

Capability Envisioning Template

AI for ISVs

June 2024



Which generative AI approach should you pursue?

Data	Customizability	Development Complexity	End user	
 Where is my data? What infrastructure do I need to connect it? How do I need my data to interface with Microsoft data and other sources? [Your response here] 	 How will customers interact with my app? Do I need to control its tone and behavior? Am I prepared to manage security and compliance? [Your response here] 	 How much time can I dedicate to app dev? What resources can I dedicate to this app? Who do I have available to staff to this project? [Your response here] 	 Who is the end user? How much technical know—how will my end-user have? Where would the pilot interface with the app? [Your response here] 	Adopt or extend a Copilot
				Build a custom copilot
				Build an app on Fabric

Which generative AI approach should I choose?

within ISV's own products

• Mid-high level financial investment

Align to your highest capability needs

Approach A • Limited data sources surfaced via plugin Considerations • Limited customizability/managed by Microsoft • No/low-code/low lift requirements Adopt/extend What data requirements · Non-technical end user/solution integrating within a 1p Copilot does my solution need? Microsoft products Low-mid level financial investment How much control and customization do I want? Approach B • Multiple data sources surfaced via API What amount of • Moderate to full customizability development lift can I • Pro-code/moderate lift requirements provide for this project? Build a • Non-technical end user/solution integrating within custom copilot ISV's own products • Who is my end user? • Low-high level financial investment How much of an investment can I justify based on the • Capable of complex data infrastructure Approach C projected profitability? • Full customizability/shared management with Microsoft What amount of risk can • Pro-code/high lift requirements Build an app I justify for this solution? Non-technical or technical end user/solution integrating

on Fabric

Which pattern from "Adopt/extend a Copilot" should I choose?

through Copilot Studio

Adopt/extend a Microsoft Copilot

- Data needs are met through plugins
- Little customizability required
- Low-code is preferred/required
- Designed for non-technical end users using Microsoft product
- Low projected profitability justifies a low-level investment

Enables existing Copilots to interact with ISV apps and services to expand capabilities, such as retrieving information or executing actions
Can be created with various tools such as Teams Message Extensions and Power Platform plugins

- Enables ISVs to connect their data to the Microsoft 365 Semantic Index to make it searchable and actionable for users
- Uses Fabric to connect ISV data to the Microsoft Cloud and the Microsoft Graph, which allows Microsoft Copilot to access it
- Provides Teams chatbots with natural language capabilities
- Can build on existing capabilities such as safety features that can be reused for ISV application
 - Doesn't require integration with Graph data

Pattern A

Create plugins

Pattern B

Graph connectors

Pattern C

Teams Al Library

Which pattern from "Build a custom copilot" should I choose?

Build a custom Copilot

- Requires multiple data sources (POS systems, ERPs, inventory management) surfaced via API
- Some customizability required
- Pro-code falls within ISV capabilities
- Designed for non-technical end users
- Medium projected profitability justifies a mid-level investment

• Designed for generative Al-powered chatbots that Pattern D respond to user's questions • Requires fewer coding considerations **Power Platform** • Connects existing applications to bring in data from Connectors potentially any source • Helps bring data to users' fingertips by integrating Pattern E seamlessly with user-centric data • Designed to derive insights from data in Microsoft 365 Microsoft as opposed to external sources. **Graphs API** • Enhances existing apps with data-focused AI Pattern F • Provides an intuitive low-code approach • Designed to formulate responses to user's questions and offload simple tasks Azure OpenAl Can easily call to existing APIs **Assistants** • Is an all-in-one collaborative platform with existing Pattern G pretrained models that can be combined Offers higher customizability than other patterns • Includes complex coding requiring lift, including the **Azure Al Studio** Azure AI SDK and Prompt Flow

Which pattern from "Build an app on Fabric" should I choose?

Build an app on Fabric

- Data needs require complex infrastructure
- Full customizability is required
- Pro-code falls within ISV capabilities
- Designed for data scientists or other technical end users
- High projected profitability justifies a high-level investment

 Supports existing APIs, allowing ISVs to read, Pattern H write and manage data within Fabric OneLake • Utilizes OneLake Foundation through a series of connectors, such as REST APIs for OneLake Interop with or Data Factory **Fabric** • Enables shortcuts to provide a unified experience across ISVs entire enterprise Pattern I • Enables ISVs to develop products on top of the Fabric platform or combine Fabric with pre-existing applications Build on Focuses on using REST APIs that provide experiences like data warehousing, data engineering, real-time **Fabric** intelligence, and more Pattern J • Provides ISVs with tools to create custom workloads within the Fabric ecosystem, leading to high customizability. Build a Fabric • Offers a comprehensive toolkit for integrating applications into a workload hub, providing workload an enhanced analytics user experience