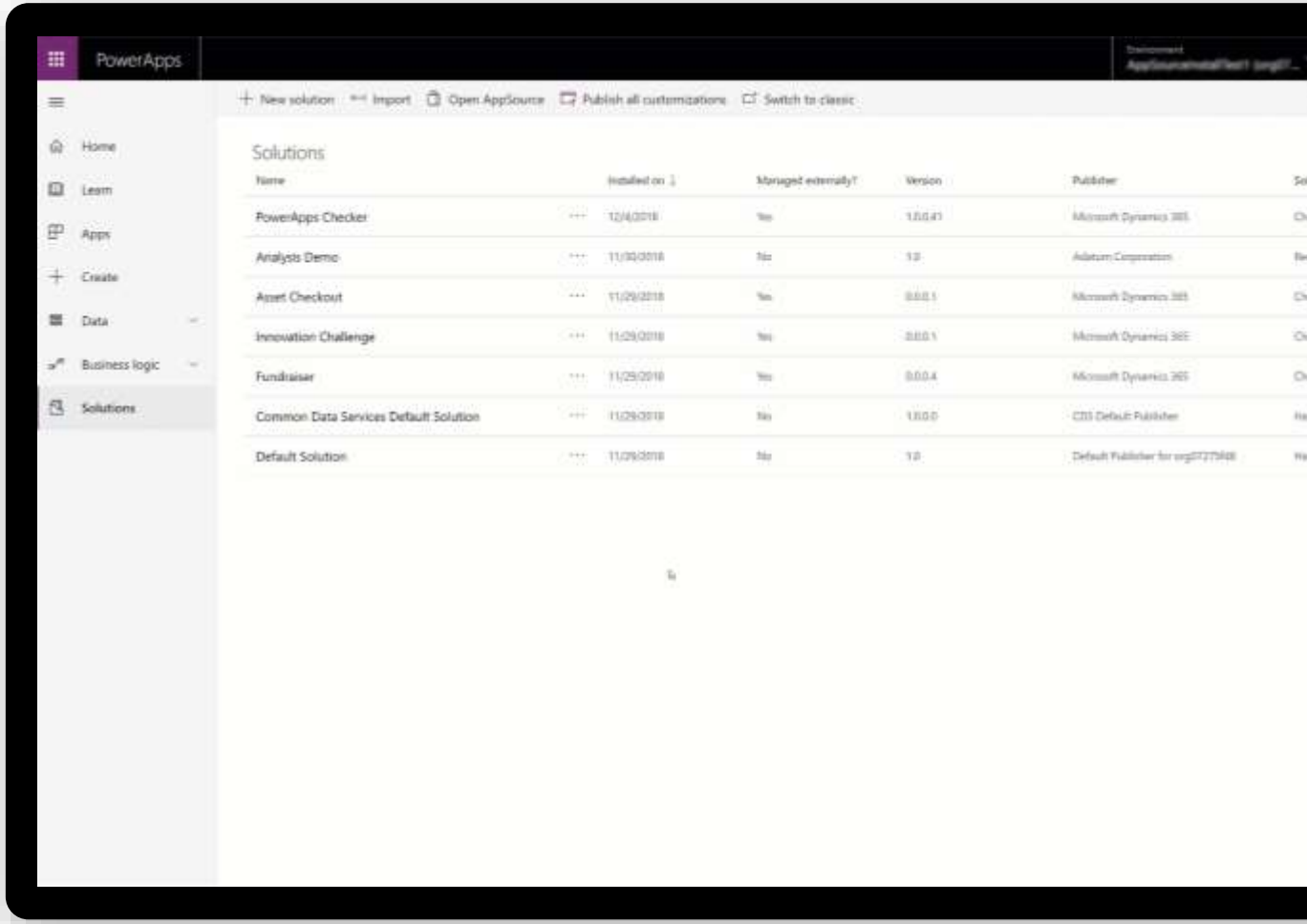
Continued investment to provide guided experience to make debugging easier

Discover accessibility issues and recommendations inline

Performance and stability risks called out by severity and location (component/line)

Identified risks supported by best practice recommendations on learn.microsoft.com

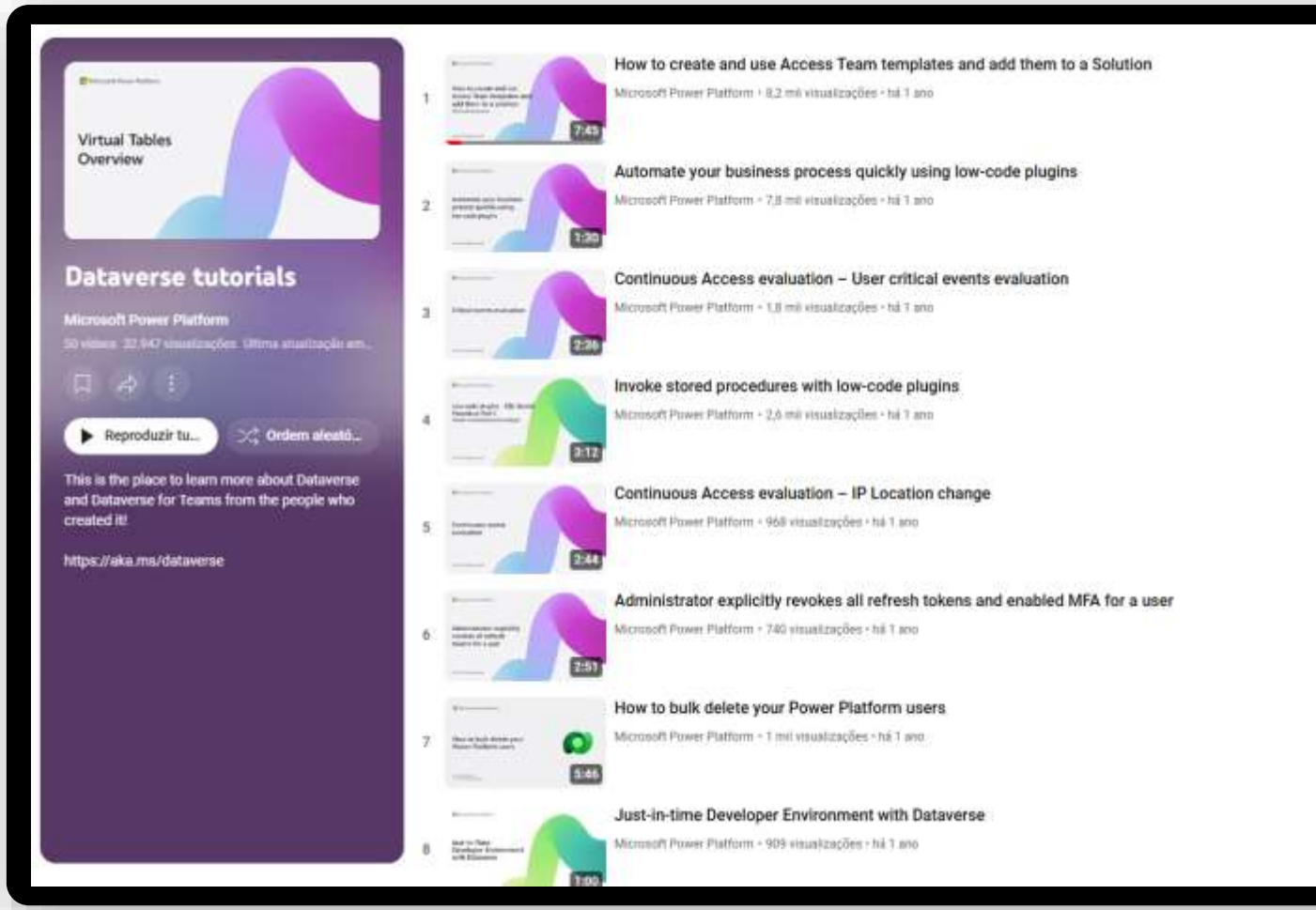
Learn more!



Additional recommended content

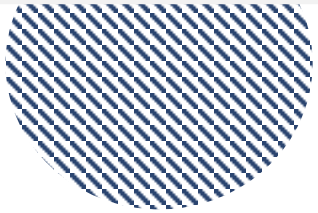
Dataverse tutorials on YouTube

This is the place to learn more about Dataverse from the people who created it!



Learn more!

For developers



Tools for developers

Microsoft Power Platform

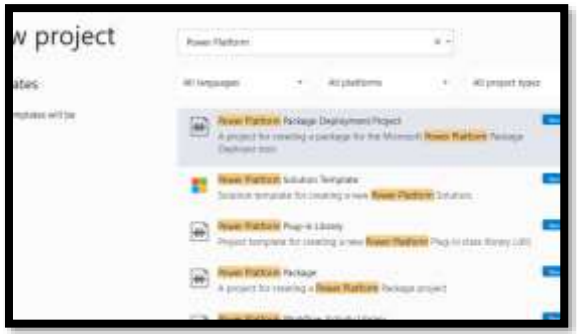
Tools used

Azure DevOps / GitHub

Solution metadata, build and deploy tasks automation

Learn more

Package Deployer



Deploy solutions, files, custom code and HTML files to Dataverse instances

Learn more

Solution Packager

Unpack a compressed solution files so they can be easily managed by a source control system

Learn more

Power Platform CLI

Empowers developers and app makers to create code components

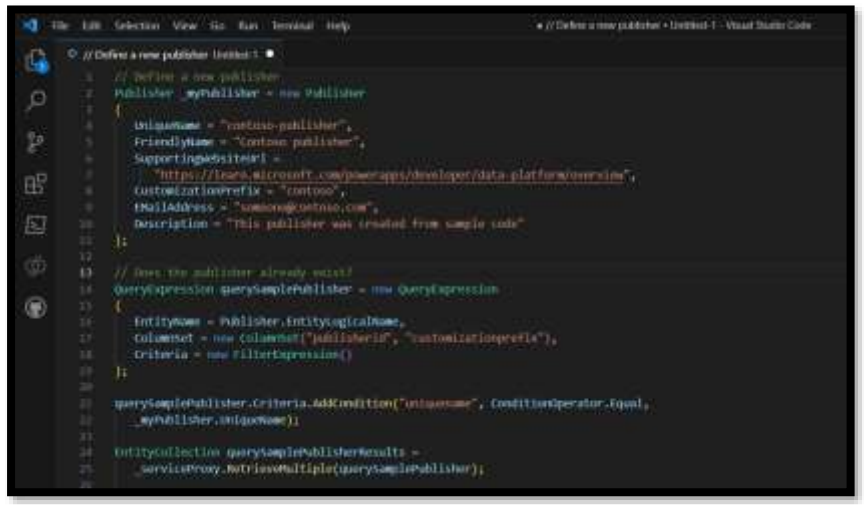
Learn more

Tools for developers

Microsoft Power Platform

Manage solutions using code

Dataverse SDK

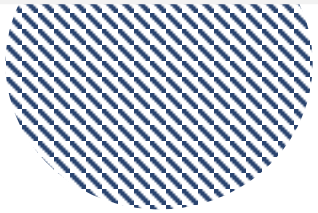


Use Dataverse SDK for .NET to create **custom automation** on your application lifecycle such:

- Create a publisher
- Create, export, import and delete solutions
- Add or remove solution components
- Detect solution dependencies and create a report

Learn more

Collaboration tools



Collaboration tools

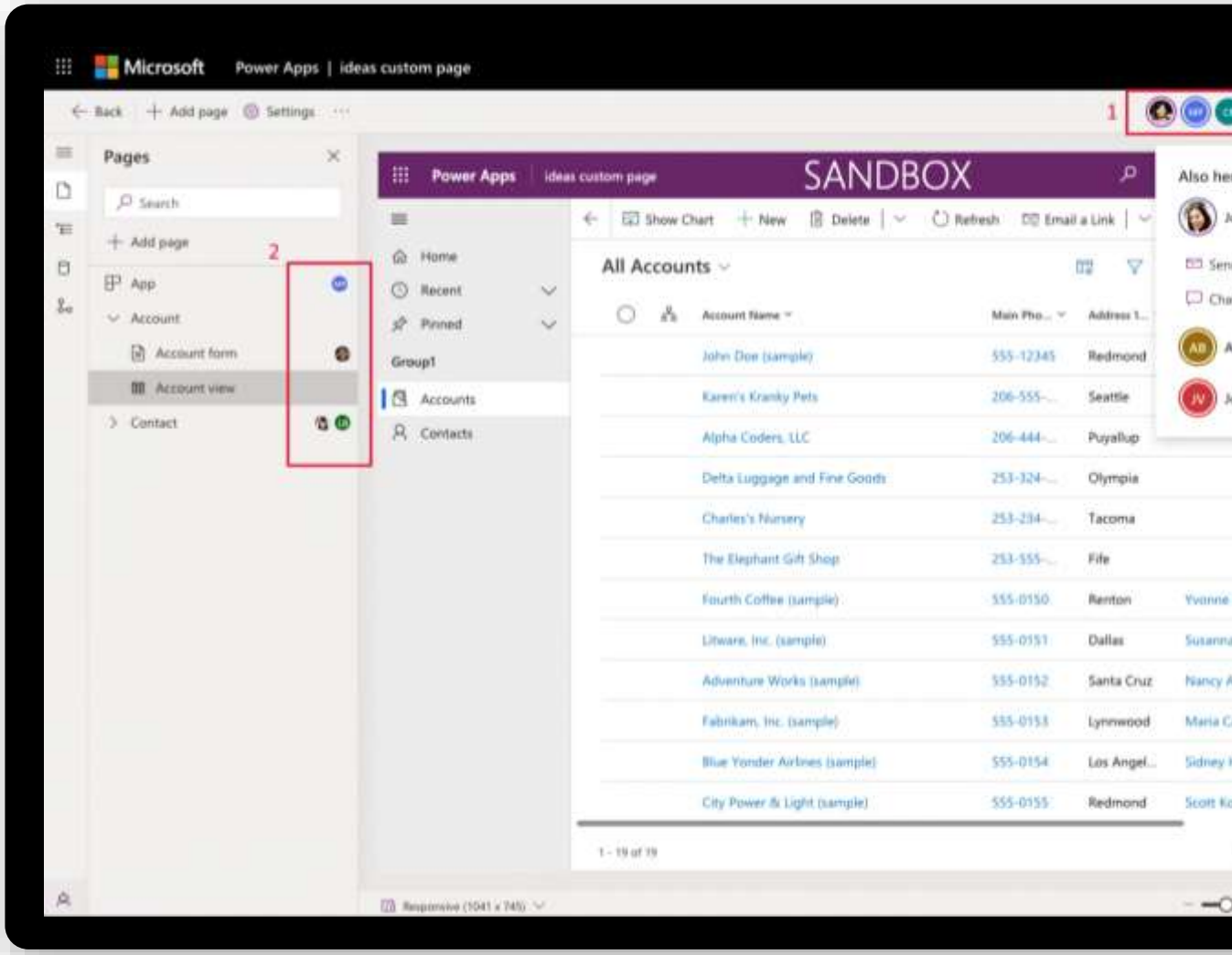
Microsoft Power Platform

Collaboration in model-driven apps

Collaboration is key to help fusion teams work together

In **model-driven apps**, the following collaboration features are available:

- Commenting
- Co-presence (preview)
- Co-authoring (preview)



https://learn.microsoft.com/en-us/power-apps/maker/model-driven-apps/collaboration-model-driven-apps

Collaboration tools

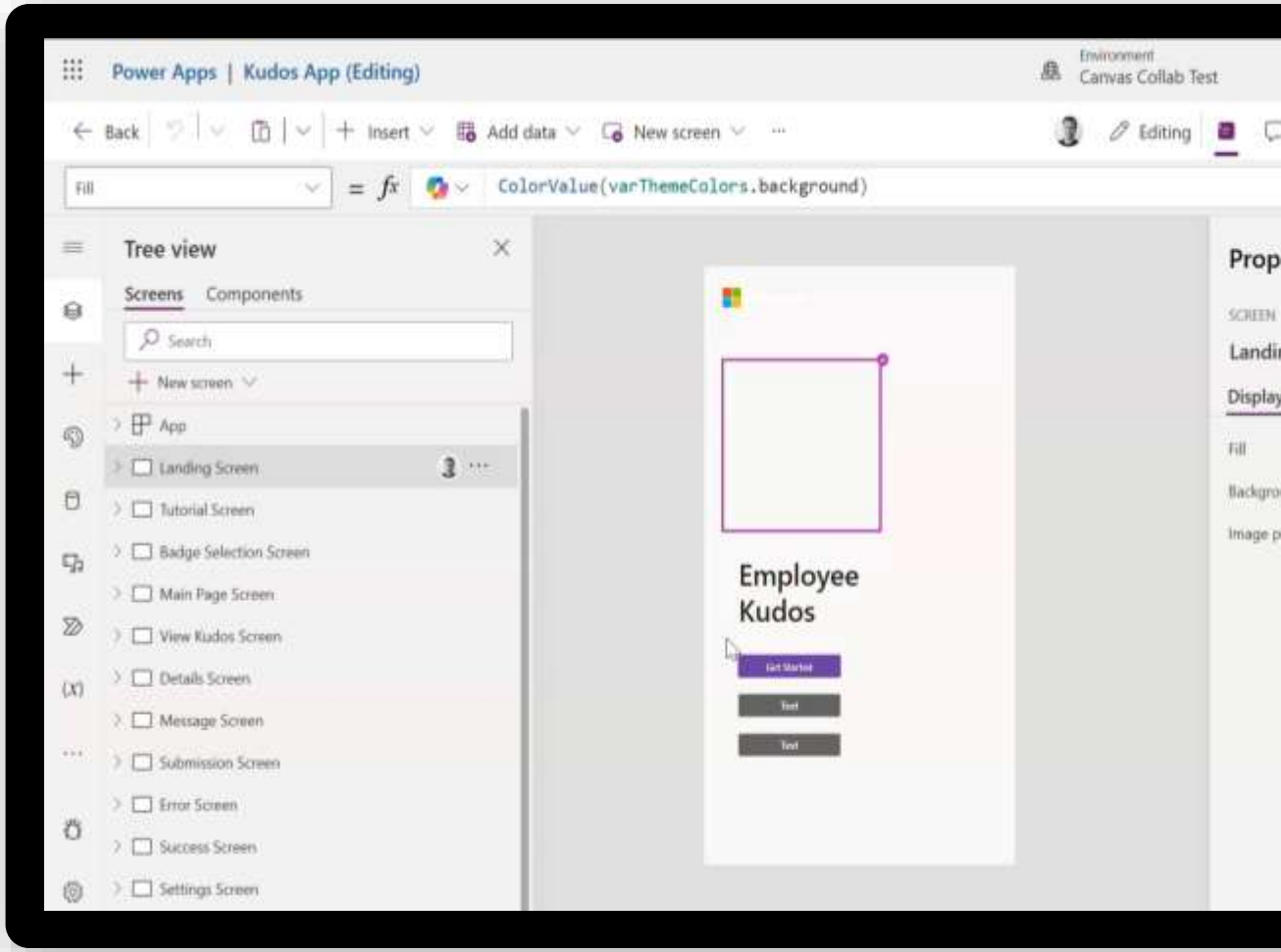
Microsoft Power Platform

Collaboration in canvas apps (preview)

Collaboration is key to help fusion teams work together

In **canvas apps**, the following collaboration features are available:

- Commenting
- Co-presence
- Co-authoring *



** Up to 10 developers*

https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/copresence-power-apps-studio#coauthoring-preview

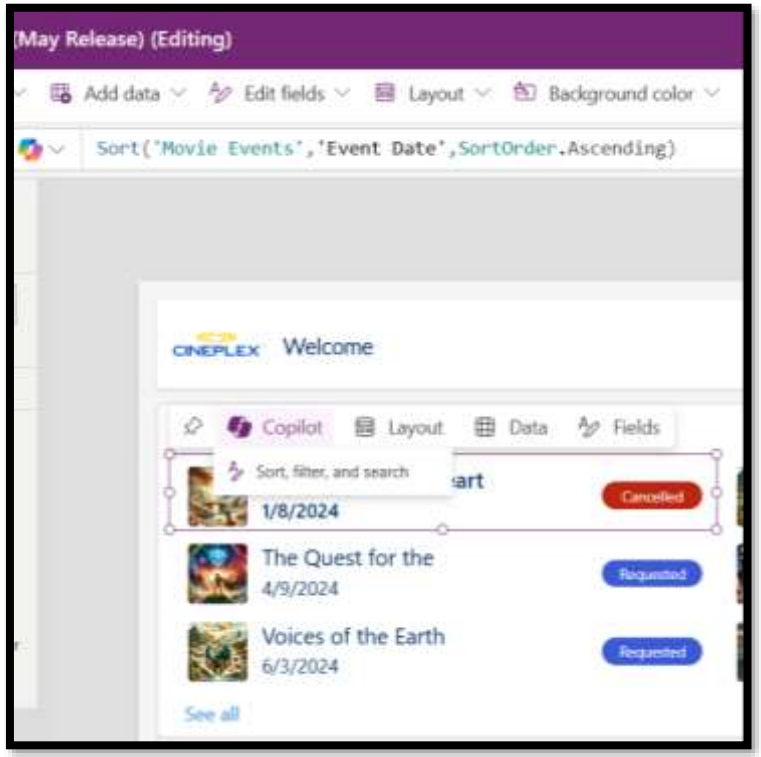
Collaboration tools

Microsoft Power Platform

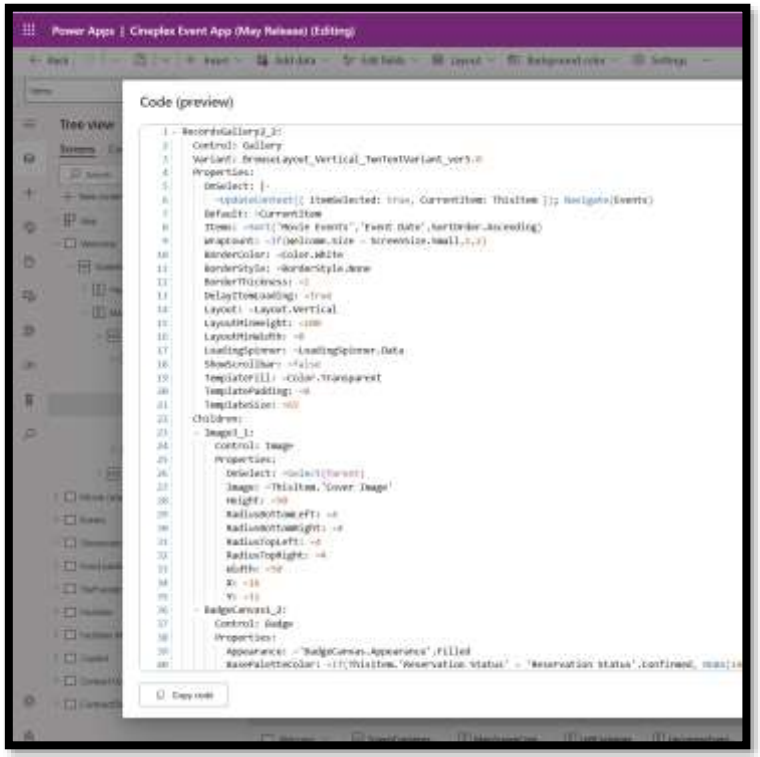
Release date: Jan 2025

Manage your source code for canvas apps (preview)

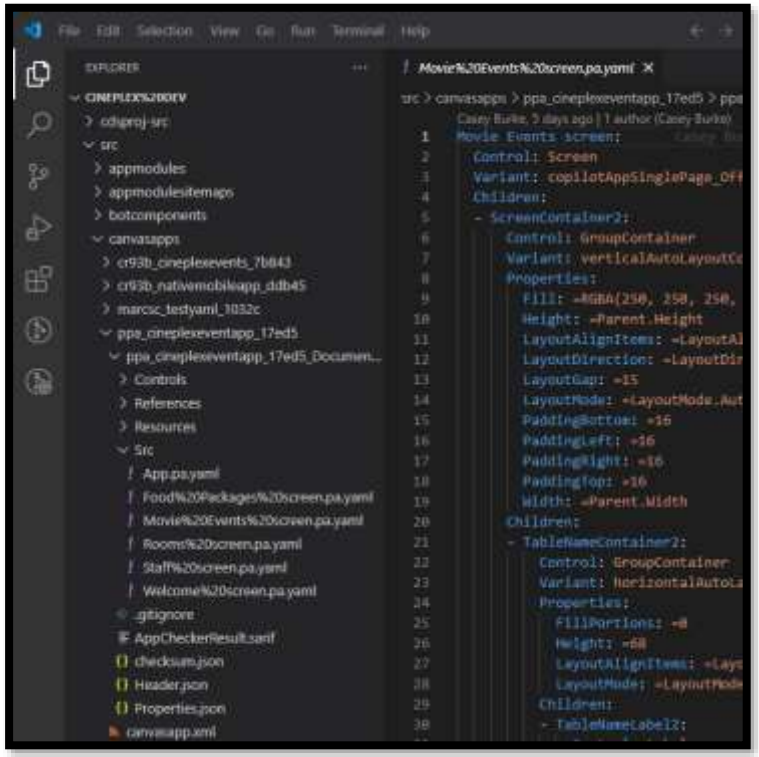
Less code ⟷ Pro-code



In the web designer with natural language, drag and drop



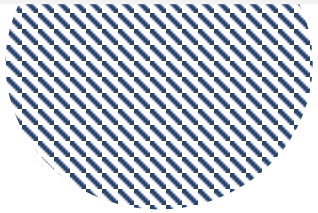
In the web designer with code view



In your preferred IDE

https://powerapps.microsoft.com/en-us/blog/power-apps-is-making-it-easier-for-developers-to-build-with-microsoft-copilot-and-each-other/

Testing and telemetry tools

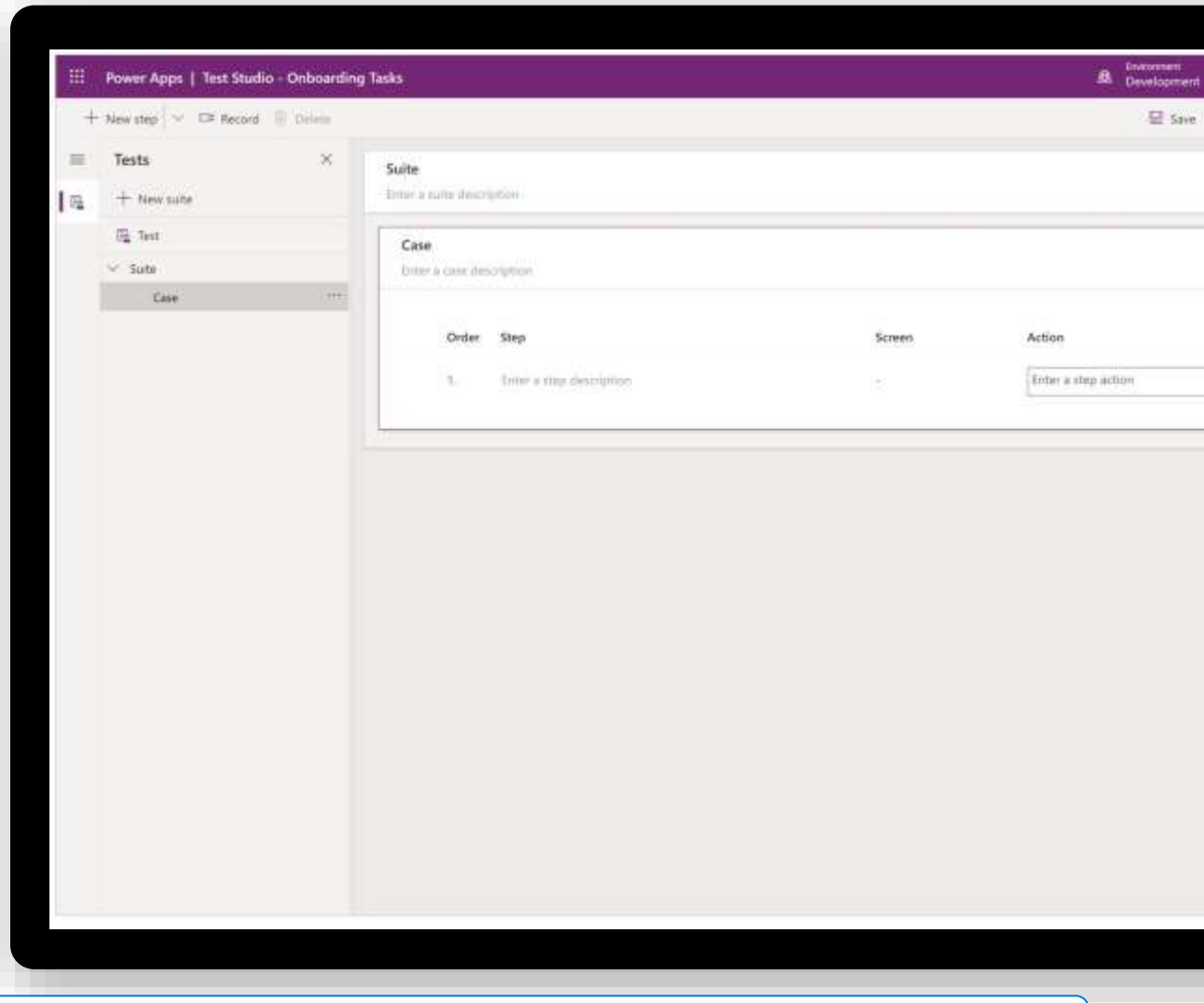


Testing tools

Microsoft Power Platform

Test Studio (only for Canvas app)

- **Low-code solution** to write, organize, and automate tests **for canvas apps**
- Write tests using **Power Apps expressions** or use a **recorder** to save app interaction to automatically generate the expressions
- Can be used **on Azure DevOps pipelines** to automate test execution



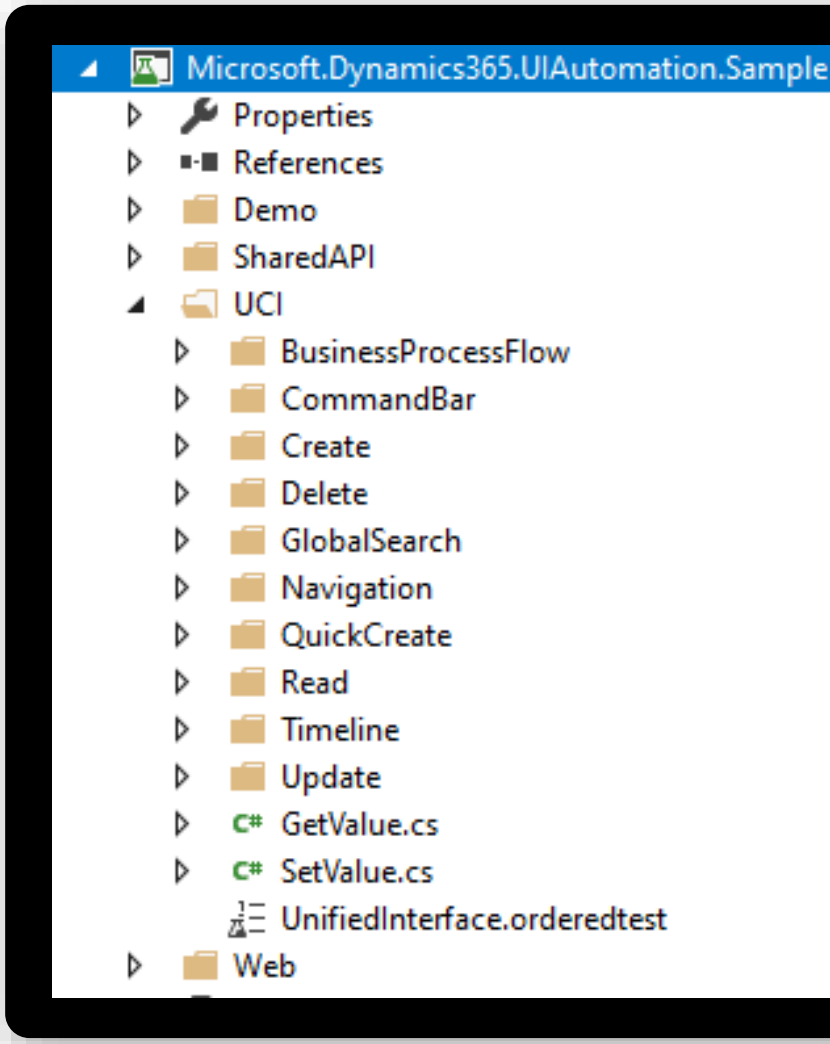
https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/test-studio

Testing and telemetry tools

Microsoft Power Platform

Easy Repro (model-driven apps)

- **An automated UI testing framework**
- Built specifically for testing your **Dynamics 365** and **model-driven apps** implementations
- Provides an **easy-to-use set of APIs** that make setting up UI testing quick and easy, abstracting and reducing the need for you to work directly with the Document Object Model (DOM)
- Can be included on **Azure DevOps pipelines**



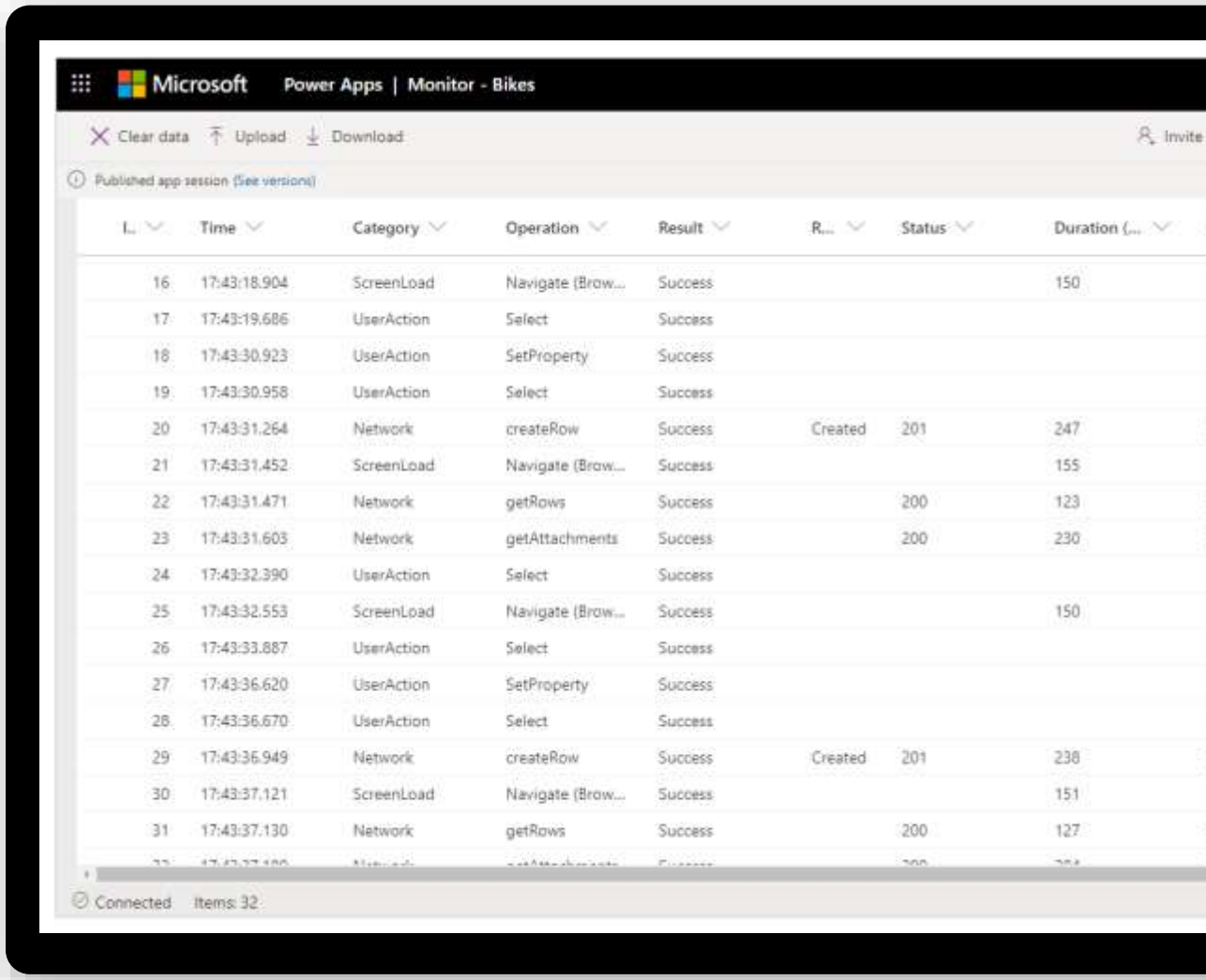
https://github.com/microsoft/EasyRepro

Testing and telemetry tools

Microsoft Power Platform

Monitor

- View a **stream of events** from a user's session to **diagnose** and **troubleshoot** problems
- Works for **canvas and model-driven apps**
- Help you **diagnose** and **troubleshoot** problems faster
- Provides a **better understanding** of how the events and formulas contained in your app work, so you can **improve performance** and identify any **errors** or **problems**.



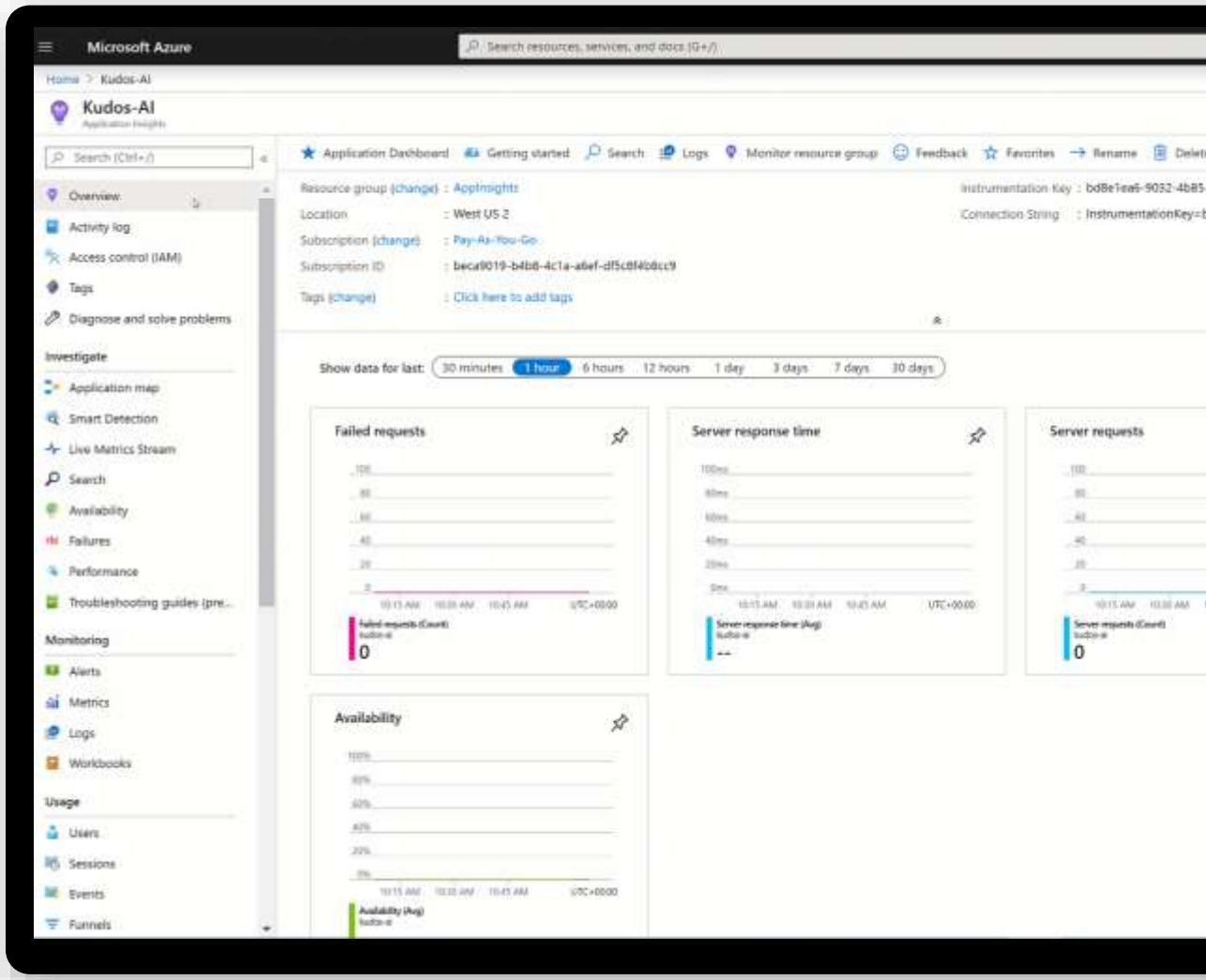
https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/test-studio

Testing and telemetry tools

Microsoft Power Platform

Application Insights (canvas apps)

- Help you diagnose **issues** and understand what **users actually do** with your canvas apps
- Help you **troubleshoot issues**, drive better business decisions and **improve the quality of your apps**.



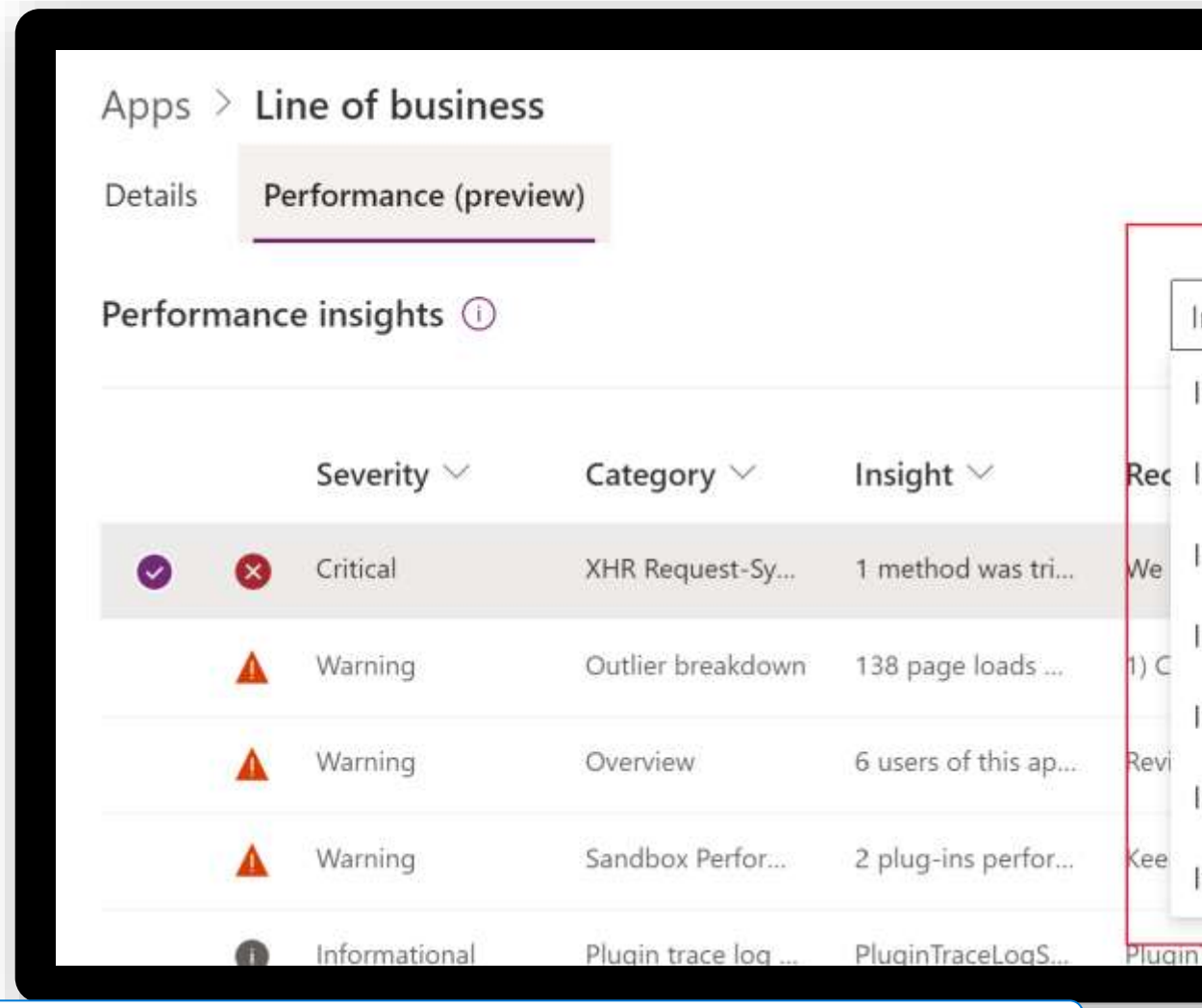
https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/application-insights

Testing and telemetry tools

Microsoft Power Platform

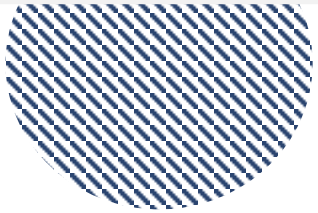
Performance insights for model driven apps (preview)

- Tools that analyzes **runtime user data** and provides a **prioritized list of recommendations** to help **improve the performance** of model-driven apps
- Insights are generated based on collected user data of your model-driven app **every 24 hours**.



https://learn.microsoft.com/en-us/power-apps/maker/common/performance-insights-overview

Deploy automation tools



Pipelines in Power Platform

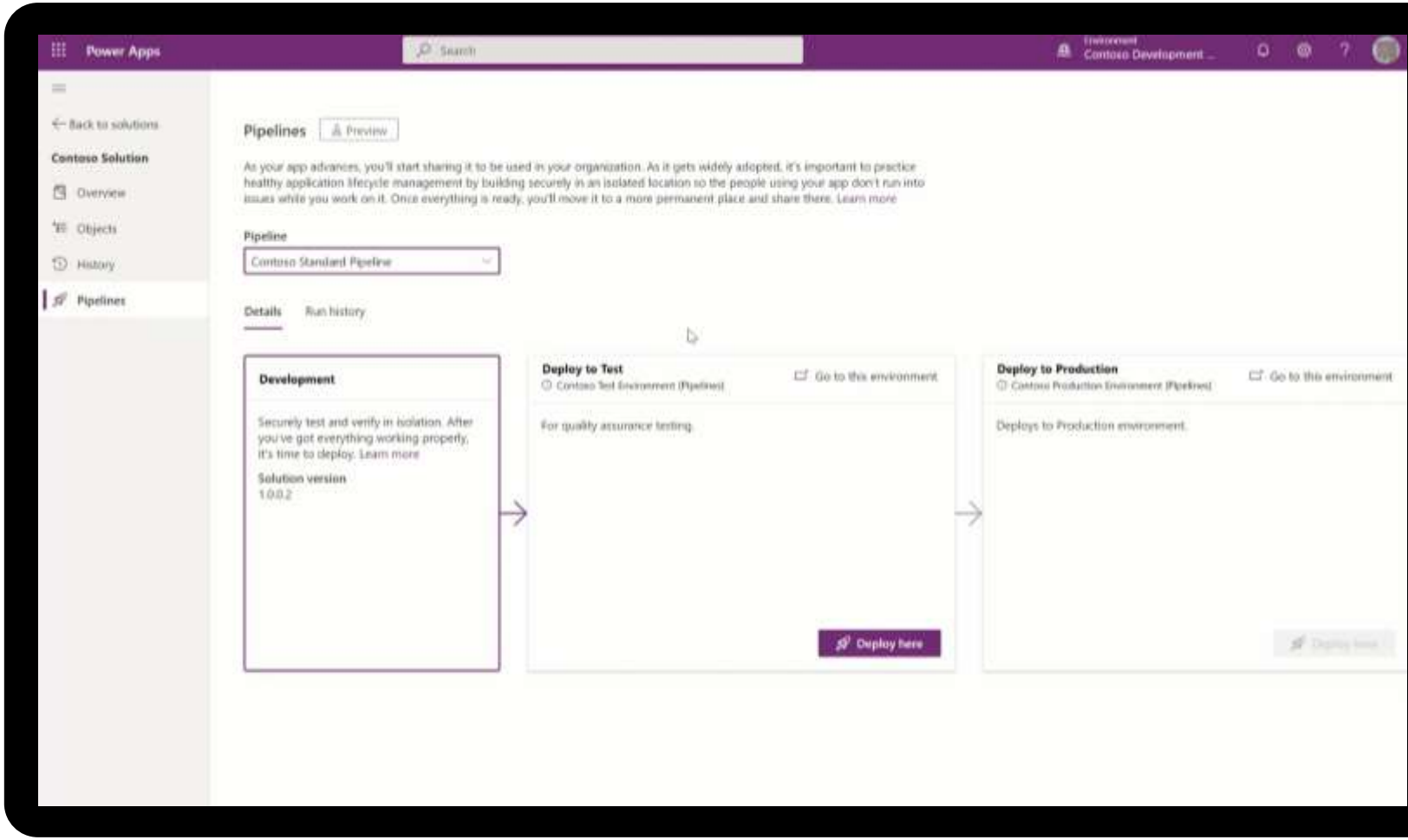
Deployment automation capabilities created for anyone

Code-First Developers can receive information and run pipelines using their preferred tools

Underlying tasks required previously to accomplish the same outcome are now handled by pipelines

Admins easily configure automated deployment pipelines in minutes then manage access and view all deployment activity across your organization within the same location

Makers have an intuitive user experience for deploying their solutions in just a few clicks – directly within the environment they're already working in



Demonstration

- Pipelines

https://youtu.be/uILb28j4lso?si=oyS0D5OIMjoAZDKy&t=94

Start me at 01:34
Stop me at 07:00

Deploy automation tools

Microsoft Power Platform

Pipeline automation with Azure DevOps

Power Platform Build Tools

Use to automate common build and deployment tasks related to apps built on Microsoft Power Platform on Azure DevOps. These tasks include:

- Synchronization of solution metadata (also known as solutions)
- Generating build artifacts
- Deploying to downstream environments
- Provisioning or de-provisioning environments
- Perform static analysis checks against solutions by using the Power Apps checker service

Learn more

Deploy automation tools

Microsoft Power Platform

Pipeline automation with GitHub

GitHub Actions

Learn more

Create workflows in your repository to build, test, package, release, and deploy apps; perform automation; and manage bots and other components built on Microsoft Power Platform.

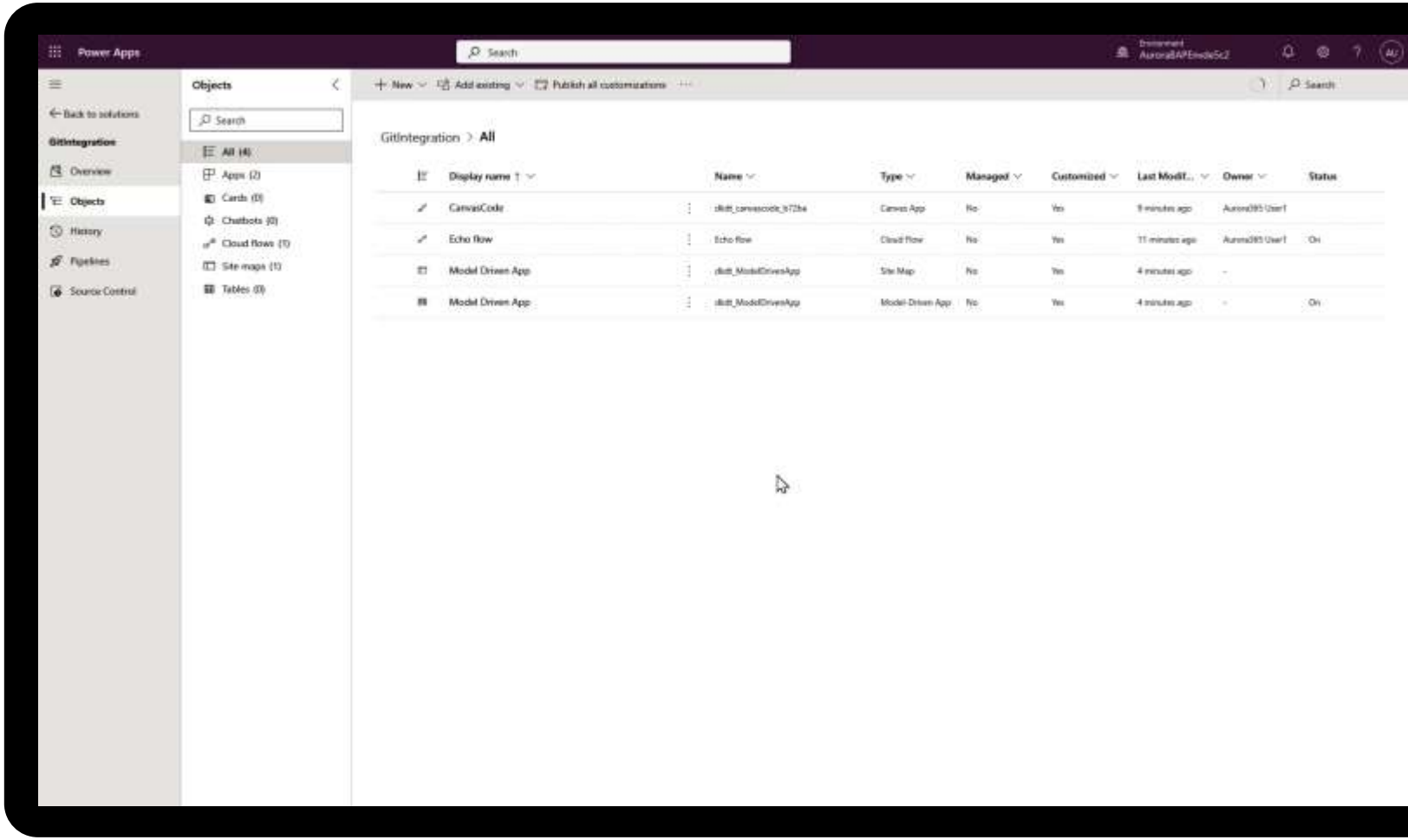
- Importing and exporting application metadata (also known as solutions)
- Deploying to downstream environments.
- Provisioning or de-provisioning environments
- Performing static analysis checks against solutions by using Power Apps solution checker.

Use a native git integration to quickly sync and deploy

Natively sync anything built in Power Platform **to source control**

Trigger ADO pipelines directly from the Power Apps portal

Work safely with reviews and appropriate processes to mitigate risk



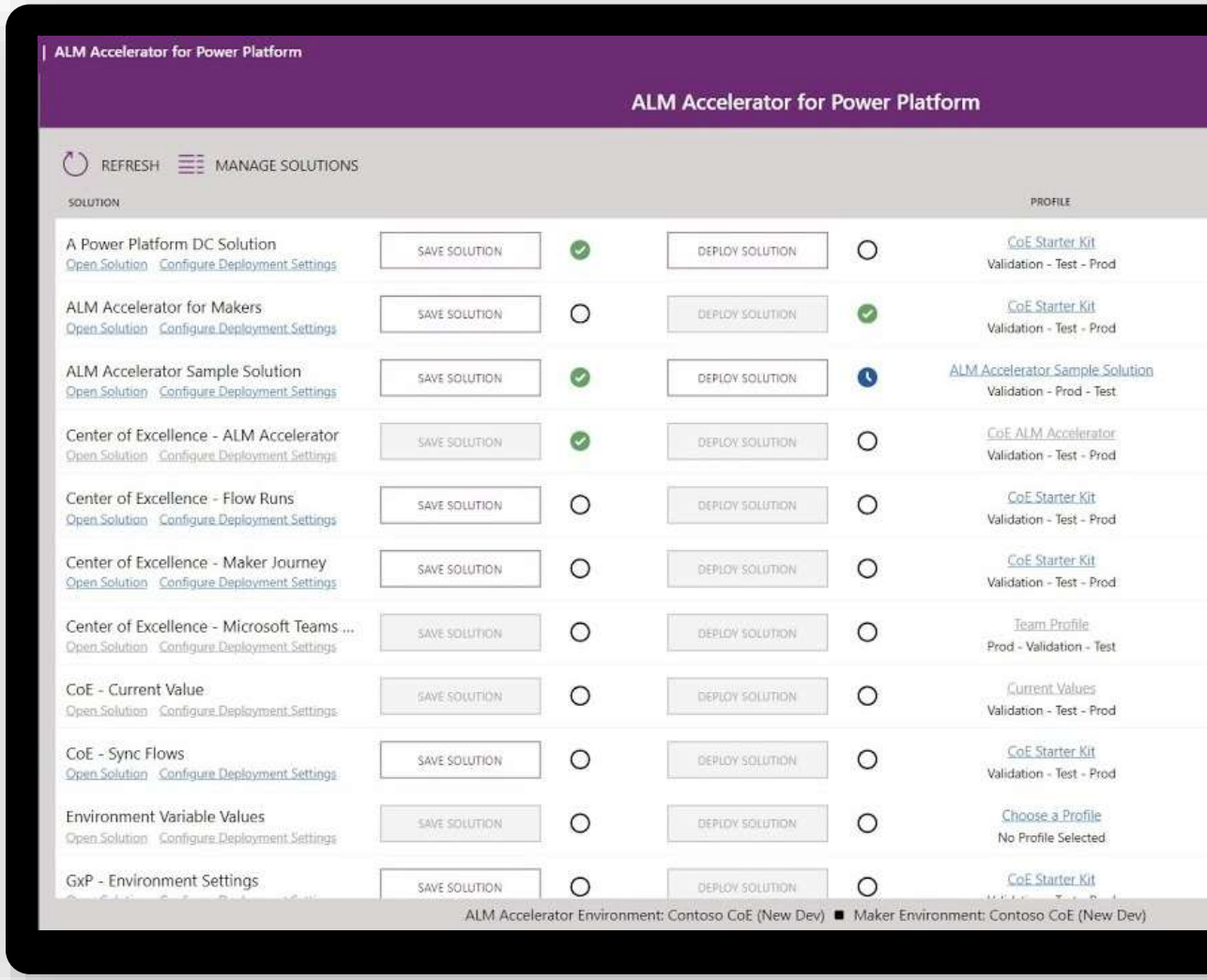
Public Preview: Sep 2024

Deploy automation tools

Microsoft Power Platform

ALM Accelerator

- Perform **source control**, **enable version history**, and deploy their solutions in Power Platform
- Its target users are makers with no ALM knowledge
- Uses **Azure DevOps Pipelines** through **a canvas app** to provide a **simplified interface** for makers to regularly commit and create pull requests for their development work
- It as a **reference implementation** of ALM patterns and practices using other in-built platform capabilities and **can be customized**



https://learn.microsoft.com/en-us/power-platform/guidance/alm-accelerator/overview

Demonstration

- ALM Accelerator (05 mins)

https://youtu.be/daK6LuR9Uuk?si=IwiJjm0HAyYFjkJz&t=158

Start me at 02:38
Stop me at 07:00

Deploy automation tools

Microsoft Power Platform

Which tool I should choose?

This table isn't exhaustive and is meant to help you make an informed decision

Capability	Pipelines	ALM Accelerator	DevOps/GitHub
IT / Developer involvement	Not required	Up-front set up	Required for every project
Source Code integration	No, but planned	Yes	Yes
Maker requires elevated privileges in target environment	Yes, but service principal support planned	No, service principal supported	No, service principal Supported
Quality Control	Minimal	Best practices	Unlimited
Democratized for Citizen Development	Yes	Yes	No
In-Product Experience	Yes	Canvas app provided, but not in maker experience	No
Support	Microsoft supported	Power CAT support via GitHub Issues and Discussions	Microsoft supported and GitHub Issues
Customization	No	Yes	Yes
Code-first development	No	Yes	Yes

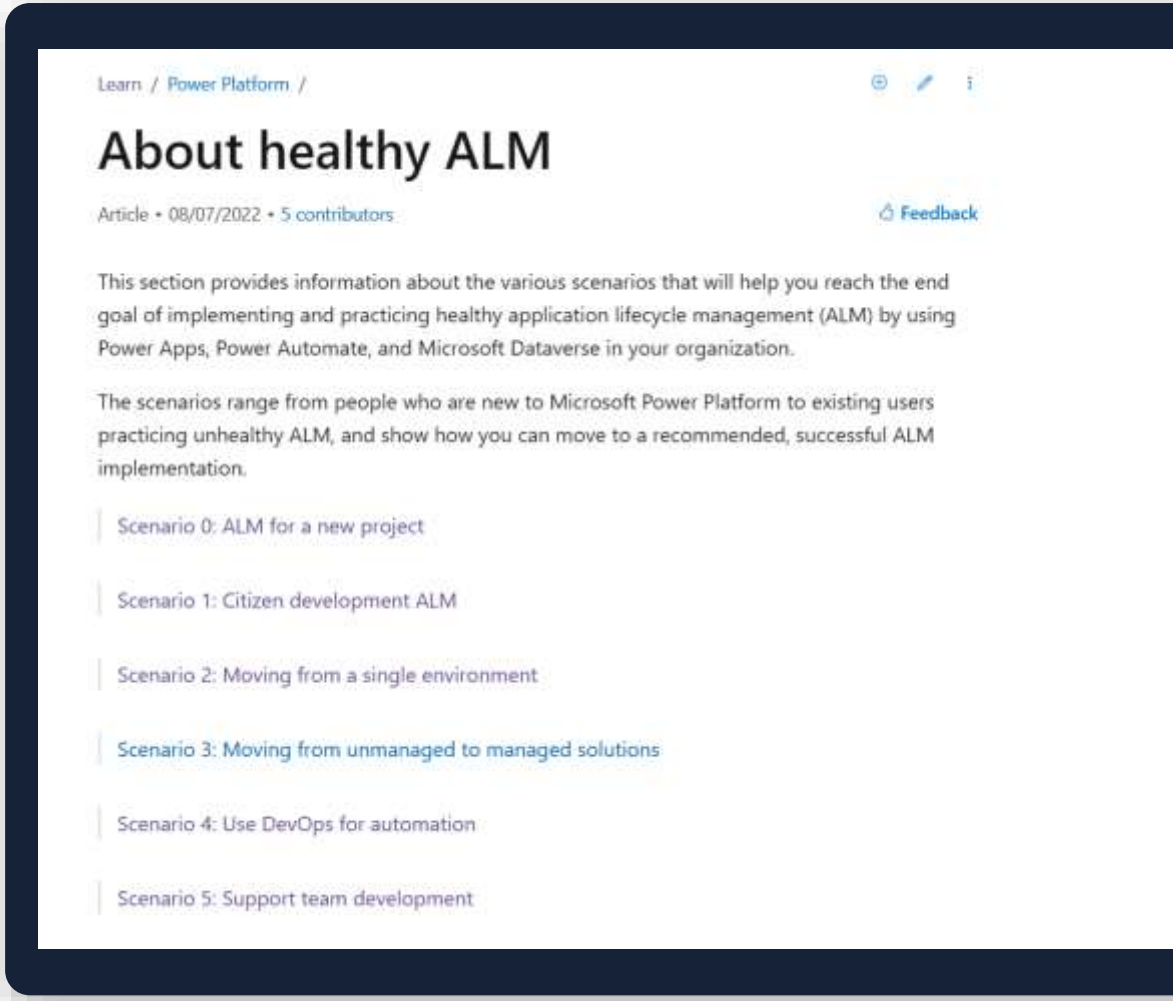
https://learn.microsoft.com/en-us/power-platform/guidance/alm-accelerator/overview#whats-the-right-choice-for-my-organization

Which ALM approach I should choose?

There are few questions you must answer first

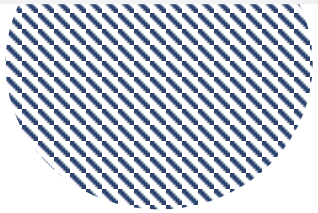
- Identify the stage of Power Platform adoption journey
- Define ALM requirements for each type of application and lifecycle stage
- Identify the ALM and source code policies
- Create a roadmap for ALM adoption

Then use this scenarios as reference



Creating and sharing custom components

Canvas components

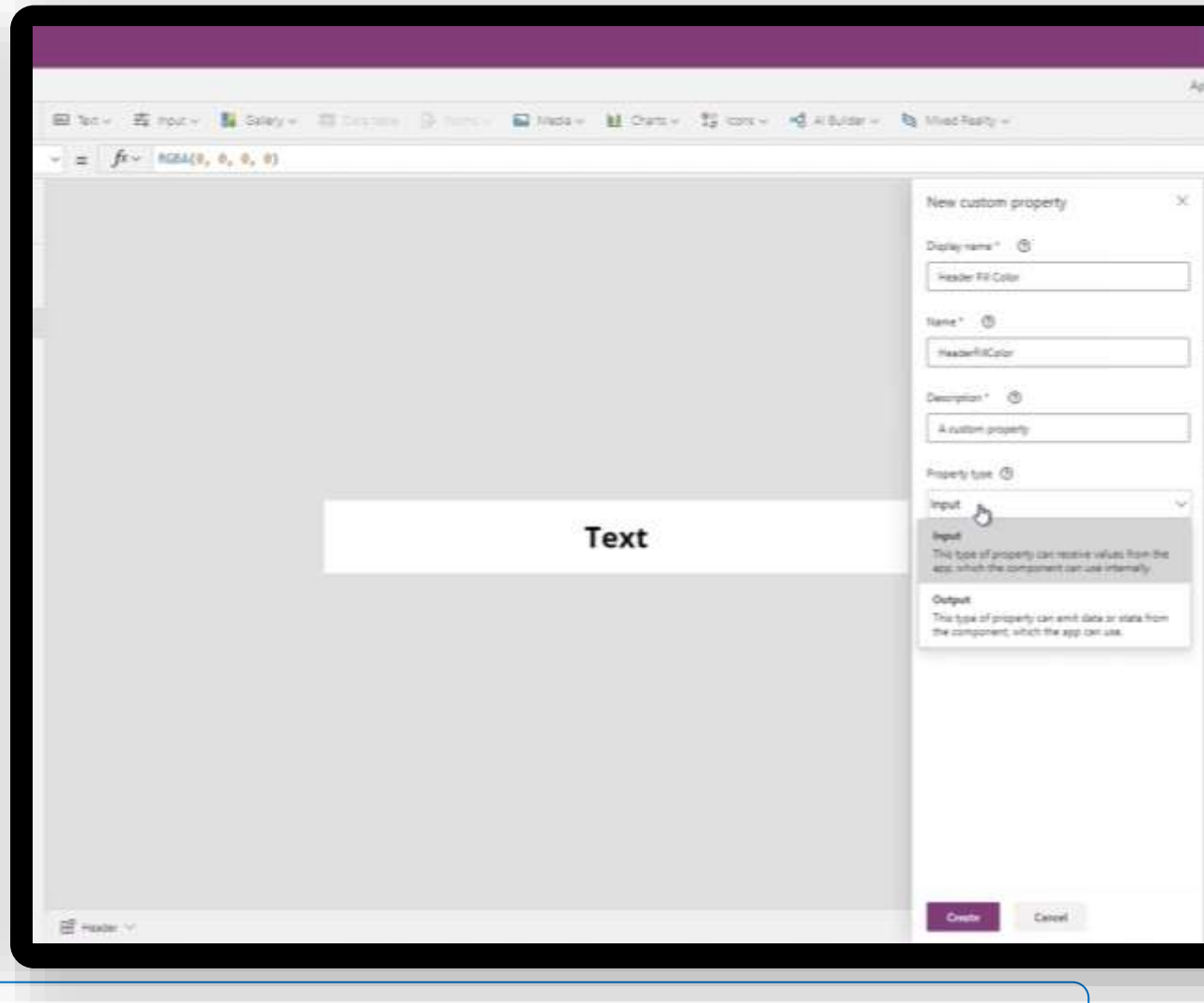


Canvas components

Microsoft Power Platform

Promote collaborative development and help ensure consistency across apps

- They are **reusable building blocks** designed for app makers to create custom controls, that can be used within a single app or across **multiple apps**
- Any changes made to a component's definition will be **automatically reflected** across all instances within the app.
- **Help reduce the quantity of controls** in an app, increasing **performance**.
- Can be **shared between environments** using **Component libraries**.



https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/create-component
https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/component-library

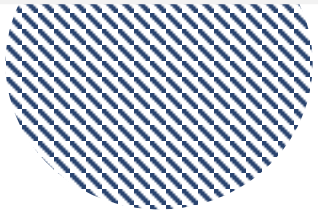
Demonstration

- Using component libraries (05 mins)

https://youtu.be/grTdm_JSasU?si=CAw40m0aeclrABUY&t=87

Start me at 01:25
Stop me at 06:11

Custom components

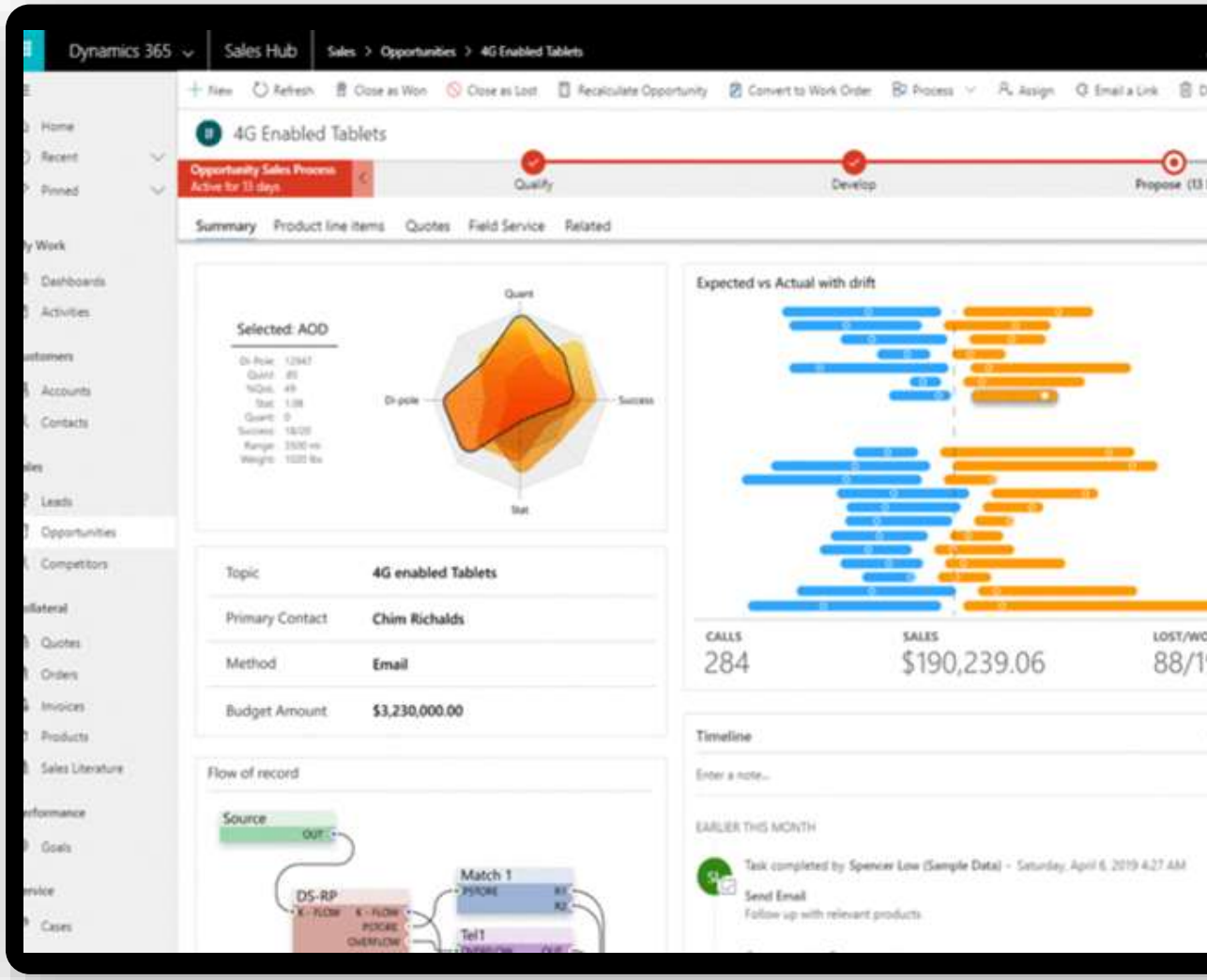


Code components

Microsoft Power Platform

Create new user experiences with Power Platform Component Framework

- Developers can build **compelling visual components** for **Power Apps** and **Power Pages**
- Custom components are **responsive**, **reusable** and support **multiple form factors**.
- Reuse your **current IP and skills–framework**. It is based on standard web technologies **TypeScript/JS**, **CSS** and **HTML5**.
- **React** and **Fabric** support available in **preview**



https://learn.microsoft.com/en-us/power-apps/developer/component-framework/overview

Additional recommended content

Building PCF Controls FULL COURSE for Beginners

Carl de Souza presents a complete walkthrough about how to build PCF controls

Learn more!

Catalog in Power Platform (preview)

Share customized and reusable components and templates

Using **Catalog**, developers, makers, and admins can deliver the **best solutions** for their users.

Crowd-source and find templates and components within their organization easily

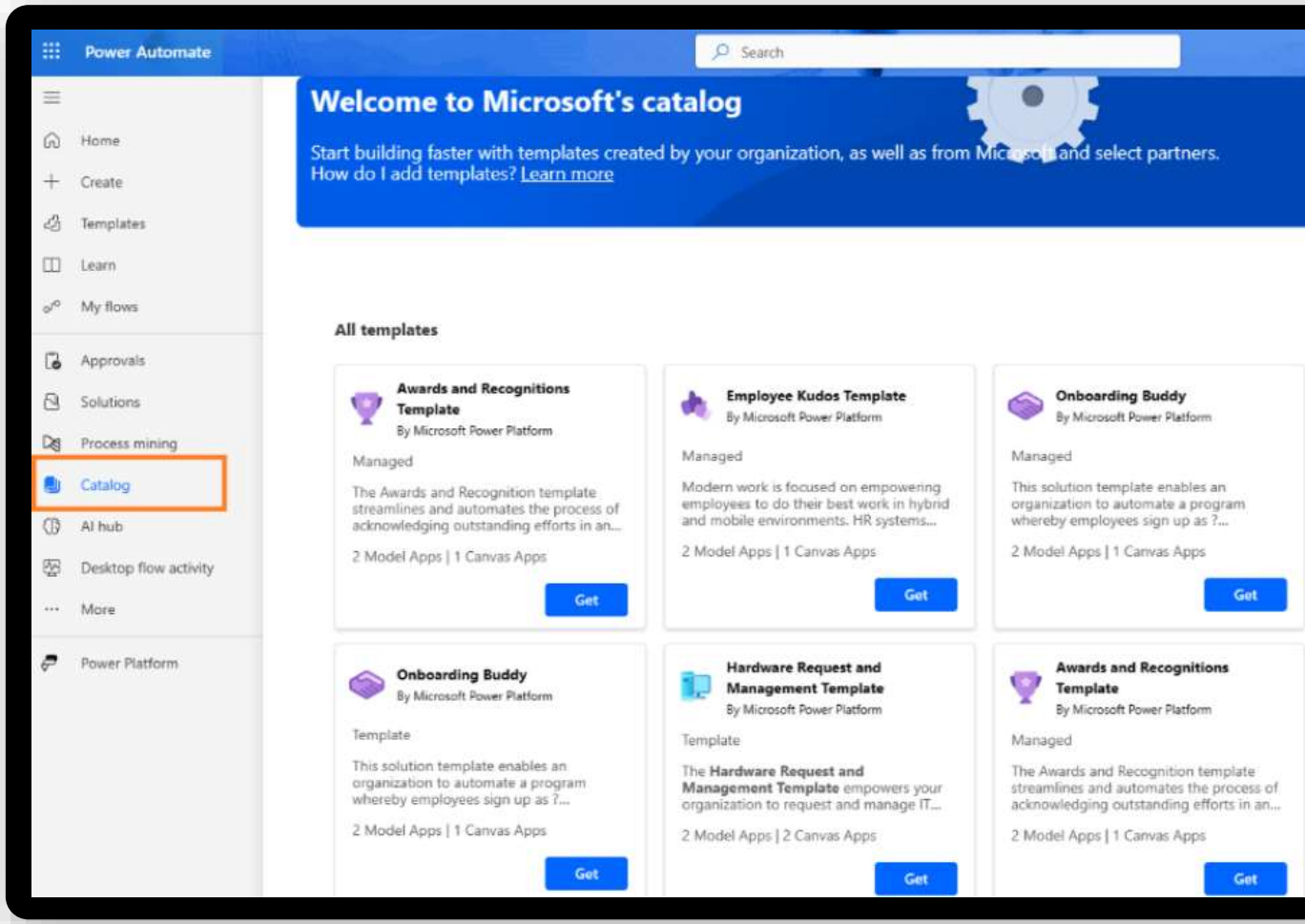
---

Find and install the latest and authoritative version of a component

---

Get started with templates and components that provide immediate value

Learn more!



Demonstration

- Catalog (05 mins)

Before you go

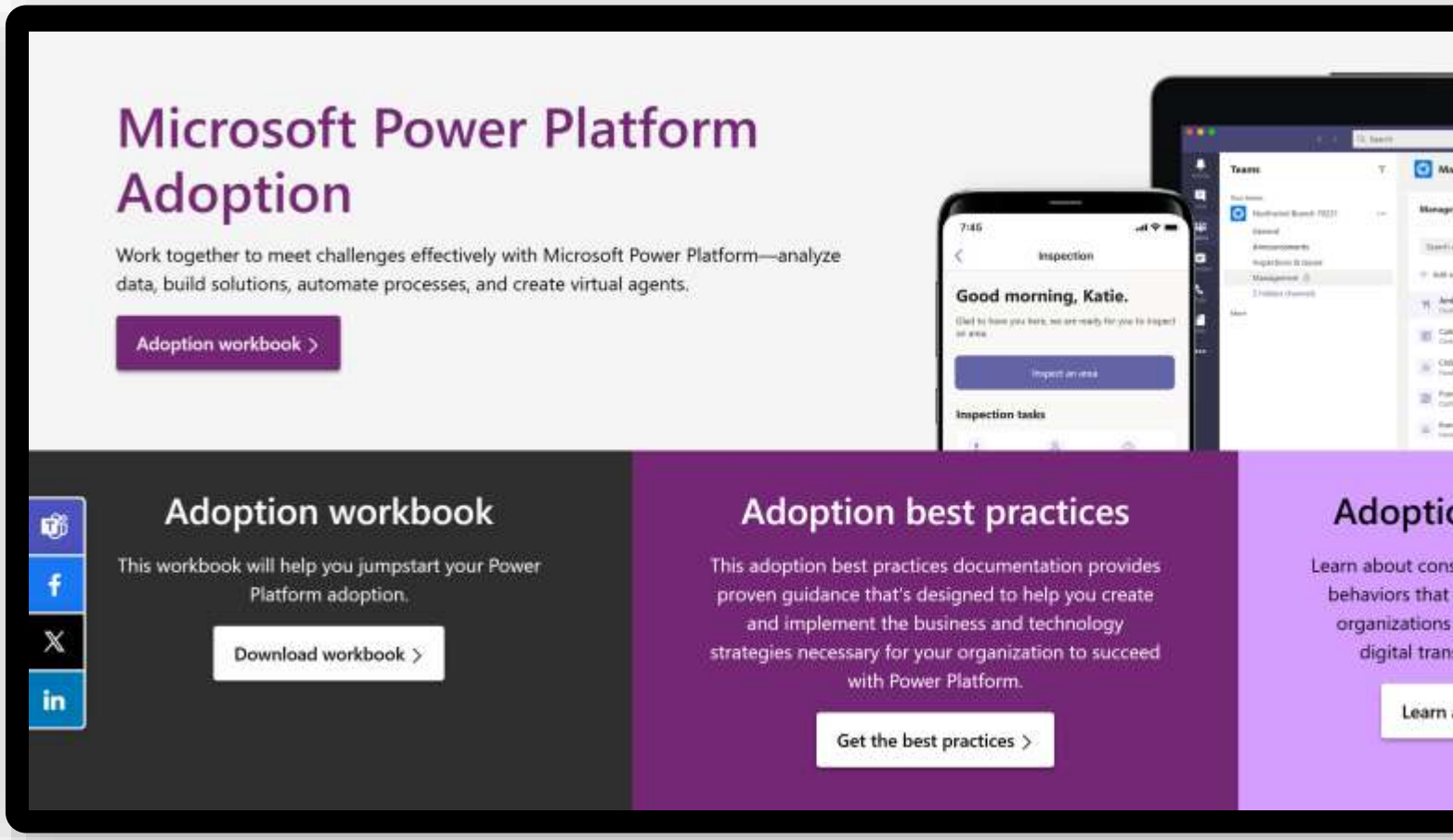
Plan your Power Platform adoption

Learn about your low-code maturity level

Learn how low-code can underpin your **successful digital transformation** and scale Microsoft Power Platform adoption to your **entire organization** with our adoption guide.

- Get guidance to help you **create** and **implement** the **business** and **technology strategies** to succeed with Power Platform.
- Learn about **themes**, **patterns**, **practices** and **behaviors** that underpin successful Power Platform implementations.
- Learn about how to **kickstart** your **Center of Excelence (CoE)**.

Learn more!



Get training

Use online resources or organize a “in a Day” workshops

Microsoft Learn have **online learning paths** from beginner to advanced level. “**In a Day**” workshops let you have a hands-on experience of building great business apps without writing code.

Power Apps

Learning paths

Power Automate

Learning paths

Power Pages

Learning paths

Copilot Studio

Learning paths

Power Platform “In a Day” workshops

Stablish guidelines and best practices

Use official documentation as a starting point

These guidelines provides **best practices**, **implementation**, and **architecture guidance** information from the **Microsoft teams** that works with our enterprise customers.

Power Apps

Power Automate

Power Pages

Copilot Studio

Stablish guidelines and best practices

Address complex topics with Power Platform whitepapers

Get **in-depth information**, **solutions**, and **recommendations** to help you address complex scenarios and inform your decision-making.

Whitepaper	Description
Develop a tenant environment strategy to adopt Power Platform at scale	Shows you how to align your Power Platform tenant environment strategy with the product capabilities and vision.
Enterprise security with Power Platform	Shows you how to align Power Platform with your security practices.
Activating Managed Environments	Explores the features of Managed Environments in Power Platform.
Application modernization with Power Platform	Explores the benefits, strategies, and best practices of modernizing applications with Microsoft Power Platform.
Migrating apps and flows from the default environment	Outlines considerations and best practices for migrating apps and flows from the default environment.
Architecture white paper	Provides a comprehensive view of the capabilities of the Power Pages platform and how it scale, offer high reliability and availability, and protect business data
Security white paper	Describes how Power Pages offers enterprise grade security and the tools and capabilities

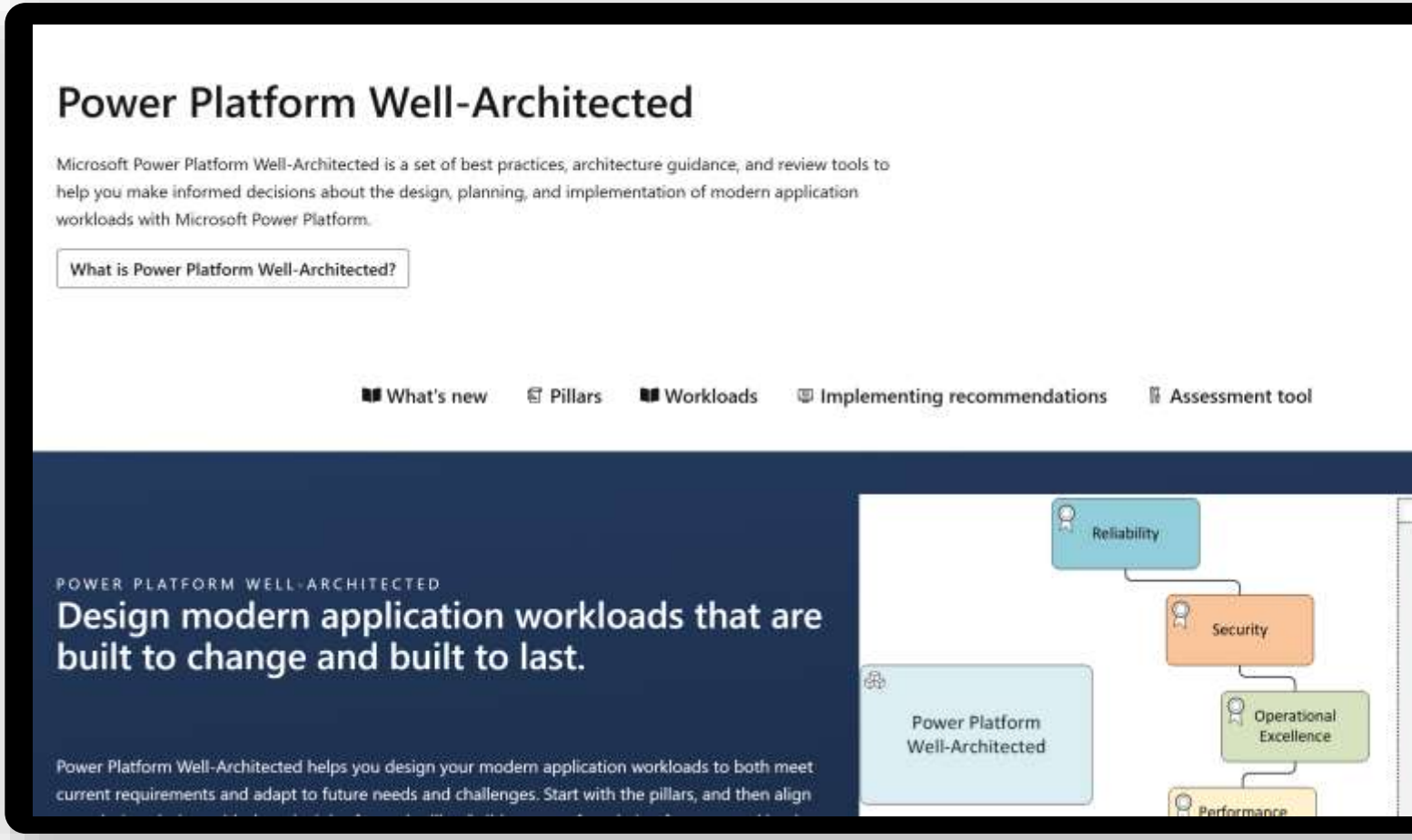
Stablish guidelines and best practices

Power Platform Well-Architected

A framework that can improve the quality of your Power Platform workloads

- A set of **best practices**, **architecture guidance**, and **review tools** to help you make informed decisions about the design, planning, and implementation of Power Platform workloads.
- Based on the methodology and guidance of the Azure Well-Architected Framework.
- Have an **assessment tool** to help **identify areas of enhancement** and **iteratively improve** your workloads.

Learn more!



Demonstration

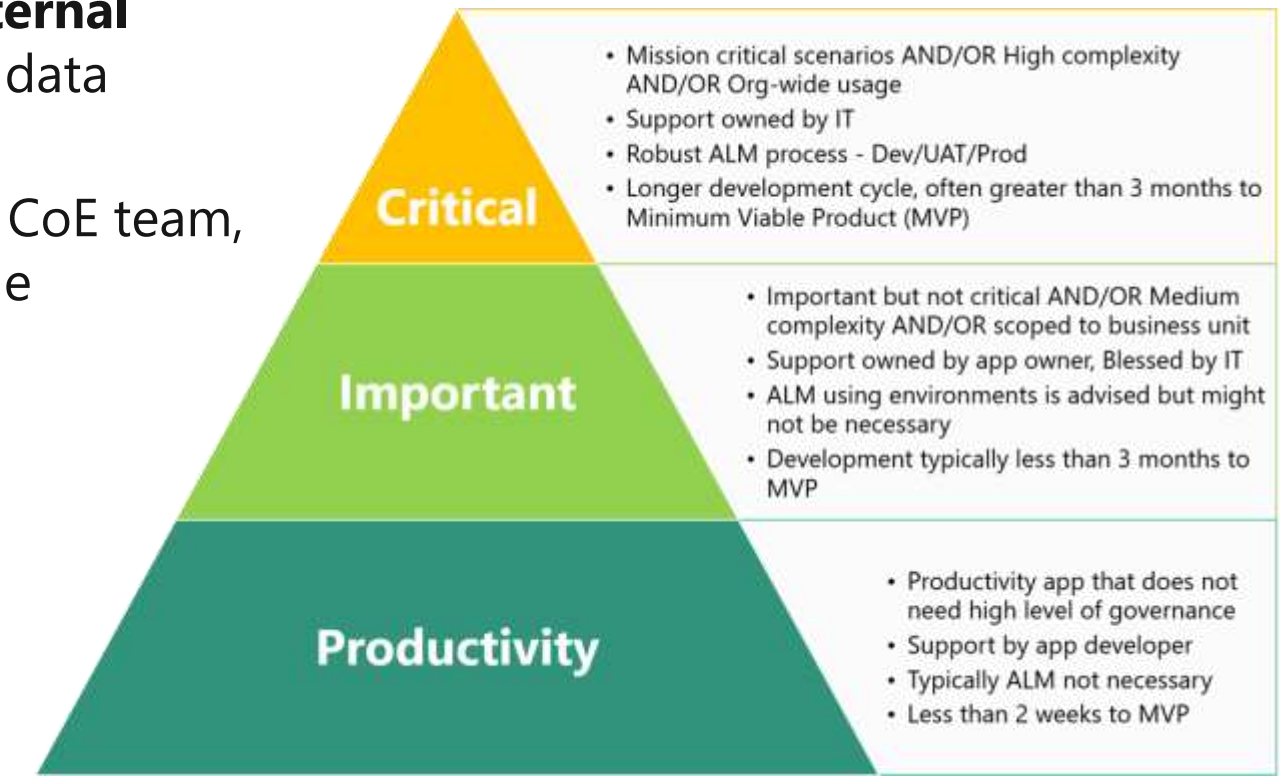
- Navigation on Power Platform documentation (05 mins)

Final thoughts

Recommendations

1) Adopt governance process that are **suitable** for your project's **characteristics** or that are **required** by **CoE team**.
2) Please refer to **CoE documentation** for **internal standards** such UI, coding, error handling, data storage e.g.
3) If an internal standard was **not defined** by CoE team, work with them to **create one based** on the **guidelines** described on Power Platform documentation.
4) Please reach CoE team about your app or flow **tier of support**, which will determine **ALM cycle**, development and support **responsibilities** and Power **Platform available resources**.

Tiers of application support (example)



Next steps

Continue sua jornada de conhecimento

Demonstrações técnicas de Power Platform

Eventos no **Microsoft Teams** conduzidos pelos especialistas em Power Platform da Microsoft Brasil. Para participar procure seu **gestor(a) de conta** (AE) ou **especialista** (SSP).

Desenvolvimento de aplicativos

Tópicos
Canvas Apps
Microsoft Dataverse
Model Driven Apps
Power Pages

Duração
03 horas

Saiba mais

Governança e ALM

Tópicos
Governança
Segurança
Monitoramento
Centro de Excelência (CoE)
ALM/DevOps

Duração
03 horas

Saiba mais

Hiper automação

Tópicos
Process e Task Mining
Cloud flows (DPA)
Desktop flows (RPA)
Gerenciamento e monitoramento
Hosted machines (VMs SaaS)
Automações e integrações

Duração
03 horas

Saiba mais

Criando copilots com Copilot Studio

Tópicos
Ecosistema Microsoft de copilots
Visão geral do Copilot Studio
Recursos baseados em IA generativa
Automações e integrações

Duração
03 horas

Saiba mais

Continue sua jornada de conhecimento

Power Platform Connect

Site mantido pelos especialistas em **Power Platform** da **Microsoft Brasil**. Nele você encontrará:

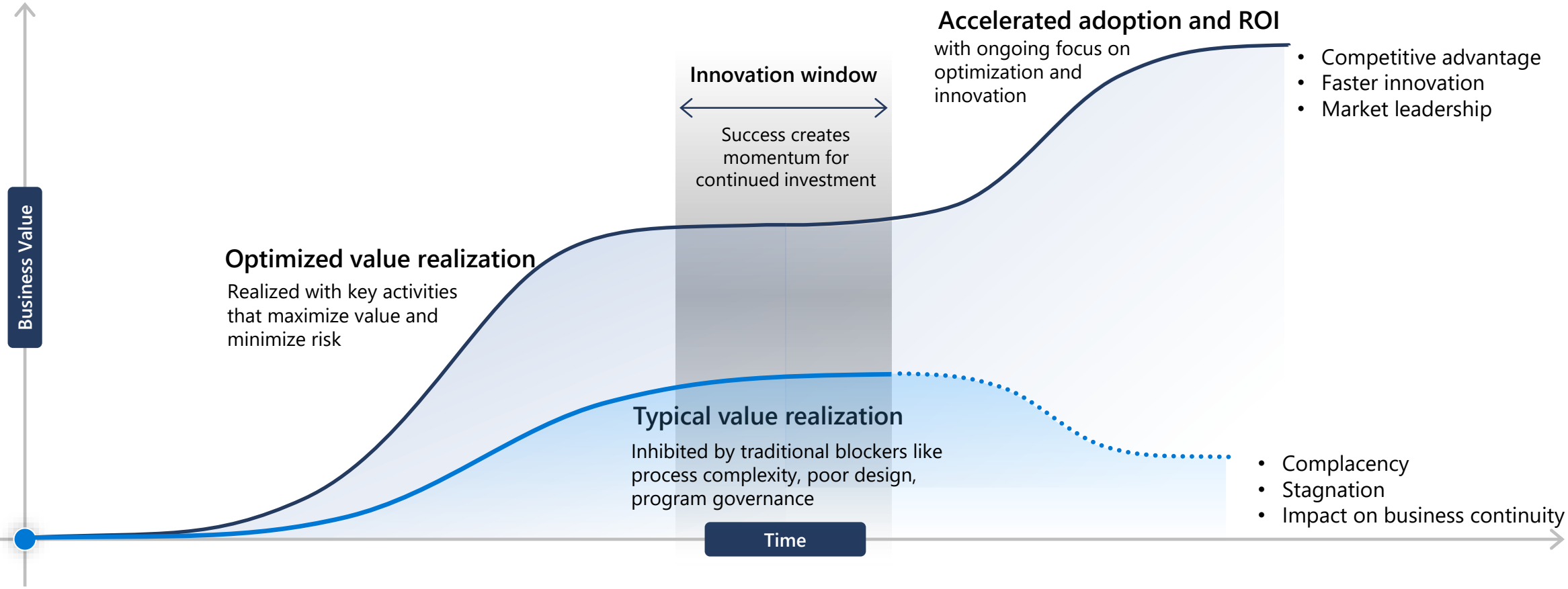
Biblioteca de conteúdo
Coleção de links oficiais das soluções que compõem a Power Platform

Eventos online
Eventos online promovidos pela Microsoft ou seus parceiros.

https://microsoft.github.io/powerplatformconnect

Maximize seus investimentos com o Microsoft Unified

Acelere seu tempo de implantação com serviços liderados por especialistas, desde o planejamento até a implantação e muito mais



Conte com especialistas para construir seus aplicativos

Com acesso direto aos especialistas da Microsoft, você pode criar aplicativos personalizados usando o Power Platform para resolver desafios de negócios e automatizar processos de negócios

O que você quer alcançar...

- Estabeleça um modelo de governança e padrões e prepare-se para a IA
- Implantação bem-sucedida da plataforma de energia em toda a empresa
- Validação de processos de projeto e desenvolvimento de arquitetura de soluções

Como entregamos...

- Assistência na criação de um Centro de Excelência (COE) para governar a criação de aplicativos e acelerar a adoção em toda a organização
- Orientação prescritiva de especialistas para acelerar a implantação com workshops de Maker e Developer para melhorar as habilidades da equipe
- Revisões técnicas e funcionais para validar planos e abordar problemas e riscos

Saiba Mais

Visite **aka.ms/Enhanced-Solutions** para agendarmos uma sessão com objetivo de determinar como nossas **Soluções Aprimoradas** podem ajudá-lo a alcançar os resultados desejados

Microsoft Power Platform

Thank you!

Ricardo de Souza
Technology Specialist - Power Platform
rdesouza@microsoft.com
https://www.linkedin.com/in/ricardodesouza/