Power Platform Adoption Framework

Second Edition
April 2020

Adopting, managing, and governing Power Platform at scale

Table of Contents

Introduction

Preface to the Second Edition

Much has happened since we first released the Power Platform Adoption Framework in 2019.

- **The platform has continued to mature** with the launch of powerful new technologies including Power Virtual Agents, AI Builder, Power Apps Portals, UI flows, and—of course—ongoing improvements to Power Apps, Power BI, Power Automate, Common Data Service, and the Dynamics 365 first-party apps.
- **The adoption framework itself has become fully-community driven** with a successful move to GitHub announced at the London Power Platform User Group on 19 September 2019.
- **Power Platform adoption has soared worldwide,** and with it so to has grown the incredible community of practitioners as we've welcomed citizen developers and full-time Power Platform developers alike to a passionate global network whose in-person and virtual events continue to grow bigger and more technically sophisticated.

And, of course, as we write this people around the world are grappling with the COVID-19 virus and the upending changes to our way of life that the virus has wrought. Power Platform is helping us through these times as well, not just thanks to solutions like those for Crisis Communications or Emergency Response, but thanks also to the #PowerAddicts and others in the worldwide Microsoft business applications community who support one another personally and professionally every day.

So we're introducing some new ideas—and new ways of thinking about existing concepts—as we enter the Power Platform Adoption Framework's second year. As a "framework", we're committed to broadly applicable best practices for adoption at scale, not to being a technical manual. We're encouraging organizations to create playbooks (literally or figuratively) that complement our shared "framework" with organization-specific technical considerations.

Context is so important here. For example, whilst the framework's best practices will apply broadly to big enterprise banks and small / medium businesses alike, those two very different types of organizations will naturally face different drivers and constraints. That's why we try to avoid being technically prescriptive unless we feel it necessary. Likewise, some adoptions feature a significant "citizen developer" emphasis, whilst others do not. Power Platform Adoption Framework is about the ability to adopt at scale either way.

Moving forward we will continue to focus on openness with the community so that the framework can truly be a globally-inspired collection of best practices for adopting Power Platform at scale. This is reflected both in the ongoing work on GitHub, and in the publisher's de-emphasis on marketing branding in the official second edition white paper. We'll honor this focus by featuring the world's great Power Platform community cities on the cover of the official release white papers, beginning with London on the cover of the second edition.

Power Platform is a first-class citizen in the cloud transformation journey. Indeed, the platform exists alongside Azure and Microsoft 365 as one of Microsoft's "three clouds". We increasingly see this trend at forward-thinking organizations in some of the most complex or regulated sectors. They are realizing that Power Platform is an enterprise-grade platform for enterprise-grade workloads.

- Andrew Welch, Microsoft MVP
Power Platform Adoption Framework team

www.PowerPlatform.af

Our Philosophy

More and more organizations are transforming their business in the cloud. They are modernizing legacy enterprise applications with enterprise-grade capabilities such as Microsoft’s Common Data Service (CDS), and they are bringing rogue IT and quasi-apps out of the shadows. Organizations are empowering citizen developers to connect siloed data, engage customers, and drive return on investment (ROI) with robust solutions in Power Apps, Power BI, Power Automate, Power Virtual Agents, CDS, and Dynamics 365.

Citizen Developers are the business users building components on Power Platform in service of their full-time job functions, but they're not full-time working on the platform. They don't work in IT. They use Power Platform to build apps, data components, automations, chatbots, and more using no-code / low-code tools. The Power Platform Adoption Framework helps to empower business users to build solutions that replace spreadsheets, legacy databases, SharePoint lists, manual processes, and other quasi-apps within the guardrails of sound enterprise management and governance of the platform.

We’ve seen it. Today, organizations are using the platform to recruit and onboard talent, automate their marketing, streamline disaster response, simplify reporting and planning across their businesses, enable field engineers and inspectors, manage complex cost-sharing across business units, bring new insurance products to market, and more.

People around the world are using Power Platform to improve the passenger experience at airports, make government more responsive to citizens, micro-target retail customers, streamline their consulting businesses, control the cost of fuel in transportation-focused businesses, connect their employees with volunteer opportunities, keep the trains running on time, and even give massages (in airport lounges).

Mature organizations understand that rigor, discipline, and best practices are required to fully adopt the platform at scale.

Power Platform Adoption Framework is the global community-driven best practice for adopting at scale.

It helps enterprise organizations:

- Get to value quickly
- Educate, train, and grow their community of citizen developers and power users
- Create durable partnerships between business, IT, and the user group community
- Continuously improve ROI on the platform by identifying and migrating new workloads
- Blend agile app development with rigorous enterprise management and governance

We’ve studied and learned from our friends, partners, and colleagues around the world. We’ve built a long history with Power Platform’s predecessor technologies and we’ve transformed businesses in the cloud. **We’re sharing our Power Platform Adoption Framework so everyone can use it as we believe it will help foster an already vibrant and thriving community of users around this technology.** As the publisher, AIS is committed to continued investment, updating, and sharing of the framework as best practices evolve. We hope you’ll join us, build with us, learn from us, and teach us better ways to do things.

Scalability Matters

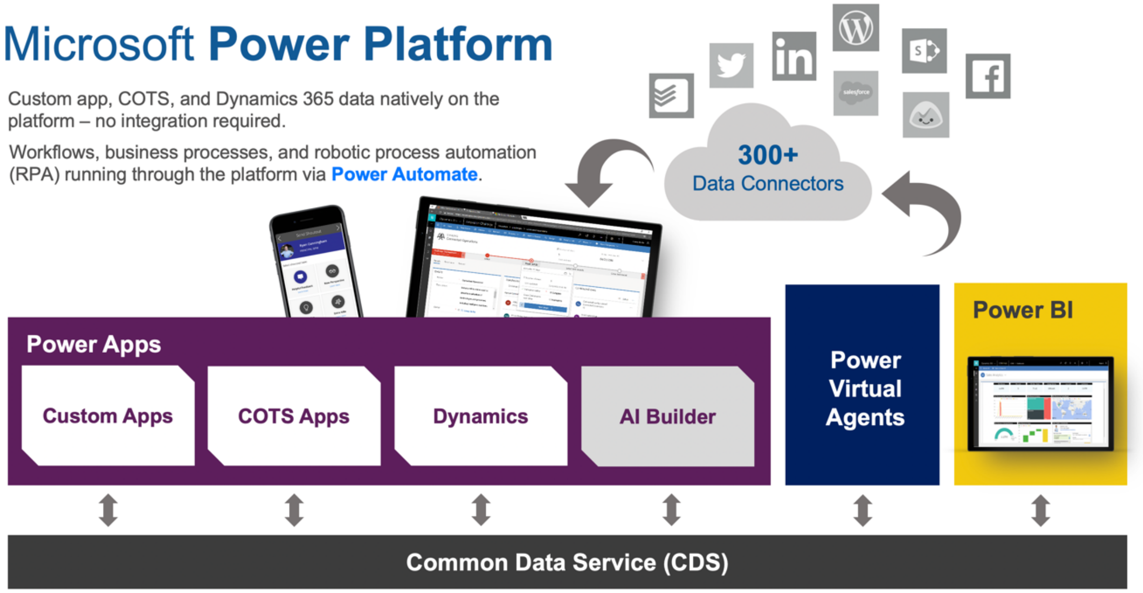
The Power Platform Adoption Framework is built on the idea of scale; in other words, implementing the platform across large organizations to solve a range of business problems, small and large. Scale matters on Power Platform for three reasons.

- **Network Effect.** The magic of the Common Data Service is in its ability to establish a single source of truth for the data stored within it, and then for app experiences to be built atop that data to address specific use cases. The more apps running on CDS, the more valuable that single source of truth becomes. The more users interacting with those apps and storing data in CDS, the richer that data source becomes.
- **Standardization.** Once upon a time, organizations embraced the applications inside of Microsoft Office because they worked similarly and played well together. It was easier to train and support on a single platform. The same principle applies to Power Platform. Rather than leave our users' everyday app interactions to a hodgepodge of spreadsheets, one-off databases, and third-party tools, IT organizations use Power Platform to establish a single ecosystem in which dozens, hundreds, or even thousands of workloads can live, work, and be secured.
- **Return on Investment.** This is about the math of license costs. Workloads running on separate platforms or applications incur separate license costs. If your organization pays X amount for a Power Apps license and runs one app, the cost to license that app is X. Adding a second workload reduces that cost to X / 2. Four workloads are X / 4. By the time we've migrated our twentieth workload to Power Platform, we've reduced the license cost per app by a factor of twenty. This is important because in the world of Azure, or other similar cloud infrastructure services, we pay for consumption. In other words, our costs rise the more we consume. On Power Platform, we're incentivized to consume more so we can increase the ROI of our license investment. Migrating workloads onto the platform also allows us to retire them (and their license and support costs) elsewhere.

Power Platform

We can't truly embrace what we don't understand. Having a common understanding is critical to determine common best practices for adoption and beyond. Power Platform is a vastly capable technology, but many still struggle to put all of its pieces together in their mind. We'll take a moment to change that here.

Power Platform Basics


Microsoft Power Platform
Custom app, COTS, and Dynamics 365 data natively on the platform – no integration required.
Workflows, business processes, and robotic process automation (RPA) running through the platform via Power Automate.
300+
Data Connectors
Power Apps
Custom Apps
COTS Apps
Dynamics
AI Builder
Power Virtual Agents
Power BI
Common Data Service (CDS)

Let's start with the "app". We find that's what many people think of when they think of this technology, so we'll begin there. These are Power Apps! Find them peeking out of the purple box in the diagram above. Full-time Power Platform developers and citizen developers alike use Power Apps to build applications that run natively in mobile, web, and desktop environments across any popular combination of devices and operating systems.

There are three flavors of Power Apps:

- **Custom mobile and web apps** are built specifically for your organization by some combination of in-house Power Platform and citizen developers, or with Microsoft partners.
- **Mission-focused COTS apps** (Commercial Off The Shelf), also known as "ISV Apps", are built by Microsoft partners to meet specific yet common use cases. They run on the platform and are often made available in AppSource (think the app store for Power Platform).
- **Dynamics 365 (D365)** is Microsoft's suite of large business applications that meet common requirements such as customer relationship management (CRM), marketing automation, field service, and more. You can think of D365 as a suite of big Power Apps.

Let's also take a moment to discuss AI Builder. While not an app type itself, the AI models we create with this toolset allow us to build a variety of predictive and automated activities directly into our Power Apps. Among them:

- Form Processing, which allows us to read and save information from standard documents
- Object Detection, which recognizes and counts things in images
- Prediction as to whether something will happen (based on data, of course!)
- Text Classification by meaning so that text is easier to analyze

Power Virtual Agents, similarly, applies AI and automation in allowing us to create virtual agents—chatbots—in a no-code fashion like what we've come to expect elsewhere on the platform. These virtual agents allow organizations to automate elements of their engagement with customers (externally) and employees (internally). Experienced developers can extend Power Virtual Agents via the Azure Bot Framework.

Each of these capabilities sit atop the CDS. If you're familiar with XRM, or the underlying database in recent versions of Dynamics 365, then you already know CDS. Otherwise, what's important to know here is that all the data from all the apps installed in an environment lives in CDS.

Put another way, CDS is the single source of truth for the data you use in your apps. Take geographic locations, for example. We often have a need to reference a physical place across many of our apps. But creating a different dataset of physical places living inside of a single app means that we often create different ways of representing or naming the same place. The United Kingdom or UK? West Virginia or WV? Or W.Va.? Kiev or Kyiv? With CDS, we create the data set once, and then make that data available across apps. Repeat the same trick with your employees, customers, products, and more. We create a single source of truth by separating the data source (CDS) and then building the transactional layer with which users interact (Power Apps) on top. Then we can secure it and automate its powerful built-in tools resident inside of CDS.

It's all about the data.

Though Power Platform integrates well with Microsoft technologies such as SharePoint and Azure data storage options, the Common Data Service is Power Platform's native data source. While there are use cases that make it appropriate to store data outside of CDS, it's best to think of Power Platform with CDS as the best choice for relational data and business process automation.

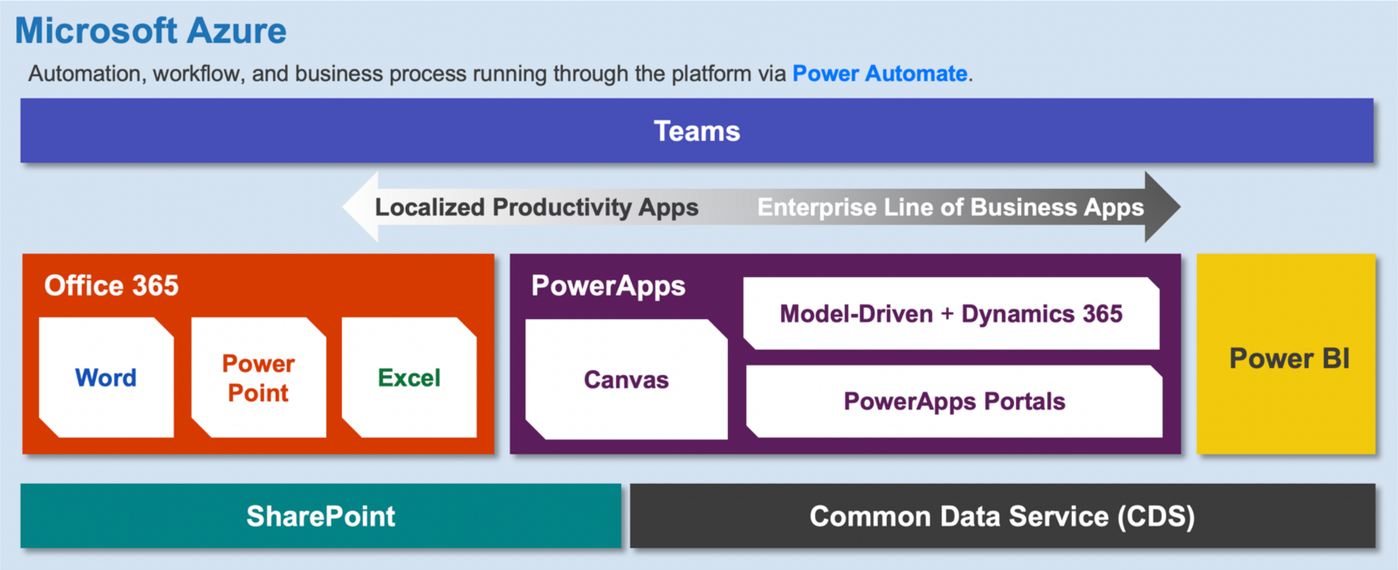
Then we point Power BI at it. Power Platform's answer to rich, visually compelling data visualization and business intelligence, Power BI consume all the data stored in CDS and then integrates with data found across hundreds of third-party services via out of the box (or custom, if you need) connectors. The result is charts, dashboards, and other visualizations that are as aesthetically stunning as they are rich in content.

Those data connectors plug seamlessly back into Power Apps. This means we can easily and quickly mash up third-party data found anywhere from Facebook to Salesforce and beyond in both our visualization layer (Power BI) and our transactional layer (Power Apps).

Finally, Microsoft Power Automate serves as the connective thread that automates and drives business process throughout the platform and in integration with Office 365 and those hundreds of other data connectors.

Power Platform in Broader Context

Power Platform sits within a broader context as one of Microsoft’s “three clouds”. As organizations increasingly seek a wholesale path to modernization in the cloud, it is becoming likewise important to conceptualize Power Platform in conjunction with the rest of the Microsoft stack.


Microsoft Azure
Automation, workflow, and business process running through the platform via Power Automate.
Teams
Localized Productivity Apps
Enterprise Line of Business Apps
Office 365
Word
Power Point
Excel
PowerApps
Model-Driven + Dynamics 365
Canvas
PowerApps Portals
Power BI
SharePoint
Common Data Service (CDS)

Power Platform sits to the right of the diagram. It's important to note here the different forms that Power Apps can take.

- **Canvas Apps,** which are the (generally) mobile applications built from the user interface level (e.g. "blank canvas", visually speaking)
- **Model-Driven Apps**, which are apps built directly from the data model in CDS and rendered via a pre-defined user interface; Dynamics 365 applications are "model-driven"
- **Power Apps Portals**, which are portal websites that surface data from inside of CDS to internal or external users

Though it is native to the platform, CDS is not the only source of data from which Power Apps may be built. To the left of the chart we see that SharePoint lists may contain data atop which canvas apps are built. This is often the point of entry for citizen developers and less complicated use cases. It's well known what else lives in SharePoint (Office 365 data).

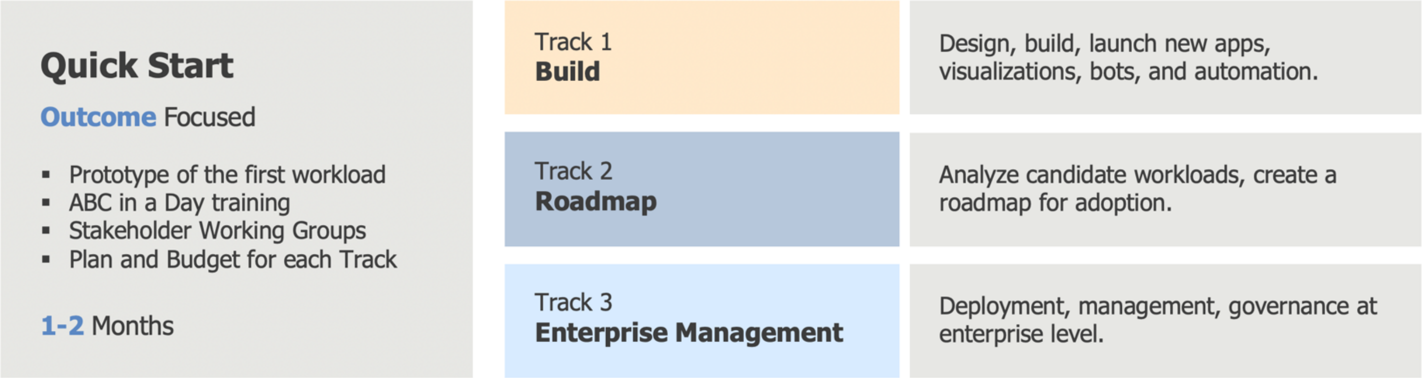
This sets us up for a nice spectrum (shown in grey) whereby "localized productivity" canvas apps meet lighter-weight, less complicated use cases and are often built by citizen developers using SharePoint (or even Excel) as a basic data repository. The workloads become more complex as we progress to the right of the spectrum using CDS for data and ultimately employing a combination of canvas, model-driven, and portal apps.

Returning to Office 365 for a moment we see that all of this can be surfaced and delivered to users via Teams (or as independently run applications, of course). Automation, workflow, and business process run through the entire stack via Power Automate.

Finally, because the entirety of the platform sits inside of Microsoft Azure, we're able to extent the functionality of our solutions to utilize the services available in Azure (including, as we'll discuss later, the ability to build apps directly from data in Azure).

Adopting at Scale

The Power Platform Adoption Framework provides a holistic approach to adopting the platform at scale, so organizations realize the advantages to scale, discussed previously. It is an alternative to a haphazard implementation, where implementing capabilities based on the need of the moment results in a barrage of half-baked environments and poorly conceived enterprise governance. It takes us from thinking in terms of a single app that's needed right now, to thinking in terms of a platform with all its accordant network effects, standardization, and return on investment.



Quick Start

Outcome Focused

- Prototype of the first workload
- ABC in a Day training
- Stakeholder Working Groups
- Plan and Budget for each Track

1-2 Months

Track 1 **Build**	Design, build, launch new apps, visualizations, bots, and automation.
Track 2 **Roadmap**	Analyze candidate workloads, create a roadmap for adoption.
Track 3 **Enterprise Management**	Deployment, management, governance at enterprise level.

Quick Start

We begin the road to adoption with a Quick Start period. The goal is to get to value and plan for a time-boxed period of about one to two months. Any longer and we're dragging our users into busyness with limited results and possibly risking our credibility and trying their patience. Power Platform promises to get the capability into users' hands quickly, and as experts in the platform, it's our job to make good on that promise. Quick Start has four primary outcomes or deliverables.

Prototype of the first workload (e.g. your first combination of apps, BI components, automations, and bots working together to solve for a use case). This is about getting to value quickly. Here we're selecting a high-priority workload, co-designing alongside business users and gleaning their expertise in how the app should work, and then building it out as a prototype. Consider:

- Choose a workload that is important in its meaning and impact to the organization, but not so mission critical that users and business leadership are terrified of migrating it onto a new platform. The idea here is to demonstrate real value to the organization in a short period.
- A workload that only matters to a handful of users is a poor choice for that task. Don't expect perfection in the prototype. We time-box its development explicitly to avoid feature creep. This definition of prototype here is whatever we can get working on the platform in about six weeks. There will be time after the Quick Start to get it to full operating capability.

ABC in a Day Training. The task is to get prospective citizen developers and power users proficient in using the platform to build useful things. Microsoft's App in a Day and Dashboard in a Day workshops are useful here, as might more in-depth courses be. However, the objective is to get the most enthusiastic and promising users up to speed quickly. Don't expect expertise at this stage.

Stakeholder Working Groups. While we're learning about what Power Platform should look like in the organization through all pieces of the Quick Start, the stakeholder work groups are where those ideas come together. These are facilitated sessions that explore the current and needed state of IT infrastructure to support the platform, technical governance, and organizational change required to make the technology successful across the enterprise. We will review the status of related technologies used within the organization (e.g. Dynamics 365, legacy InfoPath or SharePoint apps, or even third-party services, such as Salesforce), and other hot topics uncovered through Quick Start. These work groups are most productive when facilitated by technologists with expertise in Power Platform and group facilitation in business. Consider working with a Microsoft partner for these facilitated sessions to realize value expeditiously.

Plan and Budget for each Track. We've learned a lot through Quick Start. The prototype has helped us get a feel for the business use cases we'll be migrating to the platform; training exposed us to the skill level and enthusiasm of our prospective citizen developers, and the stakeholder work groups have given us the chance to paint the picture of platform adoption. Lastly, we will create a project plan, timeline, and budget for executing the adoption along three parallel tracks. Here, it's best to think about a year into the future. In other words, what do we need to do to build or migrate workloads, roadmap future workloads, and mature our enterprise management and governance of the platform over the next twelve months?

Note that previous iterations of the framework referred to "Quick Start" as the "Discovery Phase". We felt that this was insufficient to drive home the idea of getting to value quickly. Some obsolete diagrams may continue to use "Discovery".

On Track

Three big buckets of work follow the Quick Start. We prefer to think of them as tracks as they should happen in parallel to one another as a coherent journey, for as long as the organization uses Power Platform.

Track 1: Build. The first track of work focuses on designing, building, and launching new workloads onto the platform as components in Power Apps, Power BI, Power Automate, or Power Virtual Agents. Avoid falling into the trap of thinking Power Platform just about a single capability (e.g. "building canvas apps in Power Apps"). Rather, when we build, we're knitting together myriad capabilities into comprehensive solutions. "Build" is about creating apps, BI components, flows, bots, AI models, PCF components, dataflows or custom connectors / integrations where necessary, and even apps for use inside of Microsoft Teams (and probably lot more).

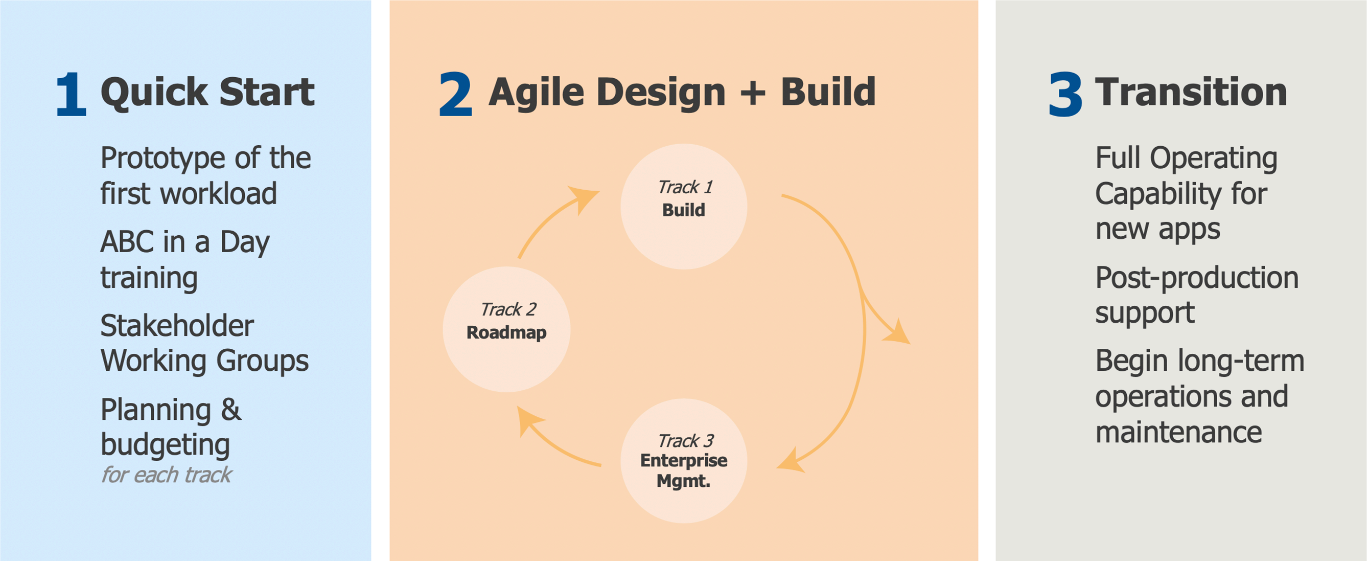
Workloads are the comprehensive things we migrate to or build anew on the platform. The framework really seeks to avoid overusing the word "app" for a few reasons. First, Power Platform is about so much more than just "apps" (see above). Second, new technology and better ways of doing things mean that we may not find a 1:1 match between the solution we had and the solution we're now building on the platform. For example, in the past you may have crammed five different (mostly) unrelated use cases into the same Access database or SharePoint app because there wasn't a practical way to share data between them. But now, with Power Platform, we can migrate that data into CDS and create five purpose-built apps atop it. We can also automate some of the manual effort previously involved in those use cases and visualize that data in Power BI. Power Platform is more than building apps... it's about building entire "workloads".

Track 2: Roadmap. The second track feeds the first. Power Platform is well suited for many types of workloads — large and small, simple and complex, enterprise-focused and small scale. The Power Platform enables the modernization of legacy technologies, ranging from do-it-yourself spreadsheets and Access databases to tangled enterprise deployments of InfoPath and SharePoint apps and enterprise-grade systems such as CRM and ERP. Therefore, it's critical that we directly work with business owners, and users alike, to identify and analyze candidate workloads and to create a roadmap for adoption. These candidate workloads should be prioritized and re-prioritized on an on-going basis so that our Track 1 activities remain focused on the most important next steps.

Track 3: Enterprise Management. Our third and final track is the foundation for the other two. Enterprise Management, which we will discuss in greater depth, is about the deployment, management, and governance of the platform at an enterprise level. It can be tempting to ask, “why do we need enterprise management around no-code/low-code technology?” The answer is that enterprise management establishes guard rails for the platform within the enterprise so developers can build beautiful and useful things with the confidence that what was built will be managed, secured, and otherwise looked after in a compliant way, following best practices. Enterprise Management is about creating the environment—both technically and culturally speaking—in which your users can thrive.

Going through a Phase

It’s useful to think about the timeline and the lifecycle for Power Platform adoption as a series of three phases.



1 Quick Start

Prototype of the first workload

ABC in a Day training

Stakeholder Working Groups

Planning & budgeting *for each track*

2 Agile Design + Build

Track 1 **Build**

Track 2 **Roadmap**

Track 3 **Enterprise Mgmt.**

3 Transition

Full Operating Capability for new apps

Post-production support

Begin long-term operations and maintenance

Quick Start. We covered the Quick Start in more depth in a previous section, but suffice it to say, this is the quick start on the platform, where the goal is to get to value and plan in a time-boxed period of about one to two months. The Power Platform Adoption Framework begins here.

Agile Design-Build. Work on the three tracks takes place in an agile fashion, which makes the Agile Design-Build phase different from Quick Start as it is ongoing throughout the life of the platform inside the organization. We consistently build new capabilities in the form of PowerApps, Power BI visualizations, and automation in Power Automate (Track 1). We are always laying out the roadmap for candidate workloads for build out and automation on the platform (Track 2). We continue to build and nurture the management of the platform at an enterprise level (Track 3), either in terms of incorporating advances in the technology or in nurturing the user community and citizen developers within the organization.

Transition. When an app, visualization, or automation component is ready for deployment to production and general use within the organization, it’s important to transition that workload out of its design-build development and into its long-term home. Many IT organizations refer to this as “Operations and Maintenance” (O&M), though regardless of the naming, the idea is that apps transition from a mode where they are actively built to a mode where they are actively supported for end users. This includes what we traditionally think of as “GoLive” or “cutover”, but also includes post-GoLive support, tuning and optimization, and—importantly—the answer to the ever-persistent question in software: “Who does the user call if this thing breaks in the middle of the night?”

Enterprise Management

Let's spend some time thinking about enterprise management, our Track 3 in the Power Platform Adoption Framework.

Earlier we said that enterprise management is about establishing guard rails for the platform within the enterprise so developers can build beautiful and useful things with confidence that what they build will be managed, secured, and otherwise looked after in a compliant way following best practices.

Pillars of Enterprise Management

So, what are those guard rails? Our framework proposes five big categories of items that need to be accounted for when implementing and maintaining enterprise management of the Power Platform.

Platform Management. Which tools do we use to manage development, backlog, road map, and the platform itself? Will we deploy the CoE Starter Kit (yes!)? If so, do we intend to further customize it? What must we do organizationally in order to establish and maintain the Center of Excellence itself? How do we organize to use and expand the platform across the enterprise effectively? Here we're talking about organizational relationships (i.e., who "owns" the platform at an enterprise level, how the platform is funded, and where will that responsibility / accountability / authority live within the organization).

Enterprise Architecture. How do we set up Power Platform environments within the tenant, and use that architecture in order to support other pillars such as ALM and maturing our security model? How do we license and authenticate users? Which re-usable components shall we supply to developers (and how shall we supply them)? Are there other data sources in the organization's overall data ecosystem (e.g. data warehouse, Azure data services, etc.) for which we need to account and possible integrate? Here we're primarily concerned with how the platform fits within the larger IT picture, from both a technical and governance perspective.

Application Lifecycle Management. Who develops new and migrates existing workloads on the platform, i.e., citizen developers, Power Platform developers on the in-house IT staff or CoE, or with contracted support from a Microsoft partner (usually some combination of the three)? Which development best practices do we employ? How do we configure and manage the deployment pipeline and source control? Are we leveraging DevOps and CI/CD practices and tools? What does our configuration management look like, for example, how do we manage our enterprise-level data model to ensure consistency and integrity of our data? Here we're concerned with the actual nuts-and-bolts of building apps, visualizations, and automations.

Mature Security Model. Are there security accreditation measures to which we must adhere? In the early days of platform adoption, we must take care to begin the work of adhering to these measures early enough in the process so that administrative matters don't delay the deployment of our first workloads. How do we employ platform security within apps? Across apps? In the Common Data Service? What does our data loss prevention for data connectors look like? How are we configuring policies within the tenant to actualize our security model? Here we're concerned with securing our data, apps, and users from a technical and a policy/administrative perspective.

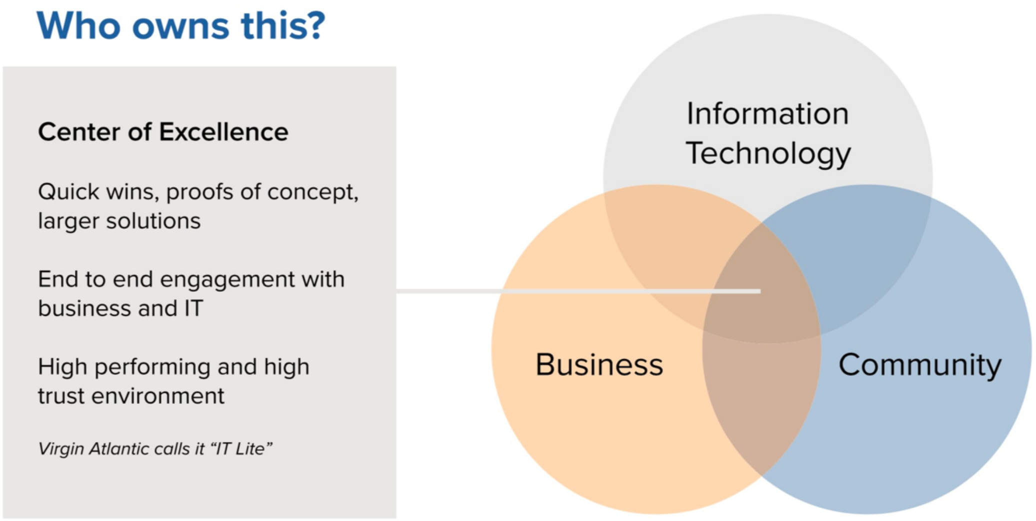
User Empowerment. Adopting Power Platform necessitates that user support go far beyond help desk services. How do we provide on-boarding and community management to our power users, citizen developers, and business owners? How do we grow and enable a vibrant user group community within the organization? Here we're concerned with empowering users to achieve their best work, so they can take advantage of the no-code/low-code tools the platform offers. This is about a transitioning from a legacy model in which IT was the "giver of things" to a forward-thinking, user centric model in which IT is the "enabler of people".

Who Owns This?

This is one of the most important questions that an organization must answer at or near the start of its platform adoption journey. Though IT organizations have traditionally “owned” new technology adoptions and long-term sustainment, Power Platform’s no-code/low-code tools make it appealing to those working in operational or functional roles within the organization (i.e., “business users” rather than “IT”). This shift in thinking presents the question of ownership, specifically who in the organization should be responsible for driving the adoption and managing the platform long-term.

Best practice that first emerged amongst many early adopter organizations, but which is now standard both for Microsoft and for the adoption framework is that of a Power Platform “Center of Excellence” (CoE). The airline Virgin Atlantic, a noted and successful Power Platform early adopter, has named their CoE “IT Lite”. Whatever you call it, this core group of platform “owners” must possess several key characteristics.

- Be capable of developing quick wins, proofs of concept, and larger solutions for “customers” around the business.
- Build durable partnerships through end-to-end engagement with business owners and the IT organization.
- Create a high-performing, high-trust culture; the CoE will touch many areas of the organization and must be a trusted actor to drive adoption of new technology.



Who owns this?

Center of Excellence

Quick wins, proofs of concept, larger solutions

End to end engagement with business and IT

High performing and high trust environment

Virgin Atlantic calls it “IT Lite”

Where this group lives on the organizational chart is to some extent a soft matter best left to the cultural characteristics of the larger organization itself. Some may find it fits naturally within an existing IT culture, while others may prefer to house the CoE somewhere within the “business” (e.g., inside of an early-adopter operating unit, or a business transformation/innovation/strategic initiatives group). In any case, this decision should be made early in the adoption, and the CoE must be empowered with authority to act on the platform’s behalf and a budget to support the adoption.

There is significant risk in combining the duties of the Power Platform CoE with those of the organization’s Office 365 team. Often the divergent responsibilities of growing an application platform and supporting productivity functionality lead combined teams to perform poorly in one or both functions. Therefor the recommended best practice is to clearly divide responsibility here whilst creating the expectation that both teams will work together closely.

Architecting for Scale

Enterprise management and governance of the platform cannot be effective if we've not architected the platform for scale within the organization.

Center of Excellence Starter Kit

Microsoft has created the "Center of Excellence Starter Kit" in order to provide the CoE (and IT management broadly) with a robust set of tools to manage and govern Power Platform within an organization's 365 tenant. The kit is itself a combination of Power Apps, Power Automate, and Power BI components. It is to be deployed into a Power Platform environment provisioned with CDS, so thus requires that the deploying organization have obtained some number of premium Power Apps licenses in order to manage the platform more broadly. This is the case even if the organization plans to focus overall Power Platform use on "productivity" applications (see "Environmental Architecture Model" below) available via capabilities in Microsoft 365 licenses only.

The CoE Starter Kit components are available to all on GitHub. GitHub provides the authoritative source of information concerning the toolkit, so will not be "reprinted" here in the Power Platform Adoption Framework. However, the introduction below is current from that Microsoft-provided material as of 31 March 2020.

> “The **Center of Excellence (CoE) Starter Kit** is a set of templates that are designed to help develop a strategy for adopting, maintaining and supporting the Power Platform, with a focus on Power Apps and Power Automate. The kit includes multiple Power Apps and Power BI analytics reports to view and interact with the data collected. The kit also provides several assets that provide templates and suggested patterns and practices for implementing CoE efforts. The assets part of the CoE Starter Kit should be seen as a template from which you inherit your individual solution or can serve as inspiration for implementing your own apps and flows.”

The following additional best practices apply to the CoE Starter Kit.

CoE Starter Kit ≠ Center of Excellence. Power Platform must be secured in any case, even if the organization has chosen to not embrace the Center of Excellence Model (and therefor not create a CoE). The adoption framework strongly recommends against this, but it is important distinction in saying that the CoE Starter Kit should be deployed in order to provide IT the tools to manage and govern the platform regardless of whether a CoE has been established organizationally.

Starter Kit is just the start. Conversely, installing the CoE Starter Kit does not mean that you have established a fully functioning CoE. The most important word here is, perhaps, "starter". In other words, the kit provides a starting point and a toolset for beginning to manage and govern the platform, but absent configuration / customization in order to meet the needs of the organization, as well as human processes to govern the platform, the kit itself will be insufficient. Securing the platform / installing the "Starter Kit" does not mean that you've done what's necessary to adopt, manage, and govern the platform.

Make the CoE Starter Kit part of your ALM. The starter kit is deployed as a managed solution. Many organizations find it necessary to further customize the kit's components in order to meet their specific needs. It is therefor wise to build proper ALM in the form of a deployment pipeline around the kit. This means establishing environments for "CoE DEV", "CoE TEST", and "CoE PROD". The CoE Starter Kit solution should be deployed to each as a managed solution. A "CoE Customizations" solution should then be established as unmanaged in DEV; customizations to the starter kit should be made here. Those customizations should then be pushed as a managed solution to TEST and then to PROD. It is feasible that the CoE solutions could be pushed to the organization's "Critical" PROD environment (see "Environmental Architecture Model" below) if a dedicated CoE PROD environment were not desired.

Environmental Architecture Model

Organizations whose Power Platform footprint has grown quite large sometimes report struggling with a workable model architecting the platform across multiple environments. Questions such as “How do we ensure mission-critical workloads are not disturbed by citizen developers” and “Should an environment be owned by central IT or business units' themselves” are common.

These challenges reinforce the importance of the “Enterprise Architecture” pillar in general and the importance of sound environmental planning from the very start of the adoption.

It is important to segment all workloads by criticality. In other words, all production workloads across the organization should be categorized as either Productivity, Important, or Critical as shown below.

Productivity

- Localized productivity apps
- Do not require high level of governance
- Minimal support needs

Support by the app developer:

- Typically built by a citizen developer
- Often no review or approval process
- Typically <2 weeks for first iteration
- High level of iteration

Important

- Important but not critical
- Medium complexity
- Scoped to business unit

Blessed by IT:

- IT blessing at app or user level
- App owner often first level of support
- ALM using environments is advised
- Development cycle typically ~3 months to pilot

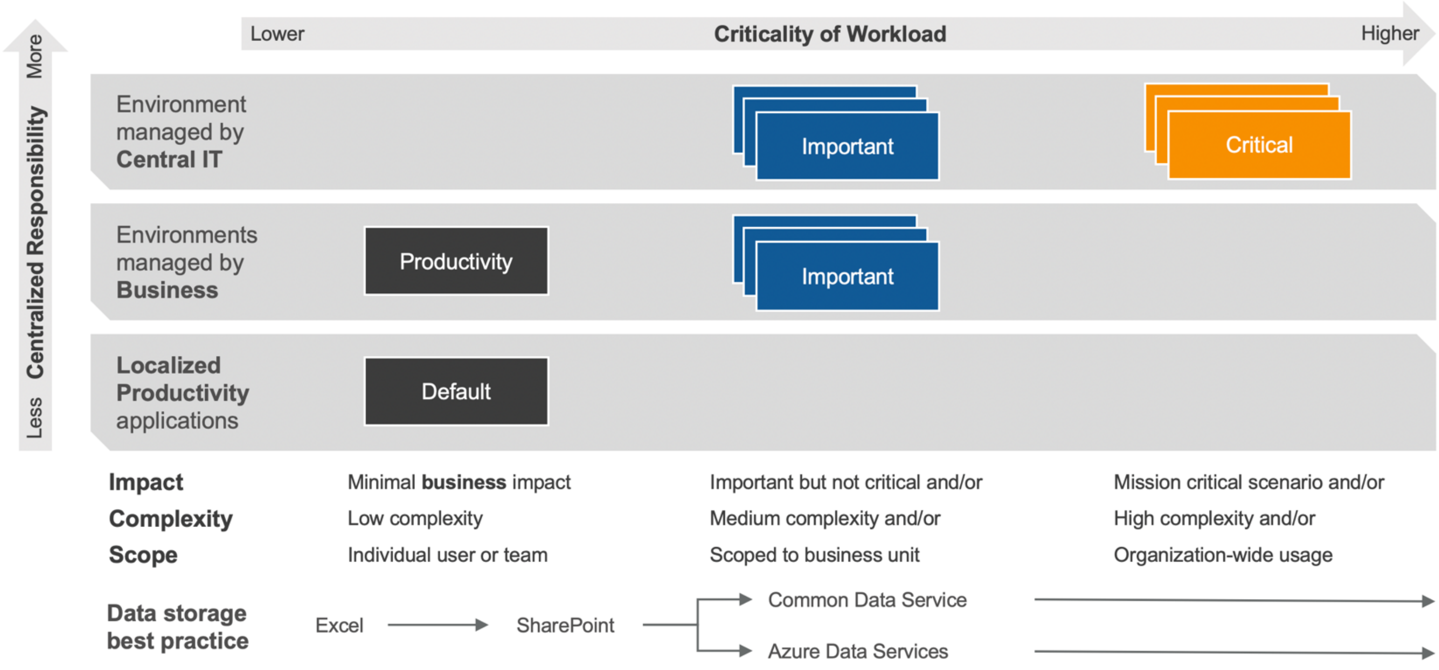
Critical

- Mission critical scenario
- High complexity
- Organization-wide use

Support by IT:

- Robust ALM process via DEV, TEST, and PROD environments at minimum
- Longer deployment cycles

From there we can apply the Adoption Framework’s Environmental Architecture Model shown in the diagram below.



Lower
Criticality of Workload
Higher
More
Centralized Responsibility
Less
Environment managed by **Central IT**
Important
Critical
Environments managed by **Business**
Productivity
Important
Localized Productivity applications
Default

Impact	Minimal **business** impact	Important but not critical and/or	Mission critical scenario and/or
Complexity	Low complexity	Medium complexity and/or	High complexity and/or
Scope	Individual user or team	Scoped to business unit	Organization-wide usage

Data storage best practice
Excel
SharePoint
Common Data Service
Azure Data Services

Here we have “Centralized Responsibility” (less and more) running along the vertical axis. There are generally three levels of centralization of responsibility for an environment and the workloads that it contains:

- **Central IT.** These environments are proactively managed by a central IT organization. The definition of “central” might vary depending on the organization (e.g. quasi-independent subsidiaries may have their own “Central IT”), but the fundamental idea here is that these environments are wholly owned by an element of the organization focused on IT.
- **Business.** These environments are owned by the business units within the organization, though they are to be proactively managed by IT personnel embedded in those business units. In these scenarios, the central IT organization may have provisioned the environment or provided guidelines to business units in managing that environment, but the central premise is that them management and support of that environment is fundamentally the responsibility of the owning business unit.
- **Localized Productivity.** These environments are generally highly unmanaged, provisioned essentially for the purpose of providing citizen developers a place to deploy apps categorized as “Productivity”. Many organizations choose to use the Power Platform “Default” environment for this purpose, often re-naming that environment as “Productivity” or “Local Productivity”.

Beneath the tiers of responsibility we have the criteria of impact, complexity, and scope that we established above in segmenting workloads by criticality.

”Criticality of Workload” (lower to higher) runs along the horizontal axis. Environments should then be created in order to house apps that align the critical, important, and productivity categorization to ownership as shown in the diagram. Note that the three-stack of environments indicates that best practice here is establish application lifecycle management best practices (e.g. DEV > TEST > PROD)—using automation where possible—and to not ever commit changes directly in production.

Finally, the Environmental Architecture Model suggests best practice for data storage along the spectrum of criticality:

- Excel should be used sparingly, if ever, and only to provide data storage for the most basic and temporary workloads.
- SharePoint is an acceptable best practice data storage for productivity workloads built by citizen developers but should be avoided for more advanced productivity and certainly for important and critical workloads.
- Common Data Service is the data source native to Power Platform and should be considered the default data source for important, critical, and some more sophisticated productivity workloads.
- Azure data services can provide an alternative to CDS in organizations with significant pre-existing Azure data investments / infrastructure. However, significant functionality—particularly within Power Apps (e.g. the ability to create model-driven apps)—is sacrificed when using Azure as a data source.

Microsoft has addressed this topic with additional guidance in a blog post of 30 October 2019: Establishing an Environment Strategy for Microsoft Power Platform.

Nurturing the Community

Let's consider the community for a moment as it is the user group community—particularly of citizen developers and power users—within an organization that makes Power Platform so compelling.

Shared Developer Community

Past convention has been to categorize individuals creating functionality on Power Platform as either "citizen developers" or "pro developers". While best practice is to embrace the citizen developer characterization, the adoption framework likewise recommends against the use of the pro developer characterization (citizen developers are "professionals" at what they do, too). "Power Platform developer" is the recommended characterization of those whose full-time job is to create functionality on the platform.

Citizen Developers

Citizen developers build components on Power Platform in service of their primary non-IT job function. They are not working on the platform or in IT full time. The applications they develop are generally characterized as:

- Personal productivity
- Team use
- Specific to the citizen developer's business function

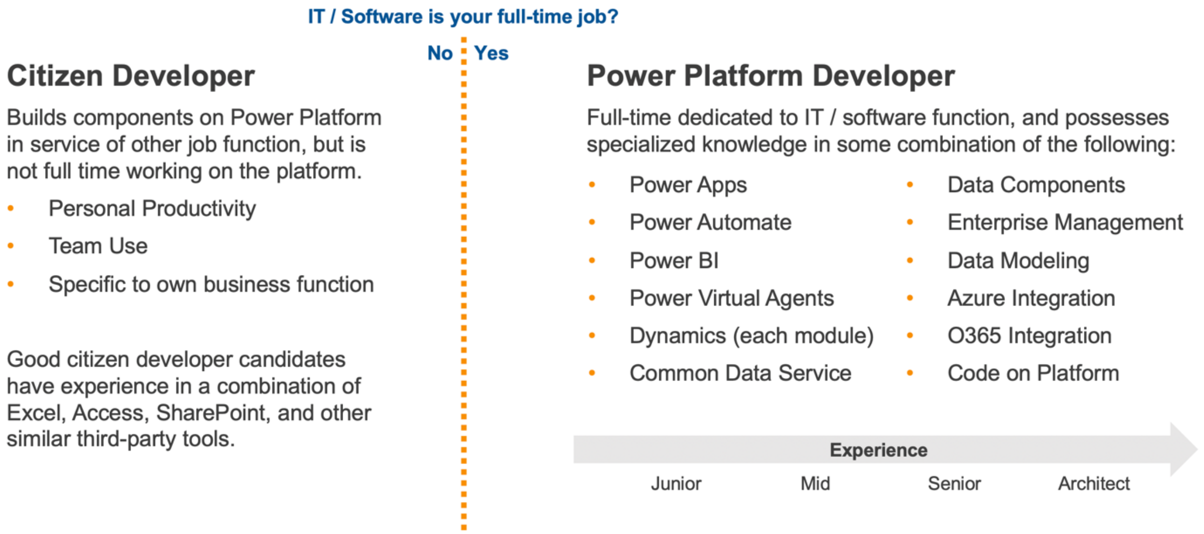
Citizen developers often come to the platform because there are power users or have experience in a combination of Excel, Access, SharePoint, and other similar third-party tools.

The real dividing line here is the question of whether IT / software is the developer full time job. If it is not, that developer can generally be understood as "citizen developer". Those whose full-time job is in IT or software are not citizen developers.

Power Platform Developers

"Power Platform developers" are those whose full-time job focuses on creating functionality on the platform. They may have come from a citizen developer or from a software engineering background.

The platform has become too large for any single person to reasonably possess expertise across the entire suite of technologies, so Power Platform developers possess specialized knowledge in some combination of the areas shown in the diagram below. They also possess varying degrees on a continuum of experience ranging from "junior" to "architect".



IT / Software is your full-time job?

No | Yes

Citizen Developer

Builds components on Power Platform in service of other job function, but is not full time working on the platform.

- Personal Productivity
- Team Use
- Specific to own business function

Good citizen developer candidates have experience in a combination of Excel, Access, SharePoint, and other similar third-party tools.

Power Platform Developer

Full-time dedicated to IT / software function, and possesses specialized knowledge in some combination of the following:

- Power Apps
- Power Automate
- Power BI
- Power Virtual Agents
- Dynamics (each module)
- Common Data Service
- Data Components
- Enterprise Management
- Data Modeling
- Azure Integration
- O365 Integration
- Code on Platform

Durable Partnerships

When we talk about durable partnerships, we're distinguishing between many traditional software development projects that often-treated partnerships between business units in a purely transactional way, versus the importance of creating partnerships around Power Platform that transcend a single app development effort.

In other words, a big part of the CoE's job is to build and nurture partnerships between business, IT, and the citizen developer user community on an ongoing basis rather than an ad hoc single-app-focused basis.

Best practices have emerged as to specific actions that may be undertaken to nurture these durable partnerships, and we expect that the global community around the platform will continue to grow and refine these in the future.

Supporting IT

- Use administrator dashboards, flows, and platform administrative data connectors to automate platform management and provide insights into platform usage
- Establish single archive (e.g., SharePoint) for technical white papers, development standards, transition/hand over documents, and other administrative material
- Keep abreast of technology advances, sharing updates and new features
- Host learning lunches and show and tell sessions to evangelize solutions within the organization
- Provide technical expertise to support Tier 3 help desk requests that cannot be solved via lower tiers of the IT help desk

Supporting Business Units

- Host drop-in hours to talk about business problems and technical challenges; these need not begin with Power Platform as the solution
- Facilitate road mapping (identification, analysis, and prioritization, i.e., "Track 2") of candidate workloads for migration to the platform
- Lead co-design sessions wherein technical and business subject matter experts design PowerApps, Power BI, and Flow solutions to business-driven use cases
- Organize hackathons to solve business use cases in short, focused periods
- Keep abreast of technology advances, sharing updates and new features
- Host learning lunches and show and tell sessions to evangelize solutions within the organization

Supporting the Community

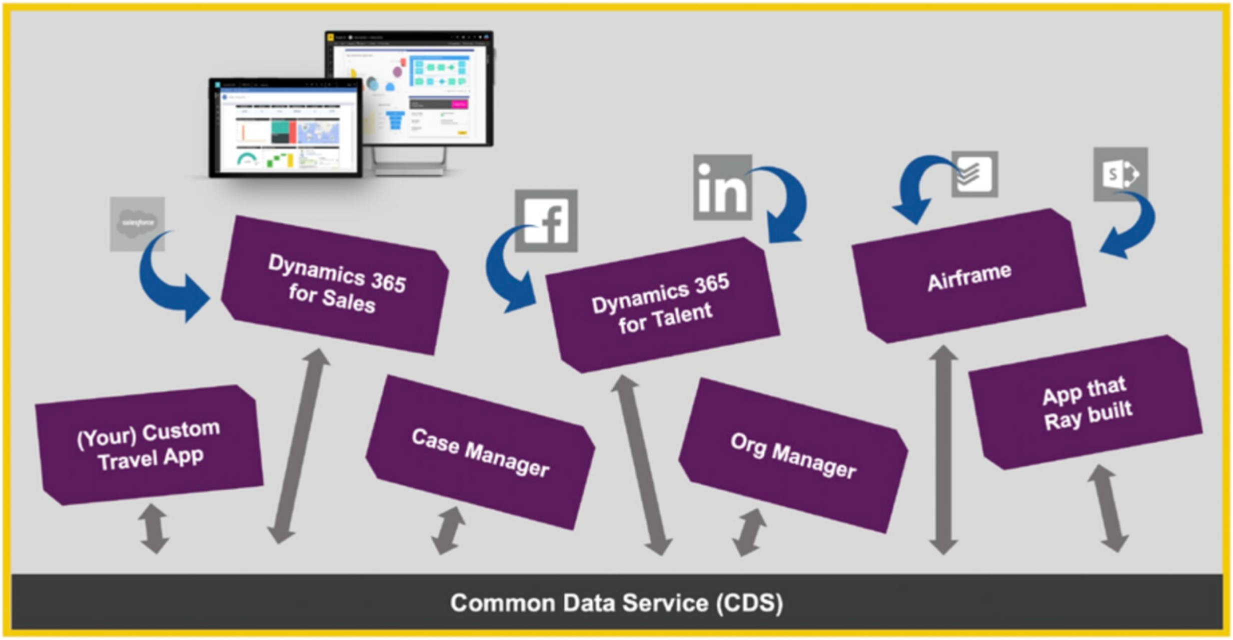
- Offer structured citizen developer onboarding training that provides newcomers with the baseline knowledge they need to be successful creators of no-code solutions within the organization; this may take the form of Microsoft's App in a Day/Dashboard in a Day/Flow in a Day classes
- Develop re-usable components and templates that save time for citizen developers, and create technical and/or stylistic consistency throughout the organization
- Automate routine project management tasks, for example, if there are a series of things that every citizen developer must do before deploying a new app, automate the creation of those tasks in the new app creator's Outlook task list
- Curate success stories of successful citizen creations, and encourage community growth and collaboration by sharing these throughout the organization
- Use Microsoft Teams to create a single communication, collaboration, and knowledge sharing hub for citizen developers across business units
- Establish learning paths for specific topics and disciplines through which citizen developers can further develop their skills

Perhaps most importantly, establish mentoring and coaching relationships amongst technical experts in the CoE and the subject matter experts and citizen developers out in the business units. Citizen developers will be most successful (and their work will pose less risk being deployed to production) when they are mentored by technical experts when they are constantly able to reach back to solution architects and platform specialists to help them through thornier issues such as data modeling. Do not expect citizen developers to be technology experts; they are doing dabbling with the technology in service of their "day jobs", not the other way around.

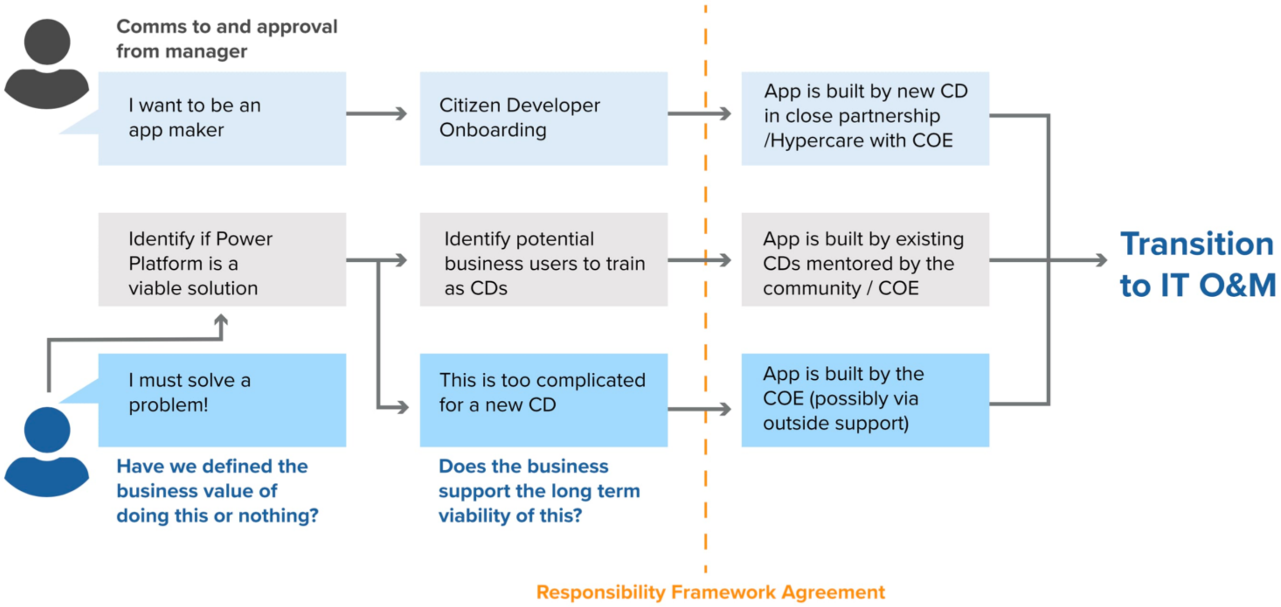
The Big Idea

So let's knit all of this together.

The Power Platform Adoption Framework offers a best practice-driven approach to adopting Power Platform at scale. Scale is valuable because of the network effects, standardization, and return on investment it offers.



Ultimately, we're employing the framework to build and sustain a vision of Power Platform that looks something like the diagram above: A single source of truth for our data in the Common Data Service and apps and flows that automate and facilitate a myriad of different workloads. These workloads are built atop that single source of truth, connected to third-party services where necessary, and wrapped in Power BI. They are built to surface data through richly interactive data visualizations and managed at an enterprise scale to reduce risk and enable citizen developers to do as much of this work as possible.


Comms to and approval from manager
I want to be an app maker
Citizen Developer Onboarding
App is built by new CD in close partnership /Hypercare with COE
Identify if Power Platform is a viable solution
Identify potential business users to train as CDs
App is built by existing CDs mentored by the community / COE
Transition to IT O&M
I must solve a problem!
This is too complicated for a new CD
App is built by the COE (possibly via outside support)
Have we defined the business value of doing this or nothing?
Does the business support the long term viability of this?
Responsibility Framework Agreement

When all the pieces are working as intended, we've created a repeatable methodology for turning the enthusiasm of citizen developers and the challenges found in our business into working apps, visualizations, and automation.

Sometimes the impetus for **something new comes from a business user** who raises his or her hand and says "I wanna' be a citizen developer!" Mature organizations have the wherewithal to onboard those folks as citizen developers and channel their enthusiasm in the right direction.

Sometimes the impetus for **something new comes from a business owner** with a problem to solve. Mature organizations possess the discipline to determine if Power Platform is the viable solution to solve that challenge, either with help from citizen developers or with seasoned pros.

In either case, the organization's purposeful adoption of Power Platform, the definition of responsibilities, and a rigorous approach to enterprise management have positioned it to meet the business need in one of three ways:

- App is built by a new citizen developer in close partnership ("hypercare") with the CoE
- App is built by existing citizen developers mentored by the community / CoE
- App is built by the CoE, possibly via outside contract support

Ultimately, we transition whatever we build to its long-term home — IT O&M, or however the organization has decided to sustain these workloads into the future.

About the Publisher

AIS originated and publishes this global, community-driven, open source standard for adopting Power Platform at scale. We are the experts in adopting, managing, and governing the Power Platform in enterprise-grade organizations. We've worked with Microsoft on the Power Platform since its inception, and we've built a talented core team—including multiple recognized Microsoft Most Valuable Professionals (MVPs) for Business Applications. This team works globally, 24/7, and optimizes costs amongst talented experts worldwide. As a leader in taking our clients to the cloud with Azure and Microsoft 365, we're able to extend the Power Platform across the entirety of Microsoft technologies.

We work with large organizations across their entire adoption journey beginning with the 1-week Power Platform accelerator, getting to value quickly with a prototype solution built as part of Power Platform quick start, and reaching maturity via Power Platform adoption. The AIS team has provided development, road mapping, and enterprise management services to global organizations across the Fortune 500 and government.

With nearly 40 years of history, dedication, and deep understanding of Microsoft technologies, AIS brings experience and expertise in delivering business application platforms and solutions to the enterprise. We've spent decades helping commercial and government organizations use technology to transform the way people work to achieve desired business goals—from SharePoint to Office 365 to Azure, and now the Microsoft Power Platform.

www.ais.com

11440 Commerce Park Drive, Suite 600, Reston, VA 20191
Phone: (703) 860-7800
sales@ais.com