

Measure success: Track conversation outcomes and user feedback on AI responses

You can't improve what you can't measure: design your agent to track successful and unsuccessful outcomes while collecting user feedback on AI-generated responses.

Lab Details

Level	Persona	Duration	Purpose
300	Advanced Maker	60 minutes	After completing this lab, participants will be able to design an agent that tracks conversation outcomes and collects user feedback on AI-generated responses. They will gain meaningful analytics to identify which knowledge sources drive the highest satisfaction (CSAT) and understand patterns leading to abandoned or escalated conversations.

Table of Contents

- [Why This Matters](#)
 - [Introduction](#)
 - [Core Concepts Overview](#)
 - [Documentation and Additional Training Links](#)
 - [Prerequisites](#)
 - [Summary of Targets](#)
 - [Use Cases Covered](#)
 - [Instructions by Use Case](#)
 - [Use Case #1: The end of conversation topic](#)
 - [Use Case #2: Create a smooth and intuitive end-of-conversation experience](#)
-

Why This Matters

Advanced Makers - Many organizations struggle with understanding whether their AI agents are truly helping users or creating frustration through abandoned conversations.

Think of running a retail store without knowing if customers leave satisfied or empty-handed: - **Without conversation tracking:** You see usage numbers but can't identify why users abandon conversations or which responses cause dissatisfaction - **With conversation tracking:** You gain actionable insights into user satisfaction, identify problematic knowledge sources, and continuously improve response quality

Common challenges solved by this lab: - "Our analytics show high usage but we don't know if users are actually getting help" - "Users seem to abandon conversations but we don't

understand why” - “We can’t identify which knowledge sources need improvement” - “There’s no way to capture user feedback on AI-generated responses”

The 60 minutes you invest in this lab will transform your agent from a black box into a data-driven tool for continuous improvement.

Introduction

In today’s AI-driven customer service landscape, simply having an agent that responds isn’t enough. Organizations need to understand whether their AI assistants are genuinely solving user problems or creating barriers to success.

Real-world example: A company deployed an AI assistant on their website to handle product inquiries. Initially, they celebrated high engagement metrics—thousands of conversations daily. However, customer satisfaction surveys revealed declining scores, and support tickets actually increased. The problem? Their AI was providing responses that looked helpful but weren’t actually resolving user queries. Without proper conversation outcome tracking and feedback collection, they were flying blind.

After implementing the strategies in this lab, they discovered that 40% of conversations were being abandoned after the first AI response, and users were providing consistent feedback about specific knowledge gaps. This insight allowed them to refine their content strategy, resulting in a 65% improvement in conversation resolution rates and significantly higher customer satisfaction scores.

Core Concepts Overview

Concept	Why it matters
End of Conversation Topic	Provides structured tracking of conversation outcomes (resolved, abandoned, escalated) enabling accurate measurement of agent effectiveness and user satisfaction patterns.
Conversation Resolution Tracking	Distinguishes between implicit and explicit resolution, allowing you to understand not just if users got answers, but whether they were satisfied with those answers.
Adaptive Card Feedback Collection	Creates unobtrusive, intuitive feedback mechanisms that capture user sentiment without disrupting conversation flow, increasing response rates and data quality.
Session-based Analytics	Tracks individual user requests within conversations separately, providing granular insights into which types of queries succeed or fail most often.
CSAT Integration	Connects user satisfaction scores directly to specific knowledge sources and response patterns, enabling data-driven content improvement strategies.
Generative AI Behavior Control	Ensures proper outcome tracking even when using dynamic AI responses, maintaining visibility into user interactions across different conversation modes.

Documentation and Additional Training

Links

- [Measuring agent engagement](#) ↗
 - [Measuring agent outcomes](#) ↗
 - [Deflection overview](#) ↗
 - [Key concepts – Analytics](#) ↗
 - [Custom analytics strategy](#) ↗
-

Prerequisites

- You need to have access to Microsoft Copilot Studio using <https://copilotstudio.microsoft.com/>.
 - You can either customize the agent from LAB-10 Create a knowledge agent for your public website or create a new agent with at least one knowledge source.
 - Basic understanding of Copilot Studio topics and conversation flow design.
-

Summary of Targets

In this lab, you'll transform your AI agent from a simple question-answering tool into a sophisticated system that tracks success and learns from user feedback. By the end of the lab, you will:

- **Configure conversation outcome tracking** using the End of Conversation topic to measure resolved, abandoned, and escalated interactions.
 - **Implement intuitive feedback collection** through thumbs-up/thumbs-down reactions that don't disrupt conversation flow.
 - **Create detailed feedback capture** for negative reactions, allowing users to provide specific improvement suggestions.
 - **Design smooth conversation endings** that balance user experience with data collection needs.
 - **Understand analytics insights** to identify which knowledge sources drive highest satisfaction and recognize abandonment patterns.
-

Use Cases Covered

Step	Use Case	Value added	Effort
1	The end of conversation topic	Effectively manage user interactions by understanding when and how to seamlessly redirect users to the end-of-conversation topic, enabling accurate tracking of conversation outcomes.	15 min
2	Create a smooth and intuitive end-of-conversation experience	Customize the default end-of-conversation topic to create a more seamless, conversational-friendly experience for users.	15 min

Instructions by Use Case

Use Case #1: The end of conversation topic

Every conversation should have a conclusion – Design for clear outcomes.

Use case	Value added	Estimated effort
The end of conversation topic	Effectively manage user interactions by understanding when and how to seamlessly redirect users to the end-of-conversation topic, enabling accurate tracking of conversation outcomes.	15 minutes

Summary of tasks

In this section, you'll learn how the end of conversation topic works in Copilot Studio and how to use it effectively in your conversation design. By structuring conversations for clear outcomes, you'll enable meaningful analytics and actionable insights to improve your agent's performance.

Scenario: You've built an agent with the knowledge to answer user questions—but is it actually delivering? If your analytics dashboard isn't showing meaningful data, it's time to track successful conversation outcomes and CSAT scores. You can't improve what you can't measure.

Objective

Configure your agent to properly track conversation outcomes by implementing the End of Conversation topic and understanding how it integrates with both classic and generative orchestration modes.

Step-by-step instructions

Understanding the End of Conversation Topic

1. Navigate to the Copilot Studio agent you have created for this lab (e.g., LAB-10, or a new one). <https://aka.ms/MCSStart>
2. Go to the **Topics** tab, display **All**, and select **End of conversation**.
3. Explore what the topic is doing.



Tip:

The end of conversation topic is meant to be triggered when the agent has presumably fulfilled the user's request. This can happen either after providing a direct answer, such as retrieving information from knowledge sources, or after completing a more complex multi-turn interaction where the user and agent exchange multiple messages to complete a task.

By default, when the conversation reaches this stage, the agent asks, "Did this answer your question?" At this point, the resolution is considered implicit, meaning that if the user leaves without responding, it is assumed that their query was resolved. However, if the user confirms that their question was answered, the resolution becomes explicit, and they are then prompted to provide a Customer Satisfaction Score (CSAT) to rate their experience.


Configuring Classic Orchestration Mode

4. Explore other topics in your agent. By default, are they redirecting to the End of conversation topic?

For newly created agents, that shouldn't be the case. The End of conversation topic must be redirected to explicitly from the places where you feel the user request has been

fulfilled.


5. Assuming the Generative mode for orchestration is **Disabled** on your agent (you can see that option either in the Overview tab, or in Settings, under Generative AI).

 Generative AI disabled

Generative AI disabled

6. Let's add a redirect to the End of conversation topic from the **Conversational boosting** topic.

Go to the **Topics** tab, display **All**, and select **Conversational boosting**.

 Conversational boosting topic

Conversational boosting topic

Delete the **End current topic** node, and instead, add a new node: **Topic management > Go to another topic >** and select **End of Conversation**.

Save the topic.

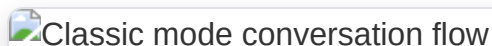
Testing Classic Mode Behavior

7. Now, test your agent in the test pane by asking a question that will trigger the Conversational boosting topic.

For example, you may ask:

What are the key metrics offered by the analytics dashboard?

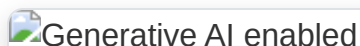
Answer the different questions until you can ask a new question. In the happy path, notice you must answer 3 questions before you can ask a new question.

 Classic mode conversation flow

Classic mode conversation flow

Configuring Generative Mode

8. Let's now try what the experience is by toggling the **Generative mode** on. You can enable it either in the Overview tab, or in Settings, under Generative AI.

 Generative AI enabled

Generative AI enabled

9. Refresh the test pane, and ask the same question.

What are the key metrics offered by the analytics dashboard?

Notice how the experience is different. The Activity map is displayed and shows you the agent's reasoning based on the user query.

Important:

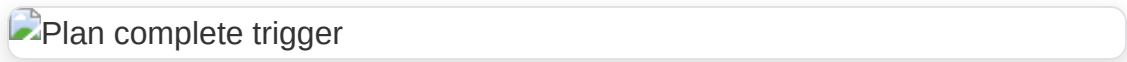
In the test pane, notice that you are no longer prompted with "Did this answer your question?". Why isn't End of conversation triggered? That is because the Conversational boosting topic wasn't traversed with generative mode.

Creating the Plan Complete Topic

10. Go to the **Topics** tab, select **+ Add a topic**, and choose **From blank**.

Don't leave it with the default "Untitled" label. Select **Untitled** and change the text to **Plan complete**.

Then, change the trigger by hovering over the "Triggered by agent" box until the icon to swap the trigger for another type appears. Then scroll down and choose **Plan complete**.



Plan complete trigger

Plan complete trigger

Add a new node and select **Topic management > Go to another topic > End of Conversation**.

Save the topic.

11. Now, refresh the test pane, and test your agent again. What are the key metrics offered by the analytics dashboard?

Notice that after the answer is provided by the agent, the user is prompted for confirmation.



Generative mode with plan complete

Generative mode with plan complete


Understanding Topic Behavior

12. Refresh the test pane and send a simple, everyday message, like "hello". Hi!

Notice how the End of conversation topic isn't triggered. Why is that?


Open the **Greeting** topic by clicking the edit (✎) icon.

Notice how the **End all topics** node prevents this behavior. Any subsequent user messages remain within the same conversation session, as the agent assumes the user's request hasn't yet been resolved.

 Greeting topic with end all topics

Greeting topic with end all topics

13. For the rest of the lab, you may disable the generative mode.

 Disable generative mode

Disable generative mode

Congratulations! You've completed Use Case #1!

Test your understanding

Key takeaways:

- **Conversation resolution tracking** – Redirecting to the end of conversation topic allows you to track successful, abandoned, and escalated interactions.
- **Session-based analytics** – Conversations can contain multiple sessions, each with a distinct outcome (resolved, escalated, abandoned).
- **Customizing conversation endings** – The end of conversation topic can be tailored to enhance user experience, ensuring smooth and meaningful conversation conclusions.

Lessons learned & troubleshooting tips:

- If your analytics dashboard is showing too many abandoned sessions, check if conversations properly redirect to the end of conversation topic.
- Analytics dashboards don't show sessions from your own tests in the test pane. Only the interactions that happened over your deployed channels will show.
- When using conversational boosting, ensure the topic transitions correctly to the end of conversation topic to capture user feedback.

Challenge: Apply this to your own use case

- How can you integrate clear conversation endings into your existing agent topics?
- Where should you collect user feedback to improve response quality?
- What patterns can you identify in abandoned vs. resolved conversations?

Use Case #2: Create a smooth and intuitive end-of-conversation experience

End conversations without friction – create a smooth, unobtrusive way to gather feedback without disrupting the flow.

Use case	Value added	Estimated effort
Create a smooth and intuitive end-of-conversation experience	Customize the default end-of-conversation topic to create a more seamless, conversational-friendly experience for users.	15 minutes

Summary of tasks

In this section, you'll learn how the default End of conversation topic can unintentionally interrupt the user's conversational flow, forcing unnecessary feedback prompts or confirmations. You'll see how to modify this default behavior to create a smoother and more intuitive experience.

Scenario: Visitors on your website frequently have multiple related questions about products and solutions. The default End of conversation prompt can disrupt their experience by forcing them into providing feedback or acknowledgments prematurely. By customizing the End of conversation topic, you'll enable a more fluid interaction, allowing users to naturally continue conversations without friction or interruption.

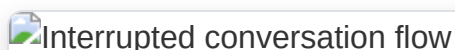
Step-by-step instructions

1. Let's start with a test. Refresh the test pane and ask two questions consecutively.

What is Copilot Studio?

What knowledge sources does it support?

Notice how the default End of conversation topic interrupts the interaction, preventing the second question from being answered until the user responds to the prompt, "Did that answer your question?"

 Interrupted conversation flow

Interrupted conversation flow

2. Go to the **Topics** tab, display **All**, and select **End of conversation**.

On the question “Did that answer your question?”, select the ellipsis (...) and open properties.

Go to **Question behavior**. Set **How many reprompts** to **Don't repeat**.

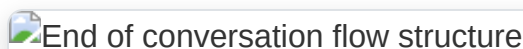
Return to the Question Properties, then select **Entity recognition**: For **Action if no entity found**, choose **Set variable to empty (no value)**.

- Below the “Did that answer your question?” question, notice the condition only tests if SurveyResponse is true. Let's add another condition path, by clicking on the (+) action above the various conditions.

In **Select a variable**, choose **SurveyResponse**. For the test, leave it to **is equal to**, and set **false** for the value.

Move everything that is under **All other conditions** under the new false branch by cutting and pasting the content.

- Add a redirect to the **Conversational boosting** topic under the **All other conditions** path.

A screenshot of a dialog box titled "End of conversation flow structure". The dialog box has a light gray background and a thin border. It contains the text "End of conversation flow structure" in a dark gray font.

End of conversation flow structure

- Save your topic.
- Let's do a new test. Refresh the test pane and ask again two questions consecutively.

What is Copilot Studio?

What knowledge sources does it support?

Notice how follow-up questions are no longer blocking the conversation flow.



Tip:

If you want the user's follow-up questions to trigger existing topics rather than always defaulting to Conversational boosting, verify the Interruptions setting in the "Did that answer your question?" question properties. By default, they allow interruptions, meaning the agent can seamlessly switch to a recognized topic based on the user's next input.

7. You can further simplify what happens after the user answers **Yes**.

After the CSAT question, add a message node asking:

Thank you for your feedback!

Feel free to ask me something else.

Then add a node **Topic management > End conversation**.

Delete everything further below that path.

8. You can further simplify what happens after the user answers **No**.

Under the **SurveyResponse false** condition path, add a new message node:

Sorry I wasn't able to help better.

You may try reaching out to our [Microsoft Copilot Studio community] (<https://aka.ms/CopilotStudioCommunity>) or submitting a [support request] (<https://learn.microsoft.com/en-us/power-platform/admin/get-help-support>).

Would you like to try again? Feel free to ask a new question.

Then add a node **Topic management** > **End conversation**.

Save topic.

Congratulations! You've completed Use Case #2!

Test your understanding

- How does customizing the End of Conversation topic improve user experience without sacrificing data collection?
- What are the trade-offs between gathering comprehensive feedback and maintaining conversation flow?
- How can you apply these principles to create more natural conversation endings in your own agents?

Challenge: Apply this to your own use case

- Where in your current agent can users experience unnecessary conversational friction?

- How can you apply these customization strategies to ensure smoother conversation endings?
 - In what ways can user feedback be naturally integrated without interrupting conversational flow?
-

Summary of learnings

True learning comes from doing, questioning, and reflecting—so let's put your skills to the test.

To maximize the impact of conversation outcome tracking:

- **Structure conversations for clear outcomes** – Every interaction should have a definitive end point that can be measured and analyzed for continuous improvement.
 - **Balance user experience with data collection** – Gather meaningful feedback without creating friction that drives users away from your agent.
 - **Implement session-based tracking** – Understand that individual queries within conversations have separate outcomes that provide granular insights.
 - **Leverage both implicit and explicit resolution** – Track when users are satisfied enough to leave (implicit) versus when they explicitly confirm satisfaction.
 - **Design for different orchestration modes** – Ensure your tracking works effectively whether using classic topics or generative AI responses.
-

Conclusions and recommendations

Conversation analytics golden rules:

- Always redirect to the End of Conversation topic when user requests are fulfilled to ensure accurate outcome tracking.
- Customize feedback prompts to match your users' communication style and reduce abandonment rates.
- Test conversation flows regularly to identify where users experience friction or confusion.
- Use analytics insights to identify knowledge gaps and refine content strategy continuously.
- Balance comprehensive data collection with user experience to maintain high engagement levels.
- Implement both lightweight feedback (thumbs up/down) and detailed feedback collection for comprehensive insights.

By following these principles, you'll transform your AI agent from a simple response tool into a data-driven system that continuously learns and improves, delivering measurably better user experiences and business outcomes.
