Microsoft Power Platform

Power Virtual Agents

Customer Engagement Playbook

Power CAT

Contents

About this playbook

The purpose of this document is to provide structure and guidance for customers & partners implementing a large chatbot project based on Power Virtual Agents (PVA). The document contains best practices and learnings from the delivery of many real-world Power Virtual Agents engagements with enterprise customers.

The document is aimed at job roles responsible for leading and delivering a complex project of this nature – e.g., Project Lead, Architect, Development Lead, and includes project approach recommendations and technical information.

The guidance in this document will help teams to mitigate risk, understand trade-offs, and measure success.

About Power Virtual Agents

This playbook assumes a basic knowledge of Power Virtual Agents, Microsoft's low-code solution to empower everyone to easily build intelligent chatbots. Please follow the link above for an overview of the service.

About Power CAT

This guide brings together real-world knowledge and lessons-learned from the Power Platform Customer Advisory Team (Power CAT). The Power CAT team, as part of the Microsoft product engineering team, works closely with enterprise customers to help them successfully adopt Power Platform services.

Engagement Principles

Adoption of low-code technologies like Power Virtual Agents can help customers envision and deliver quickly, with lower management and maintenance costs. Even with those benefits, it is beneficial to be clear on the reasons you are adopting this technology, and how it will show its value.

Customers approaching a Power Virtual Agents project should align project delivery with their own company culture and objectives. However, there are some core principles that we see adopted again and again amongst successful PVA projects:

Principles

Engage your flywheel

Many customers are surprised when their PVA bot goes live how quickly the "customer feedback flywheel" takes effect. Actionable customer feedback comes in almost instantly. Delaying for additional topics or processes means missing out on this valuable feedback. Encourage the team to see the go-live as the beginning of your project, not the finale.

Keep it simple

One of the core value propositions of Power Virtual Agents is iteratively delivering a conversational AI experience quickly. After the go-live by a project team, who is familiar with how the bot has been developed, maintainability is key to staying agile. While the tooling can allow users to create complex conversation flows, and extensibility with Bot Framework can achieve very flexible interactions, there should be careful consideration if complexity is always necessary.

Keep goals in focus

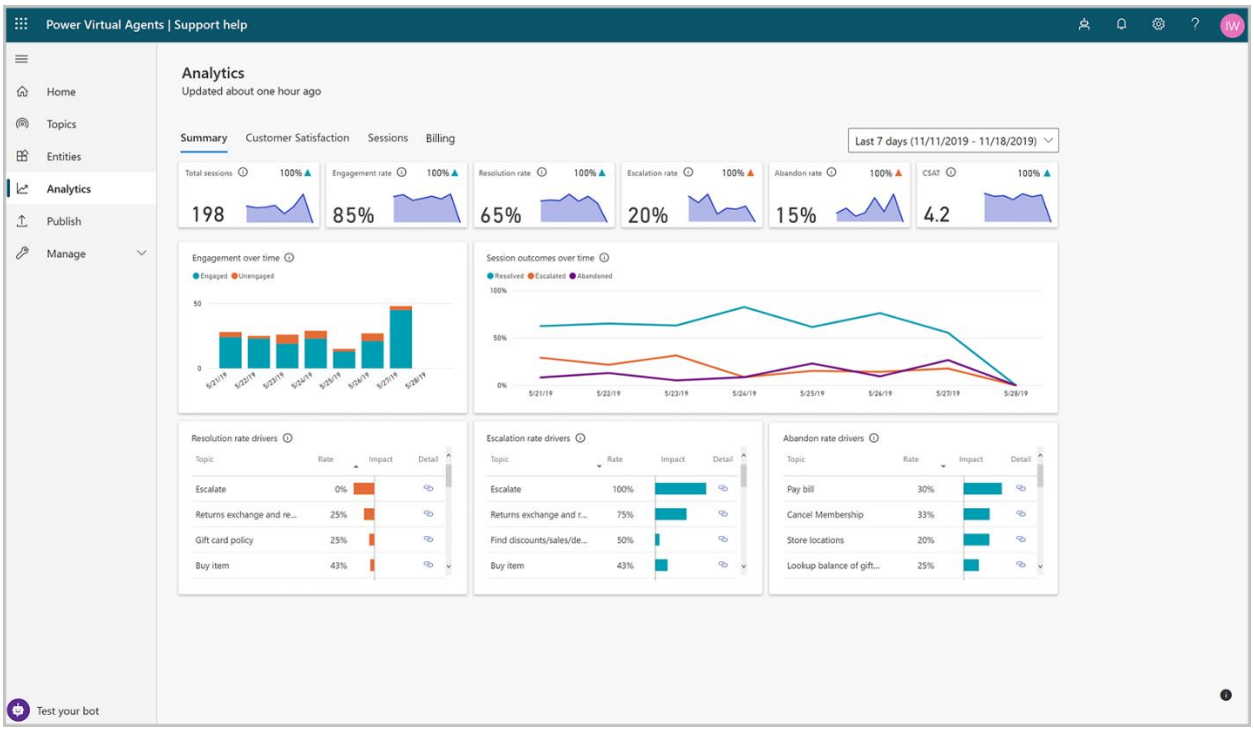
It's very easy when implementing new technology like a chatbot, to get wrapped in the minor details of implementation. Defining what your key business goals are (e.g., deflect 30% customer queries, score above 4.0 customer satisfaction), can help keep the project moving and ensure correct feature prioritization.

Setting Goals

We normally recommend customers separate the project goals around 3 distinct areas – core delivery, business metrics, and governance. If chatbots are brand-new to your organization, then the examples below are a good starting point. One clear trend we see with goals – once you go live you will learn very quickly what is important, how your customers behave, and where you would like to improve.

Core measurements

There are some key terms that relate to how Power Virtual Agents measures user behavior. It's useful to have a base understanding of them.



- **Session** – this is a user interacting with the chatbot
- **Engaged Session** – a user interacts with the chatbot, and activates a user topic (content created by the chatbot author – not one of the pre-created system topics)

Engaged sessions can have one of 3 possible outcomes – it is these outcomes that customers tend to measure once live:

- **Resolved session** – an engaged session where the user indicates that their question was answered
- **Abandoned session** - an engaged session where the user does not indicate that their question was answered
- **Escalated session** - an engaged session where the user escalates to a human agent

Importantly, resolved sessions + abandoned sessions + escalated sessions = 100% of engaged sessions. [More detail on PVA metrics + data dictionary].

Example project goals

Below is an example of goals a customer may define at project inception:

1. Delivery goals (Minimum Viable Project (MVP))
    a. Go-live with chatbot on <target> date - target less than 3 months
    b. Bot can answer top <target> use cases - using a script of test phrases
    c. <Target> complex use cases with back-end integration - this will help with deflection

2. Business goals (driven toward once live).
    a. Resolved session % - normally the project team aims to increase this number over time. This figure is often referred to as the 'Deflection' amount.

    b. Escalated session % - normally the project team aims to reduce this number over time.
    c. Abandoned session %
    d. Deflection/resolution – value (based on business value per session calculation)
    e. Monthly Active Users (often referred to as MAU)

3. Establish Chatbot governance and management model.
    a. Application Lifecycle process & automation setup
    b. Chatbot author onboarding process in place
    c. Reporting & review cadence arranged

Project Team Structure

Most customers treat a Chatbot implementation like any other project. It usually consists of the following roles:

Role Name	Accountability
Project Sponsor	The person(s) who benefits from the introduction of chatbot. Their role is to provide guidance and remove blockers.
Project Manager	The person who is accountable to track milestones, communicate status and mange risks and schedules.
Subject Matter Experts (SMEs)/Business Analysts	The people who are intimately familiar with business needs, often belonging to a business unit that will benefit from chatbot implementation. SMEs are accountable for content and alignment to corporate strategies. In addition, these people may participate in the development of the chatbot by creating topics and providing relevant trigger phrases.
Bot Developer	The people who may have low code/pro code experience in the Power Platform. Their role is to address any technical challenges that may include extensibility through Power Automate or Bot Framework Composer. They may also be responsible for ALM processes.
Power Platform Admin	The person(s) who is responsible for provisioning environment access and adherence to other Power Platform governance controls including Data-loss Prevention Policies (DLP).
Enterprise Architect	The person who will ensure that the solution aligns to existing enterprise architecture principals/standards as it pertains to security, compliance, and technology.

Stakeholders

Delivering a large bot is like any complex technology project – there are several stakeholders involved with different parts to play. This section outlines the roles and responsibilities to help align everyone in the project.

Microsoft

Microsoft delivers the Power Virtual Agents service:

1. Account team - Account Executive/Account Technical Strategist/Technical Specialist/ Customer Success Manager
    a. Alignment on success/ goals with customer
    b. Ultimate advocate for the customer within Microsoft
    c. Leads business discussions and escalations
    d. Attend all the rhythm of business calls until goals are met
2. Product group
    a. Develops product features based on customer needs
    b. Customer Advisory Team (CAT) – engages with select customers providing guidance on how to adopt and deploy technology including architecture reviews and monthly executive syncs
3. Microsoft Support – works with customers to solve issues with Microsoft software. Aims to solve issues directly but can also bring in engineering resources for challenging issues

3rd party Delivery partner (where applicable)

This is typically a System Integrator that leads various activities:

1. Project management including setting up workstreams and reviews with customer
2. Chatbot development – frequently includes developers/analysts with experience of Power Virtual Agents, chatbots and related technologies
3. Training
4. Administration/governance setup
5. Hackathons and other nurture events to drive adoption and knowledge sharing

Customer

Sets business goals, and may also have responsibility for project delivery:

1. Project manager – person assigned to ensure success of customer adoption, with visibility into resource utilization. Typically drives for clarity of requirements & creation of systems to measure progress against goals
2. Architect - the technical decision maker for the project who owns the overall technical design
3. Business lead - clarifies business requirements and success criteria. Helps short circuit decisions to help the team move forward quickly
4. Project sponsor – lead for the project success, typically a manager/ executive. Typically has periodic syncs with the working team to ensure progress is getting made

Sample Project Overview and Timelines

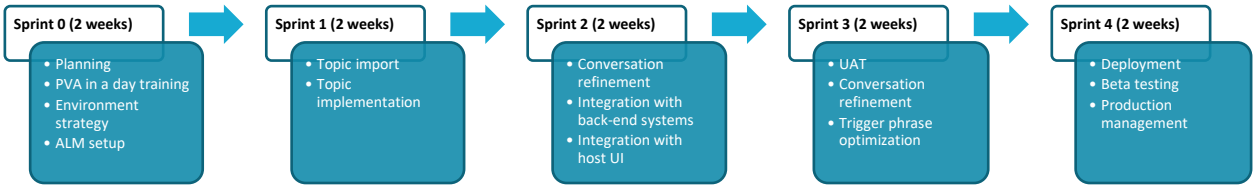


Diagram above shows an illustrative 10-week PVA bot development

Chatbot projects based on Power Virtual Agents, like any technology, can have wildly different timelines and specifications depending on the needs of the particular use case and organization. However, one advantage that low code technologies can bring, is rapid time to market as illustrated in the 10-week timeline above.

The advantages in choosing a rapid delivery timeline as your initial project phase include:

- Developing a minimum viable product to accelerate customer adoption
- Customer feedback is generated very quickly (see **Engage your flywheel** section) to provide insights on what can be improved
- Return value back to your organization in an accelerated manner

Whether this is right for your use case will depend on organizational need, support, and the ability to align the necessary workstreams.

Workstreams

Setting up an adoption platform requires several workstreams that can execute in parallel, each with their own goals, timelines, and owners. These are as follows, with summary callouts on responsibilities:

New bot development

a. This workstream focuses on building out the related content for the chatbot use case.
b. Content creation/development tasks

⏩ Please refer to the **Action Development** section for more details

Power Platform Administration

a. Many customers will already have Power Platform/M365 administration teams/CoEs that are accountable to the management of the technology set.
b. This team may focus on activities such as
    - Environment setup
    - Security/Access
    - DLP Policies
    - Gateways
    - Custom analytics

⏩Please refer to the **Environment Strategy** and **Custom Analytics Strategy** sections for more details

Training

For organizations that are new to building PVA chatbots, it is likely there are some training needs required for the customer to be familiar with the technology. The following resources can be leveraged to assist customers with building PVA chatbots.

Training can be delivered through various channels. These channels include Technical Sales Professionals (TSPs), Microsoft Technology Center (MTC) Architects and Microsoft certified partners. In exceptional circumstances, engineering resources like Power CAT may be able to facilitate these sessions.

a. PVA in a Day is an instructor led, hand's on course that will familiarize chatbot makers with Power Virtual Agents. The content includes:
    a. Overview and Prerequisites
    b. Build your first chatbot
    c. Topic Suggestions
    d. Improve chatbot with entities, variables, and topic redirect
    e. Topic Design Best practices

b. The PVA Architecture Series is another resource that can be consumed on-demand. This content focuses on considerations that customers should account for when embarking on a chatbot project. The content includes:
    a. Planning your first chatbot
    b. Building your first chatbot
    c. Extending your PVA chatbot with the Power Platform
    d. Extending your PVA chatbot with the Bot Framework

e. Advanced topics
f. Security and Governance.

Testing

User acceptance testing will likely take up a significant portion of delivering your PVA chatbot – ensuring not only the integration of the complete solution (with backends/handoff clients/APIs), but also of the chatbots responses to customer input. More information on testing, please refer to the testing strategy section.

Production Deployment/Readiness checklist

Ensuring that you are ready to deploy is a critical step in having a successful launch. Here are some considerations:

1. Have all scenarios been tested end to end to validate business requirements have been addressed?
2. Do you have a cutover plan that identifies step by step all the activities required to successfully go live?
3. Has this cutover plan been tested and validated?
4. Do you have access to all the production resources that will be needed?
    a. Environment
    b. DLP policies
    c. Gateways (if needed)
    d. Azure resources (if needed)
    e. External APIs (if needed)
5. Do you have a communication plan established to inform both internal and external stakeholders of the changes? This includes both pre and post deployment.

Rhythm of Business (ROB)

A Rhythm of Business (ROB) meeting allows all participants, involved in the project, to provide status updates and identify any blocking issues so that they can be get resolved by stakeholders. These meetings should be very actionable and driven by dates and business goals.

1. Customer driven – Microsoft/partner attend but customer is responsible for booking meetings and setting the agenda. The goal of this meeting is to run it as an engineering project, with success metrics and scorecards, and likely consists of:
    a. Standups for chatbot development, 1-2 times a week
    b. Weekly demos for progress
    c. Weekly status updates emails highlighting blockers
    d. Bi-weekly bug prioritization
    e. Internal nurture – celebrate success
    f. Periodic retrospectives
    g. Weekly office hours to support customers adoption
    h. Teams channel for open communication
2. Jointly owned (Microsoft/partner)
    a. Architectural reviews to guide core decisions for the platform
    b. Monthly exec reviews

c. Scorecard review with progress, risks, blockers, and feedback
d. Forum for demos as appropriate

Bot Development Lifecycle

Within this section, we are going to discuss the various stages of building out a conversational chatbot.

Requirements Gathering

Across various Power CAT engagements, we discovered a few common scenarios that involve requirements collection, including the following:

1. Topic generation can be done automatically using “Suggested topics” feature or manually using the conversation designer. The automated generation of topics is limited to single turn topics. User can generate the topics by pointing to public sites or PDF/Excel/Word docs that are hosted in a public site. While this process can accelerate building out the bot, it should be noted that generally, additional work will be needed to optimize trigger topics at minimum.

2. For manual authoring of topics, start with Identifying the key “Categories “or use case areas under which the Bot will have the intents/topics. For example, a Retail focused bot may have categories such as:
    a. Billing and Payments
    b. Orders and Shipping
    c. Account Management etc.

    Under each of these buckets you will be building out the individual topics and sub-topics.

3. Identify the list of topics for each category and list them under Topic Name, identify the next level of granularity of Topics as sub-topics and list them as subtopics. These will translate into actual topics in the PVA.

Column1	Use Case ID	Customer Statement/Question	Topic Name	Sub-Topic Name	MVP	Phase 2	Type (Single turn vs Multi turn)	Channels (Web, Apple chat, whatsapp etc.)	Adaptive Cards (show rich content/forms)	Dynamic List (eg. Showing Purchase history)	Latency sensitive actions <5 secs	BF Skill	BF Composer	PA Flow	IVR (Voice)	Visio Conversation Flow
	Billing and Payments															
	BP-001	"How much do I owe for my service	Payment Balance	Balance inquiry	X				Y	Y		A-001		Y	X	Y
	BP-002	"I need to pay an invoice"	Make a payment	Make a payment (CC or ACH)	X	X			Y	Y					X	Draft
	BP-003	"Can I prepay for my order? "	Make a payment	Prepayment (CC or ACH)	X	X			Y	Y					X	Draft
	BP-004	"Can I setup automated payments?"	Make a payment	Enroll in Auto Pay	X	X			Y	Y					X	Draft
	BP-005	"can I pay my bill with the card on file that ends with 4444?"	Make a payment	Pay or Prepay with payment method on file	X	X				Y						Draft
	BP-006	"what is my payment history?"	Payment history	Payment History		X			Y	Y						Draft
	Orders and Shipping															

Note: A link to this spreadsheet is included in the Appendix section

4. Determine the phase of Delivery for the identified topics. It is always recommended to have subset of topics marked for MVP and build out the Phase 2 after that. This will help incorporate the feedback from MVP.

5. A bot typically has a mix of large multi-turn topics or short single turn topics. Identifying the type of topic will help the Bot author to build the topic design accordingly. Understanding the type of topic needed will help understand the approach/considerations. Examples:

a. Single-turn: Q: “What is your return policy” A: “Returns must be made within 30 days with a valid proof of purchase”.
b. Multi-turn: Q: “I’d like to make a return” A: “Sure what is your order number” Q:”123456” A: “Which item would you like to return” (etc.…)

Generally, customers implementing several single-turn topics (e.g., FAQ) do so with a shared approach for all – for example:

a. Review/Edit FAQ document/source
b. Import using the “suggested topics” tool
c. Review topic triggers
d. Review topic response

This can deliver valuable content to the bot very quickly and provide a baseline ‘surface area’ of bot triggers (the phrases your bot can respond to meaningfully). However, most customers spend the majority of bot development time building multi-turn topics, where more impactful processes are implemented. Multi-turn topics are built on a case-by-case basis, with some design consideration being made for each. Consider the following:

a. Define a goal (or goals if building complex topic/subtopics)
b. For each goal, what data is needed from the user, what is needed from a backend source?
c. Is data privileged/does it need to be secured?
d. Consider the outcomes/exit conditions for each goal (e.g. success, escalate to agent, retry etc.)
e. Multi-turn topics can often be designed as a large funnel (e.g. generic trigger phrases) that then help to filter a customer requirement. A good approach when doing this is to redirect to another (subtopic) when the requirement has been clarified. This subtopic can have more specific trigger phrases, making the bot easier to use for those with clear needs.

6. A particular topic or bot may be available across multiple channels like web, social channels, IVR etc. It is important to identify what channels are planned for a topic. This will help to ensure that the topics you build renders and works consistently across multiple channels.

   Note: not all features of PVA are available across all channels. Be sure to test across all relevant channels.

7. The topic can be built completely inside PVA in no code fashion, however if you want to add some additional pro code capabilities not available natively in PVA, you may have to use the extensibility options available in PVA like the Bot Framework Composer, Skills (if necessary) and Power Automate flows, which are more pro-code or low code. It will be helpful to identify

upfront the topics that needs pro code / low code extensions as part of requirements collection. This will help the Action developers to start focusing on those topics and actions.

8. There are some key capabilities that can enabled through extensibility. Below is the most common list of use cases. You can add these to your requirement collection to track if the topic will need any of these capabilities. For example, a topic may need the following capabilities which will need Bot Framework Skills or Composer or Power Automate

    Power Automate (low code)
    - Connectivity to various SaaS platforms using out of box connectors
    - Connectivity to on-premises systems using On-premises Data Gateway
    - Lightweight workflow or orchestration

    BF Composer Actions: (low + pro code)
    - Complex back-end API integrations
    - Adaptive cards display (rich content / forms)
    - Dynamic lists (bind on an array variable)
    - Generate random responses
    - Channel specific authoring
    - Multi-language support
    - Language Generation - conditional language generation that allows you to provide your chatbot with some personality

    BF Skills Actions: (pro code)
    - Access attachments from Omnichannel for Dynamics 365 Customer Service
    - Complex business logic/integrations best written in code

9. An important consideration to note when planning out your bot is Power Virtual Agents is the Software as a Service (SaaS) deployment model. Integrating with Platform as a Service (PaaS) components (e.g. APIs running in Azure, or a Azure Bot Service) or platforms running on premises can add additional complexity in terms of administration or security. Customers should ensure that these dependencies are identified early, giving time for appropriate governance to be agreed upon.

10. You can also refer to the PVA Architecture guidance video series for partners and customers. In this series, we focus on the following topics:

    - Planning your first bot
    - Build out your PVA bot
    - Extend your PVA bot with the Power Platform
    - Extend your bot with Azure
    - Advanced topics
    - Governance

We also provide a sample solution so that you have an end-to-end experience on the relevant topics.

11. As part of developing the actions associated with a given topic, also identify the business APIs needed, and any business rules that needs to honor for that topic flow. Also ensure that you have the access to these businesses APIs to avoid delays.

12. Next, you can identify the various design activities like Conversation tree drafts using Visio, user experience mockups for Adaptive Cards and Conversation end state (how the conversation is designed to end) that will be associated with a given topic.

13. Finally obtain internal stakeholder approval for each topic, as this is key for going live. Following the above framework will help you organize and identify the requirements for your bot before building it

Action Development

Action development is key for enabling high value scenarios where the Bot can perform actions on behalf of the agent or user. We have found a few common elements that are considered as part of Bot requirements collection.

For example, in a typical conversational commerce scenario we have identified six customer service categories that span across verticals and will need development of actions. You can use these as some base line ideas for the action development for your bot:

- Billing and Payments
- Orders and Shipping
- Account Management
- Troubleshooting

Billing and Payments: Actions developed for this scenario include making payments through the bot, seeing the billing history, viewing invoices etc. PVA is not PCI compliant so will not be able to support collection of credit cards info out-of-the-box, however you can build an action to leverage the existing card from the user account and use it for enabling payment scenarios.

Orders and Shipping: Order and shipping actions typically include checking order or shipping status, showing the list of orders, return or cancel orders etc. These actions will have to be latency sensitive since listing purchase history, processing orders etc. typically take longer time.

Account Management: The account management actions in the bot typically include login action or silent sign in, so bot can collect the user information like name, order status etc. You can use this to light up many context-aware actions in the bot like greeting the user with their name, showing their shipping status upfront, checking gift card balance, checking reward points etc.

Troubleshooting: These are actions that can run diagnostic tool etc. or execute a step-by-step list of actions the user to troubleshoot the issue the customer is facing. These are typically technical support use cases. The PVA extensions today only support server-side actions. If you want to build a client-side action it will have to be custom implemented.

These actions can be built using Bot Framework Composer, Bot Framework Skills or Power Automate cloud flows depending on the complexity of the actions and the integrations involved. The action development template helps you to organize your actions in a seamless way. As you plan your actions for the bot capture the below information so you can organize and track the Actions,

1. Action ID (Unique Identifier format for the action)
2. Name of the Action
3. Use cases affected by the action (from the Requirements Template)
4. Sprint for the Action development (tracking the timeline for completion)
5. Priority
6. Type of Action:
    - Bot Framework Composer
    - Bot Framework Skill
    - Power Automate cloud flow
7. Endpoint for the Action (Skill, cloud flow) or Bot Framework topic
8. Action Description
9. Action Sample Data - Define the payload for the action to help with the development
10. Test Accounts - Any test account info needed if you are developing an action that requires user authentication
11. Input/Output for the actions to help define the variables and understand what information needs to be collected by the bot for the action to execute and provide the desired output
12. Steps -document the steps that needs to be performed by the action
13. Business Rules - Document the business rules required for the action, for e.g. for cancellation action, the action should check the business rule on when the order was placed
14. Business API endpoints - Document the Business APIs that will be called by the action and their readiness

Action ID	Action	Use Case(s)	Sprint	Priority	BF Compose	BF Skill?	Endpoint	Action Desc.	Action Sample Payload	Test Ac
A-001	CheckPaymentBalance	BP-001	1	1	No	Yes	TBD	Checks for the payment balance using account ID		

Application Life Cycle Management (ALM)

Organizations should also start planning their Application Lifecycle Management processes to ensure they can smoothly promote their chatbots across environment landscapes.

A foundational construct involved in the ALM process is the use of Solutions. Solutions allow you to logically group related components together so they can be deployed to other environments as a single deployable unit.

There are two types of solutions: Managed and Unmanaged and poses an important question for organizations when choosing between which type of solution they should be exporting their chatbot as.

In general, unmanaged solutions are applicable in development environments where changes will occur. However as organizations promote their chatbot from development to test, quality assurance, or production environments, they should consider using managed solutions which prevent people from editing components within a managed solution. Using managed solutions promotes making changes back in the development environment to ensure changes are consistent and deployed in a consistent manner.

Once we have standardized on whether managed or unmanaged solutions will be used, we need to determine how we will transport these solution packages across environments. We have two methods for promoting solution packages:

1. **Manual**

   As a bot maker, we can export a solution package manually from the Power Automate or Power Apps maker portal. During this process we can select whether we want to export the bot as a managed or unmanaged solution from our development environment. We subsequently need to choose our target environment by using the environment picker in the maker portal. Once we are in our target environment, we can choose to import our solution package into that environment.

2. **Automated**

   Microsoft provides Power Platform Build Tools for both Azure DevOps and GitHub actions. Using these tools, we can assemble pipelines that allow us to automate our ALM process. This may include:

   - Exporting our solution from our development environment
   - Unpacking our solution and persisting our source code in a code repository
   - Packing our solution back together from our source code repository
   - Importing solution package into target environment

   Another deployment consideration for organizations is the sharing permissions for the chatbot. Once your bot is in your target environment, who should get access to maintain the bot and how much access should they have? Power Virtual Agents have the following roles:

   - Bot Manager
     - View, edit, configure, share, chatbot but don't delete it
   - Power Automate User
     - Create and add flows to the chatbot
   - Transcript Viewer
     - Can view transcripts of chat sessions with end users

   Large Chatbots

   When it comes to large bots, further considerations are required. For chatbots that have more than 150 topics, deployment processes can timeout using manual processes. Having an automated deployment process that takes advantage of solution "updates" vs "upgrades" may reduce the amount of time to deploy.

Fallback Strategy

PVA supports adding a customizable fallback topic to support scenarios when the Bot does not have a topic that matches the user's intent / query. This fallback topic can be modified to provide answers for long tail queries which do not have topics built in PVA. Some of the common approaches for fallback strategy include using QnA Maker for surfacing FAQ/Single Turn style topics or using a custom search to show search results.

Environment Strategy

Microsoft provides a comprehensive Environments strategy in our product documentation. However, there are some considerations related to Power Virtual Agents that should be accounted for.

- Region/Data privacy

  Environments are bound to a specific region. Does the chatbot have any regulatory requirements related to privacy and residency? For example, a foreign government customer may have a requirement to provision their chatbot under their country's tenant instance instead of USA tenant, ensure the data lives in their country specific data center. Power Virtual Agents lets you choose where you want your data to live. Power Virtual Agents can be deployed into the Microsoft Azure datacenters (also referred to as "regions") listed here. You can create a bot in your tenant's location by default, or you can choose the datacenter you want to use when setting up your environment. Data will be stored in the United States if a bot author's tenant location isn't listed under the Data locations table below.

  Please refer to the following link for additional information on Power Virtual Agents regions: Regional settings and data locations - Power Virtual Agents | Microsoft Docs

Azure geographic locations (geos)	Azure datacenters (regions)
Asia Pacific	Southeast Asia (Singapore), East Asia (Hong Kong)
Australia	Australia East (New South Wales), Australia Southeast (Victoria)
Canada	Canada Central (Toronto), Canada East (Quebec City)
Europe	West Europe (Netherlands), North Europe (Ireland)
France	France Central (Paris), France South (Marseille)
India	Central India (Pune), South India (Chennai)
Japan	Japan East (Tokyo, Saitama), Japan West (Osaka)
South America	Brazil South (Sao Paulo State) (As there is only one region in Brazil, customer data in Brazil South may be replication to South Central US (Texas) for disaster recovery purposes)
Switzerland	Switzerland North (Zurich), Switzerland west (Geneva)

United Kingdom	UK South (London), UK West (Cardiff, Durham)
United States	East US (Blue Ridge, VA), South Central US (Des Moines, IA), West US (Quincy, WA)

- Security Roles/Privileges
  Environments include predefined security roles that reflect common user tasks with access levels defined to match the security best-practice goal of providing access to the minimum amount of business data required to use the app.

  Users in the environment must have the Environment Maker security role before a PVA bot can be shared with them. For users who make apps that connect to the database and need to create or update entities and security roles, you need to assign the System Customizer security role in addition to the Environment Maker security role. This is necessary because the Environment Maker security role doesn't have privileges on the environment's data.

  System Administrators of the environment need to assign the Environment Maker security role to the user before you share the bot. If you're a System Administrator, you can assign the Environment Maker role when you share the bot. You can only share a bot with individual users, which means you can't share it with a security group or distribution group in your organization. You also can't share with users or groups outside of your organization.

  Here is the quick reference on the various Security Roles and privileges,

Security role	Database privileges*	Description
Environment Maker	Customizations	Can create new resources associated with an environment, including apps, connections, custom APIs, gateways, and flows using Microsoft Power Automate. However, this security role doesn't have any privileges to access data within an environment. More information: Environments overview Environment Makers can also distribute the apps they build in an environment to other users in your organization. They can share the app with individual users, security groups, or all users in the organization. More information: Share an app in Power Apps
System Administrator	Create, Read, Write, Delete, Customizations, Security Roles	Has full permission to customize or administer the environment, including creating, modifying, and assigning security roles. Can view all data in the environment. More information: Privileges required for customization
System Customizer	Create (self), Read (self), Write (self), Delete (self), Customizations	Has full permission to customize the environment. However, users with this role can only view records for environment entities that they create. More information: Privileges required for customization
Bot Transcript Viewer	View Chat transcripts	After assigning the Bot Transcript Viewer security role to a user, that user can access conversation transcripts for all bots that they create or are shared with in the environment.

Bot Manager	Create, edit, share bot	Bot Manager can view, edit, configure, share, publish bot but not delete it.

*The scope of these privileges is global, unless specified otherwise.

- Security Groups

At the bot level, the "access" option controls who can access your bot. You can select one of two security groups:

  - **All bot managers**. This lets only bot managers to chat with the bot. You can share your bot so other bot managers can access it.
  - **Everyone in my organization (***Organization name***)**. This lets everyone in the organization access and chat with your bot. Users outside of the organization will see an error when chatting with the bot.

- Data Loss Prevention (DLP) policies
  Organizations want to safeguard their data by using existing Microsoft Power Platform capabilities like data loss prevention. With DLP enforcement in Power Virtual Agents, bot authors can ensure that data used by connectors and skills is properly managed and authenticated.

- Shared vs Dedicated
  The environments for the PVA can be either dedicated or shared with other Dynamics 365/Power Platform Apps. In the table below are some pros and cons for shared vs dedicated environments.

Dedicated environment	Shared environment
PRO	**PRO**
Independent PVA entities. No touchpoint with any other Dynamics 365 entities.	Independent PVA entities. No touchpoint with any other Dynamics 365 entities.
Storage impact: • **Bot entity**- Minimal • **Bot component entity –** Minimal • **ConversationTranscriptEntity –** May become large based on the Data retention policy. Alternative is to move data periodically to Azure Blob storage.	Storage impact: • **Bot entity**- Minimal • **Bot component entity** – Minimal • **ConversationTranscriptEntity** – May become large based on the Data retention policy. Alternative is to move data periodically to Azure Blob storage.
Control the bot release schedule to Production autonomously without depending on other Apps. Enable faster deployment cycles.	Connects to Dynamics 365 entities through PVA extensibility with Power Platform connectors.
Easy to move to common environment in future using the ALM Import/Export	Re-use Dataverse customizations for Dynamics 365 again for PVA

CONS	CONS
Re-build the Dataverse customizations, other IT overhead for the separate env.	Dependent on the org wide Dataverse release schedule, may slow down Bot release cycles to Dataverse since Bots have typically a faster release cycle compared to Dynamics 365 Apps changes
Cannot connect to Dynamics 365 entities through Power Automate cloud flows if PVA is in different Dataverse. Power Automate cloud flows and PVA needs to be in same Dataverse environment as Dynamics 365 for using connectors to use Dynamics 365 entities data.	Will add more pressure to Dataverse usage, which adds to the cost structure.

- Omnichannel for Dynamics 365 Customer Service
  It is usually recommended to have a shared environment for PVA and Omnichannel for Dynamics 365 Customer Service. This will provide the extensibility for Dyanmics 365 entities through the connectors in a low code fashion and enable re-using existing Dataverse customizations for the data export pipeline, clean up etc.

- Environment Types
  Customers generally have somewhere between 3 and 5 environments for their PVA from Development to Production, using the Production and Sandbox environment types. Example of a 5-environment strategy:

  - **Development (sandbox environment)** – a sandbox environment where developers can model complex changes before applying them to Dev/Content
  - **Build (sandbox environment)** – changes to content, configuration and new development is done here and checked in to source control.
  - **Test (sandbox environment)** – most day-to-day testing is done here – although environment is not representative.
  - **Pre-production (sandbox environment)** – representative environment for final test sign-off
  - **Production (sandbox environment)** – live environment

  More on Power Platform environment types.

  It is recommended to have the environment provisioned in the same geographical location as the tenant to ensure optimal performance. The Scale group of the environment determines the performance of the bot. The Trial and Default environments have less capacity and hence it is advised to use Production environments as much as possible, especially for the live/production chatbot that will be consuming the external traffic. The deployment of the bots across environments can be automated using Azure DevOps tools.

Custom Analytics Strategy

Power Virtual Agents provides comprehensive out-of-the-box analytics in the PVA portal that will allow customers to understand a bot's usage. Customers can view reports related to:

- Performance and usage
- Customer satisfaction
- Session information
- Topic usage
- Billed sessions

There are scenarios where a customer will want to take what is provided by default in PVA further. Generally, these break down as:

- A requirement to share analytics with non-PVA makers/users
- A requirement to store/archive bot data (longer than the default 30 days)
- A requirement for a report not covered in the OOB analytics

If this is the case, our recommended approach is that customers implement the PVA custom analytics solution. This is a set of open-source assets, distributed through GitHub, greatly accelerating the time it takes to create a report that renders in Power BI.

Important! Before implementing this with a customer – it is critical that they understand the solution is not part of the core PVA offering and will require configuration and support by themselves.

Below we will walk through the components in the report and how they work together.

Dataverse

The analytics shown in the PVA portal come from a data service residing within PVA itself. At the same time, usage data is also written to a customer's Dataverse instance. Both sources have a data retention of 30 days (the customer can change this for Dataverse all-up – although there is a different approach – see below).

The important tables used for custom analytics are as follows:

- Bot – details of each bot in an environment. Generally a small amount of data.
- BotComponent – these are topics in the bot in your environment. Generally a small amount of data.
- ConversationTranscript – the details of the activity for all the bots in your environment. Data size is related to amount of bot usage and can be quite large. More detailed description of this table can be found here – work with conversation transcripts.

Azure Synapse Link (Azure Data Lake Storage V2)

For customers with larger PVA bots (roughly 10k sessions/month, greater than 80MB of transcript data), the recommended approach is to export bot data to Azure Data Lake Storage v2 using Azure Synapse Link. This has additional benefits of creating an archive of bot data, allowing reports on data greater than 30 days. The export will create an incremental sync of configured tables in Azure using the Common Data Model format. There is some work on top of the base template to do this:

a. Create a Data Lake and connect it to Dataverse
b. Browse to the ConversationTranscript (only, the other tables do not support incremental) table in Dataverse and select "Azure Synapse Link" to set up incremental export.
c. Follow the guidance for setting up Power BI DataFlows to process the incoming data.

Power BI

The Custom Analytics solution includes a Power BI report that processes the raw transcript data (using PowerQuery) into to a report matching the PVA Portal analytics. In addition, users have access to:

- Data for all bots in an environment
- Data as far back as the feed provides (assuming Azure Synapse Link archival is configured, otherwise 30 days)
- Raw data tables extracted from the ConversationTranscript data – which can be used to create custom reports.
- A transcript viewer, allowing users to see the actual conversations that occurred.

Important! This a complex report that performs significant transformations on the base data. Customers with ConversationTranscript tables greater than 80MB should look to use the Azure Synapse Link/DataFlows version of the report.

Testing

Testing the bot before go-live is key to ensuring that the goals of the project have been met. While testing can be done by several team roles, one of the key assets a team should consider creating is an intent matrix. This can be used to ensure the expectations of how a bot is responding are being met.

Intent Matrix

An Intent Matrix tests provides the test cases used to validate the triggering behavior of the bot before it is published. It measures the test intent/phrase against the possible outcomes of bot response. Those possible responses are as follows:

1. The intent is recognized and directed to the correct topic
2. The intent is recognized and sent to the wrong topic
3. The intent triggered 'Did you mean' for one of 2 reasons:
    a. There is more than one topic scoring a high intent score (generally meaning that topic triggers overlap)
    b. There are no topics scoring a high intent score (generally meaning missing topic triggers)
4. The intent is not recognized
5. An error occurs

Types of testing

- Project level unit testing by Bot developer: This is the Unit testing done by the Bot Developer (Content Author or the Pro Dev) during the implementation of the test cases. This is typically done in the Dev environments in isolation. This testing ensures that the key use case / action implemented is working with within the bot. The Test client / Demo web site can be used for this testing.

- Integration testing by Business SMEs: After the project level testing is done and the actions are wired up, it is advised to perform an integration testing with help of the Business SME within the organization who are vested in the Bot development. During this integration testing we can identify any gaps between the original business requirements and the current implementation. It is also a good way to identify end-to-end testing of bugs in the bot. If Omnichannel for

Dynamics 365 Customer Service is part of the implementation, the testing will uncover any integration issues between PVA and Omni-channel as well.

- Channel-specific testing: Customers publishing to more than one channel (e.g. Web & Facebook Messenger) should ensure their test planning covers testing of each available channel. Differences of rendering behavior can manifest, particularly in complex or semi-custom scenarios. Customers using PVA with an escalation channel (e.g. Dynamics 365 Customer Service) should similarly ensure testing spans any interconnected systems.

- Internal Pilot Testing for Organization (Dogfood): After unit and integration testing is complete, it is recommended to undertake an internal pilot for the bot – also known as Dogfood testing. This allows all employees to use and explore with the bot and log issues. The pilot is very useful for uncovering any issues missed by unit and integration tests. This also uncovers any early performance bottlenecks that may be present in the bot. It is also a good idea to have bug-bashes and announce prizes for finding bugs during these internal pilots to incentivize people to try out the bot.

- External phased testing: This is the final phase of testing done in the production environment as part of the ramp up to go-live. The chatbot can be exposed to limited traffic or fewer end points initially to start getting some traffic. You can run the end-to-end tests for the bot in production to ensure performance and expected results are occurring, before ramping up the traffic to all the entry points like web, SMS, social channels, mobile app etc. Ideally the phased testing can start with certain pages like “Support” page etc. before ramping up the “Home” page and more channels.

- After the final phase of testing, it is also necessary to start tracking the metrics for the bot on a daily / weekly basis to observe the performance of the bot. The common metrics tracked by organizations are number of sessions, Resolution rate, Abandon rate, CSAT, Escalation rate etc. You can also tweak the out of box analytics for additional insights by building your own custom analytics.

Capacity Modeling

While the Software-as-a-Service model used in Power Virtual Agents mitigates many of the normal application scalability/capacity concerns, there are some areas where project teams should be mindful of planning to ensure smooth running.

Quotas

PVA does employ a quota limiting numbers of requests per minute (currently 800 rpm). Customers who believe they will go above this limit should get in touch with their Microsoft Account team.

Power Automate

Customers calling Power Automate from PVA benefit from an enhanced daily quota of calls (currently 30,000). Customers who believe they will go above this limit should get in touch with their Microsoft Account team.

Demand spikes

Customers expecting very large spikes in seasonal demand should consult with the Microsoft Account team ahead of the spike to see if any local mitigation may be needed.

Go-Live Readiness

For organizations that have existing change management processes, introducing a chatbot should align to those processes. However, the following considerations should be evaluated against any existing change management processes to ensure there aren't any gaps that are aren't addressed.

Stakeholders sign off

Chatbots are often externally facing. As a result, they represent important interactions with customers or partners. In these scenarios, there are naturally some messaging needs to ensure that the chatbot accurately reflects corporate culture and messaging.

Internal communication

Deploying a chatbot is an accomplishment worth celebrating. Ensure that you take this opportunity to share with your employees the chatbot launch and encourage them to explore its functionality. For some organizations, they even host competitions to name the chatbot. This allows employees to get behind the chatbot to drive a business outcome, like improved customer service.

Updating any prior links to chatbot

During development or testing, you may have used links as placeholders within the chatbot. Ensure all links are updated to ensure users of the bot have the best possible experience.

Amplification Opportunities – with Microsoft

Microsoft is always interested to learn how customers are using our technology to improve the lives of their employees and customers. Microsoft is happy to celebrate your success by publishing posts on the product blog or in our customer case study repository on https://customers.microsoft.com/. If this is of interest to you or your customer, please get in touch with your Microsoft account team.

Post Go-Live Support

As customers approach go-live there are some points of advice we can give to ensure smooth operation of the service.

Ensure IT awareness

Although this may seem obvious, as PVA is a no-code platform, quite often the teams building out a bot may be a fusion of business and IT, or in some cases only have tentative connections to the core IT team. Ensure the customer has documented/updated processes for:

- Service architecture/design
- Application lifecycle
- Knowledge base
- Issue identification/support
- Service administration through the Power Platform Administration Center

Engaging support

When any new service is onboarded, it is important to have a defined process for capturing issues and engaging help. Your Microsoft account team will be able to help you understand support options available to you.

New User Onboarding

The core team who has built out their chatbot (or chatbot Center of Excellence) likely will want to bring on additional bot authors/editors. Ensure a customer has a process for:

- User license assignment (see above)
- Training. For editors, the PVA in a Day training materials provide an important grounding in the concepts. Customers have commented on the materials on Topic trigger design are very useful for those iteratively developing their bot.

Reviewing goals - closing the loop

One of the first sections of the document discussed goals for delivering a successful chatbot. Managing those goals as an ongoing business concern is becomes the key part in the lifecycle of the bot:

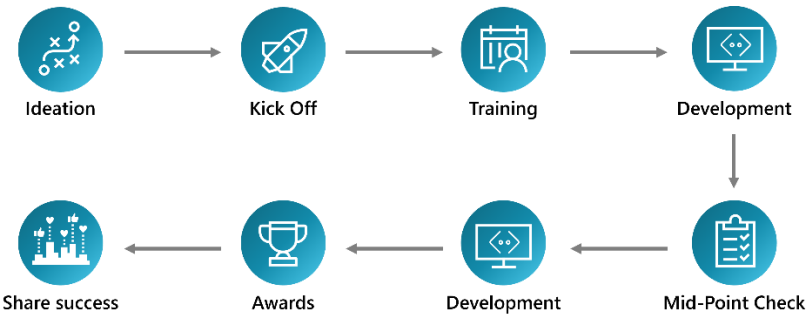
1. Building the bot
2. Check development goals have been met
3. Review bot metrics in production
4. Specify development activities to improve bot metrics
5. Return to step 1

Nurture – Bot Wars

Bot wars is an opportunity to democratize who builds chatbots within an organization. It is an opportunity for employees to learn and for the organization to scale chatbot development. The program includes the following:

1. Interested employees will participate in a Power Virtual Agents in a Day training where they will learn the fundamentals of chatbot building.
2. Once attendees have experience building chatbots, split the training group into small teams of 2-3 people.
3. Allow teams to identify opportunities that they feel they could solve using a chatbot.
4. Each team should then establish milestones over a 4-week period that includes activities each team member will perform. This is very similar to a project plan.
5. After 2 weeks, bring the entire training group together and have each team present their chatbot. This provides an opportunity for each team to demonstrate their progress and identify any challenges/gaps.
6. After 4 weeks, bring the entire training group back together to show off their final product to the rest of the group and leaders from the respective business areas.
7. When possible, provide some level of rewards to the participants of this initiative. Incentives can vary from recognition, all the way to monetary rewards.
8. Include a “path to production” period to ensure all chatbots make it into production.
9. Share success with the rest of the organization. Inspire others to do more.

10. Rinse and repeat. You have now built a program that you can run on a regular cycle to further promote chatbot development within your organization.



Bot Wars encourages organizations to learn and apply what they've learnt in real-world use cases to develop bots

Licensing

The licensing of Power Virtual Agents is relatively simple with a few key components. Most important to understand is that the license is based on chatbot Sessions. A Session is defined as:

"The session begins when a user topic is triggered and ends when one of these conditions is met—either the user's questions are answered, or the conversation exceeds 60 minutes or 100 turns."

There are 3 main ways in which customers acquire PVA licenses:

Power Virtual Agents

1. Tenant license (billed monthly, capacity pooled at Tenant)
    a. 2000 sessions (no limits on channel)
    b. Power Automate rights included (with higher throttling limits), including Premium connectors (All flows must start and end with PVA connectors)
    c. Dataverse capacity (10GB DB, 10GB File, 2GB Log)
2. Add-on license (billed monthly, capacity pooled at Tenant)
    a. 1000 sessions (no limits on channel)
3. User license (Grants permission to author bots)
    a. Currently available for no cost, but should be purchased in the same transaction as the Tenant license
    b. Assigned to users in the admin portal
    c. One license required for each bot author

An example customer, who expects up to 10000 sessions a month for their bot, which will be managed by 10 users, would purchase 1x Tenant license, 9x Add-on license, and 10x User licenses.

Power Virtual Agents for Teams

1. Microsoft 365 license
    a. Unlimited sessions (Teams channel only)
    b. Power Automate rights (Standard connectors only)

Generally, if a customer is interested in using PVA for Teams, no further licensing is required (as they need Microsoft 365 to use Teams) unless they wish to deploy their bot to additional channels (they will need the standard PVA licenses detailed above).

Chat for Dynamics 365 Customer Service (Omnichannel)

Customers deploying PVA with Dynamics 365 for Customer Service now receive a version of the PVA tenant license with their Dynamics 365 purchase:

1. Tenant license (billed monthly, capacity pooled at Tenant)
    a. 1000 sessions per tenant
    b. Power Automate rights included (with higher throttling limits), including Premium connectors (All flows must start and end with PVA connectors)
    c. Dataverse capacity (10GB DB, 10GB File, 2GB Log)

Add-on and User licenses are still required, as detailed above.

Latest information on Power Virtual Agents licensing.

Best Practices - Summary

Hopefully, what is covered in this document has provided a good foundation for successful delivery of a complex project based on Power Virtual Agents. We thought it was important to provide a short breakdown of the best practices we view as most important.

Across various engagements, we have found a few techniques to be very beneficial towards delivery success:

1. Focus on delivering immediate value. Building chatbots is an iterative initiative. You will learn from your users what they are looking for. Use these signals to drive further investment.
2. Establish role clarity. Make it clear who is on point to make decisions and there is no ambiguity in delivery.
3. Occasionally customers run into technical issues – when implementing with specific requirements or integrating with other systems. These issues can take time to diagnose; identify risks early and plan contingency accordingly.
4. Alignment on issue priorities (what is a blocker, what is a must have, nice to have, etc.) and how aggressively to push for using workarounds while the underlying issues are addressed. Workarounds are a natural part of projects as they allow the project to continue to make progress while a longer-term solution is found.
5. Determine your Application Lifecycle Management (ALM) strategy – move to production-level process as early as possible. Don't leave these activities to the end when there are timelines to be met.

Appendix

Sample Templates

PVA Requirements Template
https://github.com/microsoft/PowerVirtualAgentsSamples/raw/master/Playbook/PVA_Req_Template.xlsx

PVA Actions Development Template
https://github.com/microsoft/PowerVirtualAgentsSamples/raw/master/Playbook/PVA_action_dev_template.xlsx

PVA Intent Test Matrix
https://github.com/microsoft/PowerVirtualAgentsSamples/raw/master/Playbook/Intent_Test_Matrix.xlsx