Intake Criteria for a Copilot Agent

This document outlines the general intake questions and criteria for the development path of a Copilot Agent.

1 Functional Requirements

This section describes the capabilities or functions that the Agent should be able to perform from a user's perspective.

1.1 Purpose of the Agent

- Clearly define the primary purpose of the Agent. What specific tasks or problems is the Agent designed to address? For example, is it meant to assist with customer service, provide technical support, or help with scheduling and reminders? Be precise and complete.

1.2 Happy Flows

- Describe the standard or default path a user will follow when interacting with the Agent. This includes the typical sequence of actions or steps that lead to a successful outcome. For instance, if the Agent is for customer service, a happy flow might involve the user asking a question, the Agent providing a helpful answer, and the user expressing satisfaction.

1.3 Other Flows

- Identify non-standard paths a user might take. These could include alternative ways to achieve the same goal or different types of interactions that the Agent should handle. For example, if a user asks an unexpected question or requests a feature that is not part of the happy flow, how should the Agent respond?

1.4 Error Flow

- Outline the procedures for handling technical or functional errors. What should the Agent do if it encounters an error? Should it try to resolve the issue on its own, redirect the user to a human agent, or continue the workflow outside the chat in a different process? Specify the criteria for each type of error and the corresponding actions.

2 Knowledge Sources

The Agent can only interpret data that is compliant and clean. This section details the sources and quality of data the Agent will use.

- **Data Location**: Where is the data stored? Identify the specific databases, repositories, or systems that contain the relevant data.
- **Data Type**: What kind of data will the Agent use? This could include structured data (e.g., databases, spreadsheets) and unstructured data (e.g., documents, emails).
- **Platform**: Which platforms are used to store and manage the data? Examples include SharePoint, Azure, AWS, etc.
- **Data Format**: What is the format of the data? Common formats might include DOCX, JSON, BLOB, etc.
- **Data Quality**: Assess whether the data quality is sufficient for the Agent's purpose. This includes ensuring the data is in the correct format and style, free of ambiguities, and devoid of duplicate entries. For example:
    - Are there ambiguous questions or answers that need clarification?
    - Is there any duplicate data that needs to be cleaned up?

If the knowledge base represents 100% of your relevant data, the base set for the development environment should have 20% of that volume if the production data cannot be used. For example, for a source of 10,000 webpages, the data set for development should include 2,000 webpages.

3 Large Language Model Expectations

An LLM is not an exact science. This section outlines the expectations from the models.

3.1 Generative Capabilities

- When an answer is generated from the collected information in the knowledge source, specify how you want it to respond. This could include:
  - **References**: Should the answer include references to the original sources of information?
  - **Quotes**: Should the answer contain direct quotes from the knowledge source?
  - **Paraphrase**: Should the answer be paraphrased to provide a more concise or simplified version of the information?
  - **Other**: Any other specific requirements for the generated responses.

3.2 Question Interpretation

- Define how the bot should interpret the questions that are asked. This could involve:
  - **No Change**: Should the bot interpret the questions exactly as they are asked, without any modifications?
  - **Prompt Engineering**: Is prompt engineering allowed to refine or modify the questions to improve the quality of the responses?
  - **Other**: Any other specific requirements for question interpretation.

4 Technical Requirements

This section describes the technical capabilities or functions that the Agent should be able to perform without user interaction.

4.1 Third Party Connections

- Identify the connections to other systems that are needed. This could include:
  - **Mainframe, ServiceNow, etc.**: Specify the systems that the Agent needs to connect to.
  - **APIs**: Are APIs available for these connections? If so, provide details on the APIs and how they should be used.

4.2 Security

- Outline the security requirements for the knowledge source. This could include:
  - **General Availability**: Is the knowledge source generally available to all users, or are there restrictions?
  - **UAC or RBAC**: Are User Access Control (UAC) or Role-Based Access Control (RBAC) implemented? Provide details on the security measures in place.

4.3 Feedback Mechanism

- Decide if a feedback mechanism is needed. This could involve:
  - **User Feedback**: Is the user obliged to give feedback on the Agent's performance?
  - **Storage and Processing**: How will the feedback be stored and processed? Provide details on the feedback mechanism and its implementation.

5 Test Information

The test sets must contain validated scenarios provided by the business.

- **User Interaction**: Describe how users will chat with the Agent. An email scenario or wording is not the same as a question to an Agent. Provide examples of typical user interactions.

Test_ID	Test_Question	Test_Short_Answer	Test_Explanation
001			
002			

- **Test Scenarios**: The projected number of test or validation scenarios must be 20% of the initial training data set. A good rule of thumb here is two days' worth of production scenarios. This can be recalculated with parameters for error and variance. The Sample Size Calculator can help with this. Sample Size Calculator
    - For example, if there are 2,000 questions/conversations per week, then the number of test scenarios must be 400.