DUMPSBOSS.

Agentic AI Business Solutions Architect

Microsoft AB-100

Version Demo

Total Demo Questions: 8

Total Premium Questions: 88

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Fabrikam, Inc	2
Topic 2, Contoso. Ltd	8
Topic 3, Mix Questions	78
Total	88

QUESTION NO: 1 - (HOTSPOT)

A company deploys agents that generate responses by using Azure OpenAI resources. The agents are deployed to both the United States and Europe.

You need to recommend a governance solution that meets the following requirements: Enforces the deployment of the resources to only approved Azure regions

Provides continuous compliance verification of the resources

ANSWER:

Explanation:

You're being asked to pick the right governance tools for two different jobs: (1) stopping noncompliant deployments up front, and (2) continuously checking compliance after resources exist. For the first requirement—**enforcing deployment only to approved Azure regions**—the correct choice is **Azure Policy**. Azure Policy is the native "guardrail" service that can deny or restrict resource creation based on rules. A classic example is the built-in policy definition *Allowed locations*, which blocks deployments outside a list of approved regions when assigned at a management group, subscription, or resource group scope. That's exactly what you need when agents and Azure OpenAI resources must only be deployed in specific geographies for residency or regulatory reasons. See: Azure Policy overview and the built-in policy definitions (including location restrictions).

For the second requirement—**continuous compliance verification**—the best match is **Microsoft Defender for Cloud**. Defender for Cloud provides continuous security posture management (CSPM) and compliance tracking, including dashboards, recommendations, and regulatory compliance views that help you verify over time whether resources remain aligned with required standards. This is different from Azure Policy's "deny at deployment" behavior; Defender for Cloud is focused on ongoing assessment and visibility across your environment. Reference: Microsoft Defender for Cloud introduction and Regulatory compliance dashboard.

So, the correct hotspot selections are Azure Policy for region enforcement, and Microsoft Defender for Cloud for continuous compliance verification.

QUESTION NO: 2

A company has two Microsoft Power Platform environments named Devi and Prodi. A Microsoft

Copilot Studio agent named Agent1 is built into a solution in the Devi environment. You plan to deploy Agent1 to Prodi.

You need to make Agent1 available to the users in Prodi. The solution must minimize administrative effort.

What should you do?

A. Export the solution as a managed solution and import the solution into Prodi.

B. Export the solution as an unmanaged solution and import the solution into Prodi.

C. Create a new Copilot Studio agent in Prodi by replicating the configuration of Agent1.

D. Share Agentl with the users in Prodi.

ANSWER: A

Explanation:

To make Agent1 available in the Prodi environment with minimal administrative effort, you should use standard Power Platform ALM: package the agent in a solution and deploy it to the target environment. In production, Microsoft's recommended practice is to import *managed* solutions. A managed solution provides a controlled, governed deployment artifact that can be updated via solution upgrades and helps prevent accidental edits directly in production. This aligns with the requirement to minimize ongoing admin effort and reduce drift between environments.

Option B (unmanaged) is typically used in development because it remains editable and is not ideal for controlled production deployments. Option C (recreating the agent manually) increases effort, is error-prone, and breaks repeatable ALM. Option D (sharing) doesn't deploy anything across environments—sharing only affects access within the same environment and won't move the solution/agent from Devi to Prodi.

References: Solution concepts for ALM (Power Platform), Import, update, and export solutions.

QUESTION NO: 3

Which two components in the custom AI agent design should the CFO evaluate in the quarterly agent analysis? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point

A. the GPT models used for the agent

the GPT models used for the agent
Model choice can matter technically, but the CFO's quarterly Copilot Studio estimator-based analysis is more centered on usage and orchestration behavior than on model selection details.

B. the average characters in a chat message

the average characters in a chat message
This is too granular and is not a primary business-facing metric for ROI or adoption analysis.

C. the average session time per agent

the average session time per agent is correct:
Average session time is a practical usage and adoption indicator. It helps the CFO understand how much users are engaging with the agent and whether the agent is reducing effort efficiently or creating long, costly interactions.
Why

D. the agent orchestration method
the agent orchestration method is correct:
The orchestration method affects how the agent handles requests, invokes tools, uses knowledge sources, and consumes resources. Since this can influence both operational efficiency and cost, it is important for ROI analysis.
Why the other options are not the best fit:

ANSWER: C D

Explanation:

For a quarterly CFO review, the focus is typically on business outcomes (adoption/usage and cost drivers) rather than low-level message characteristics. In Microsoft Copilot Studio, analytics and the usage/cost estimation concepts are most meaningfully tied to how often and how long users engage with the agent, and what the agent does during those interactions (for example, whether it invokes actions/tools, uses generative answers, or routes across topics). Therefore, **average session time per agent** is a useful adoption/efficiency indicator: it helps quantify engagement and can hint at whether the agent is resolving issues quickly or causing longer, more expensive conversations. The **agent orchestration method** is also relevant because orchestration choices influence runtime behavior (how requests are routed, when tools/actions are called, and how responses are generated), which directly affects operational cost and efficiency—key inputs to ROI.

The **GPT models used** can matter technically, but CFO-level quarterly analysis in Copilot Studio is usually not driven by model-selection details as a "design component" to review for ROI; it's more about usage and how the agent operates. **Average characters per message** is too granular and not a standard KPI for adoption or ROI in Copilot Studio reporting.

References: Microsoft Copilot Studio analytics, Microsoft Copilot Studio documentation.

QUESTION NO: 4

A company uses Microsoft Dynamics 365 to manage service operations. Dispatchers coordinate service requests, and technicians perform scheduled on-site work.

You need to design a solution that will use Microsoft Copilot to improve the efficiency of the service operations. The solution must meet the following requirements:

Provide AI-driven assistance to help staff organize and resolve work orders.

Deliver contextual AI support to frontline workers as they prepare for and complete customer appointments.

Which two components should you include in the design? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Copilot in Customer Service

Copilot in Customer Service is focused more on customer support agents than on field dispatch/service execution.

B. Copilot in Outlook

Copilot in Outlook is too generic and not purpose-built for field service operations.
E . Dynamics 365 Customer Service is not the primary app for technician appointment execution.
F . Copilot Service workspace is more aligned with service agents in customer support environments rather than frontline field technicians.

C. Copilot in Field Service

Copilot in Field Service is correct:
Copilot in Field Service is designed to help service teams work more efficiently with work orders, scheduling context, task assistance, and service-related operational support. This matches the
requirement to provide AI-driven assistance to help staff organize and resolve work orders. Why

D. the Dynamics 365 Field Service mobile app

the Dynamics 365 Field Service mobile app is correct:
Frontline workers and technicians use the Field Service mobile app while preparing for and completing appointments. That is

the right surface for delivering contextual AI support in the flow of field work.
Why the other options are not the best fit:

E. Dynamics 365 Customer Service

F. Copilot Service workspace

ANSWER: C D

Explanation:

To meet both requirements, you need Copilot capabilities that are purpose-built for Dynamics 365 Field Service and available where each persona works. **Copilot in Field Service** is the right component for dispatchers and service operations staff because it's designed to help with work-order-centric activities such as summarizing work orders, assisting with next steps, and improving productivity in the Field Service experience. This directly aligns to "organize and resolve work orders."

For technicians, the best way to deliver "contextual AI support" is through the **Dynamics 365 Field Service mobile app**, which is the primary frontline surface for preparing for appointments, reviewing customer and asset history, and completing service tasks in the field. Using the mobile app ensures Copilot assistance is available in the flow of work during on-site execution.

Options like **Copilot in Customer Service**, **Dynamics 365 Customer Service**, and **Copilot Service workspace** are optimized for contact-center/service-agent scenarios rather than field dispatch and technician execution. **Copilot in Outlook** is generic productivity assistance and doesn't provide the Field Service work-order/appointment context required here.

References: Dynamics 365 Field Service overview, Field Service mobile app

QUESTION NO: 5

You are evaluating a Microsoft Copilot Studio agent that supports Microsoft Dynamics 365 Customer Service representatives.

You need to recommend a testing solution that meets the following requirements: Evaluates agent effectiveness during active sessions

Validates whether the agent delivers accurate and helpful responses Provides measurable, actionable insights for continuous improvement What should you recommend?

A. Track resolution, deflection, and accuracy by using dashboards and use scripts to ensure consistent responses.

Track resolution, deflection, and accuracy by using dashboards and use scripts to ensure consistent responses.
This question is about evaluating a Copilot Studio agent in live support operations, not just testing technical uptime or infrastructure performance. The requirements emphasize three things: effectiveness during active sessions
response accuracy and helpfulness
measurable insights for continuous improvement
That combination points to operational quality metrics and analytics dashboards. Why A is correct
Tracking resolution, deflection, and accuracy directly measures how well the agent performs in real support conversations:
Resolution shows whether the issue is successfully handled
Deflection shows whether the agent reduces human workload appropriately Accuracy shows whether responses are correct and helpful
Using dashboards gives leaders and support teams measurable, ongoing visibility into agent behavior. Adding scripts for consistent testing further supports repeatable evaluation and improvement.
From an AI business solutions perspective, this is the right recommendation because it combines: business outcome measurement

quality validation operational analytics
continuous improvement feedback loops
This is exactly how enterprise copilots should be managed after deployment. Why the other options are incorrect

B. Perform load testing to validate how the agent scales under a high chat volume.

Perform load testing to validate how the agent scales under a high chat volume
Load testing is useful for scalability and capacity planning, but it does not directly validate whether responses are accurate, helpful, or effective during active sessions from a business-outcome perspective.

C. Review historical tickets to find agents that have the shortest resolution times.

Review historical tickets to find agents that have the shortest resolution times
This may give some retrospective insight, but it does not directly evaluate the Copilot Studio agent during active sessions, and shortest resolution time alone does not prove response quality or helpfulness.

D. Measure uptime and page load times.

Measure uptime and page load times
These are infrastructure and availability metrics. They are important for system health, but they do not evaluate conversational effectiveness or answer quality.
Expert reasoning
For Copilot evaluation questions:
if the goal is business effectiveness in active sessions, use resolution/deflection/accuracy if the goal is system scale, use load testing
if the goal is infrastructure reliability, use uptime and latency

ANSWER: A

Explanation:

The best recommendation is to use operational conversation analytics/metrics (for example, resolution rate, deflection/containment, and accuracy) surfaced in dashboards, and to complement that with repeatable test scripts. The key requirement is evaluating effectiveness *during active sessions* and producing measurable insights you can act on. In Copilot Studio, you can monitor conversations and outcomes over time (sessions, engagement, escalation/transfer, containment/deflection, CSAT where available) and use those signals to identify gaps, retrain/refine topics, and improve knowledge sources. Dashboards make this continuous and measurable, while scripted test conversations help you validate that changes actually improve response quality and reduce regressions.

Load testing (B) is about capacity and performance under volume; it doesn't tell you whether answers are correct or helpful. Reviewing historical tickets for shortest resolution time (C) is indirect and can be misleading—fast doesn't mean accurate, and it doesn't evaluate the agent's in-session behavior. Uptime/page load (D) are availability metrics; important, but they don't measure conversational quality or business outcomes.

References: Microsoft Copilot Studio analytics, Test your copilot (Copilot Studio).

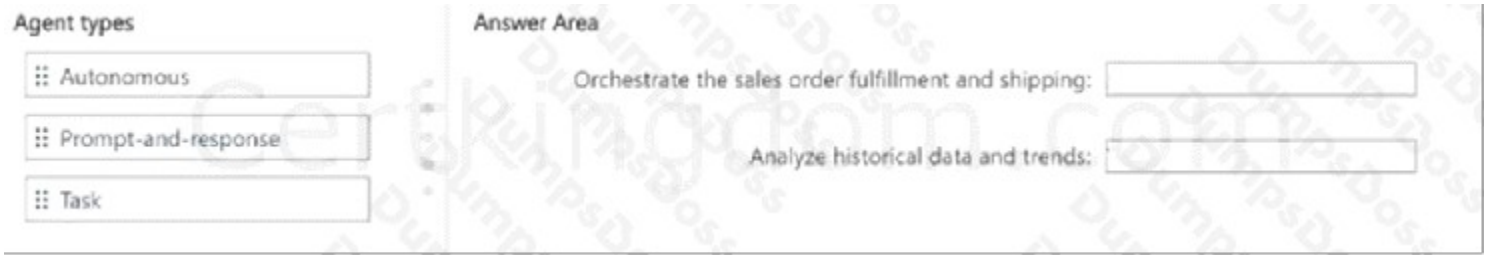
QUESTION NO: 6 - (DRAG DROP)

A company plans to implement an AI business solution for a consumer goods company. You need to create agents that meet the following requirements:

Orchestrate the sales order fulfillment and shipping of goods to customers. Analyze historical data and trends to replenish stock.

Which type of agent should you use for each requirement? To answer, drag the appropriate agent types to the correct requirements. Each agent type may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.



ANSWER:

Explanation:

You're basically being asked to match the agent type to the kind of work being done: end-to-end execution vs. a focused, bounded activity.

Orchestrating sales order fulfillment and shipping is a classic multi-step business process. It usually involves coordinating actions across multiple systems (ERP/order management, inventory, warehouse operations, shipping carriers, customer notifications) and handling exceptions (out-of-stock, address issues, shipment delays). That's exactly where an **Autonomous agent** fits best: it can plan and execute a sequence of steps, make decisions as conditions change, and keep the process moving with minimal user prompting. In Microsoft's framing, autonomous agents are intended to run workflows and take actions, not just answer questions.

Analyzing historical data and trends to replenish stock is more narrowly scoped. The goal is to perform analysis (forecasting/identifying trends) and produce an output such as a replenishment recommendation. That aligns better with a **Task agent**, which is designed to complete a specific job or function (for example, "analyze these sales trends and recommend reorder quantities"). It's not primarily about orchestrating a long-running operational workflow; it's about completing a defined analytical task.

Prompt-and-response agents are best when the user is directly asking questions and getting answers (interactive Q&A). While you could ask it to "analyze trends," the requirement here is framed as a business capability (replenishment analysis), which is better represented as a task-focused agent rather than a pure chat/Q&A pattern.

References: Microsoft Copilot Studio documentation and Copilot Studio release plan (agents capabilities).

QUESTION NO: 7 - (DRAG DROP)

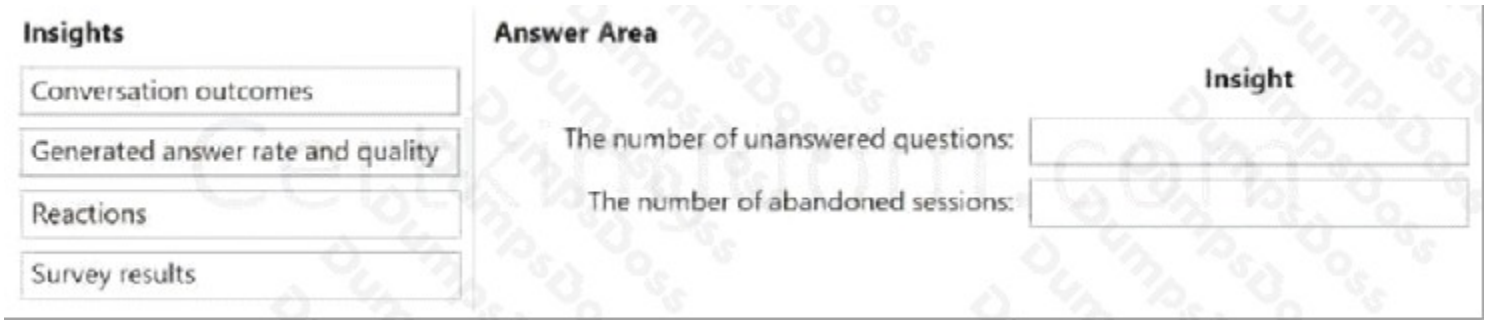
A company has a Microsoft Copilot Studio agent that has been in production for three months. The agent has received positive feedback from users.

You need to identify the number of questions unanswered by the agent and the number of

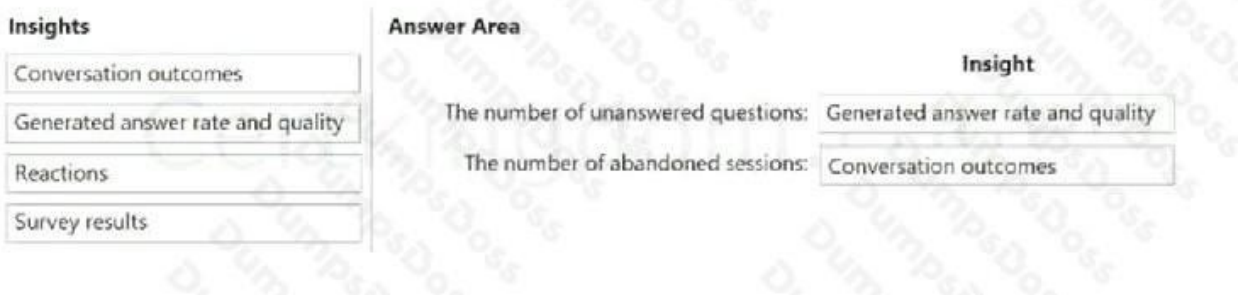
abandoned sessions between the users and the agent.

Which Copilot Studio insights should you use? To answer, drag the appropriate insights to the correct requirements. Each insight may be used once, more than once, or not at all.

NOTE: Each correct selection is worth one point.



ANSWER:



Explanation:

You're being asked for two very specific operational metrics: how many user questions the agent didn't answer, and how many user sessions were abandoned. In Copilot Studio, those two metrics live in different "Insights" views because they measure different things: one is answer-generation performance, and the other is how entire conversations end.

Unanswered questions are best measured using **Generated answer rate and quality**. This insight is designed to show how often the agent successfully generates an answer (and how good those answers are). When the agent can't produce an answer (for example, it fails to find grounding content, can't generate a response, or returns a fallback), that behavior shows up as a reduced generated answer rate and is the closest built-in insight for quantifying "unanswered" questions. This is the insight you'd use to spot knowledge gaps and coverage issues in your agent's content and configuration. See: Copilot Studio analytics and insights.

Abandoned sessions are conversation-level outcomes, so you should use **Conversation outcomes**. This insight categorizes how conversations conclude (for example, resolved vs. abandoned). If a user leaves mid-conversation or the interaction ends without completion, it's tracked as an abandonment outcome, which is exactly what the requirement asks you to count. This aligns with how Copilot Studio reports conversation completion and termination patterns. See: Conversation outcomes (analytics).

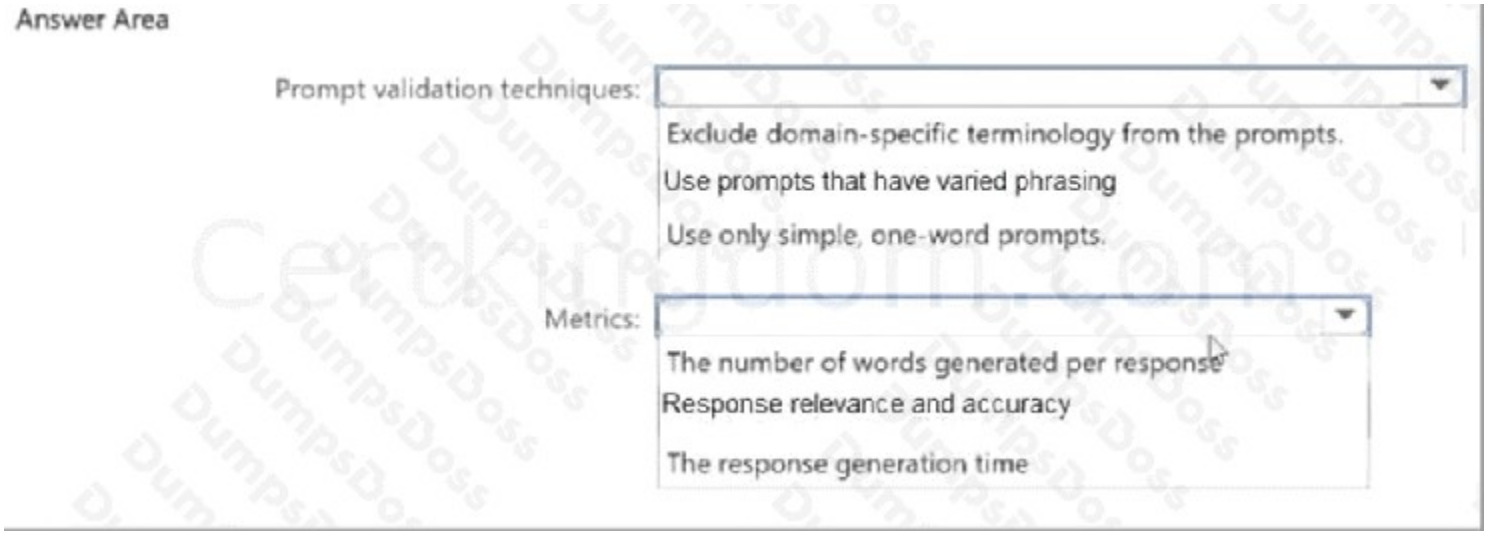
Reactions and **Survey results** are feedback mechanisms (thumbs up/down, survey responses). They're useful for sentiment and satisfaction, but they don't directly provide counts of unanswered questions or abandoned sessions, so they aren't the right choices here.

QUESTION NO: 8 - (HOTSPOT)

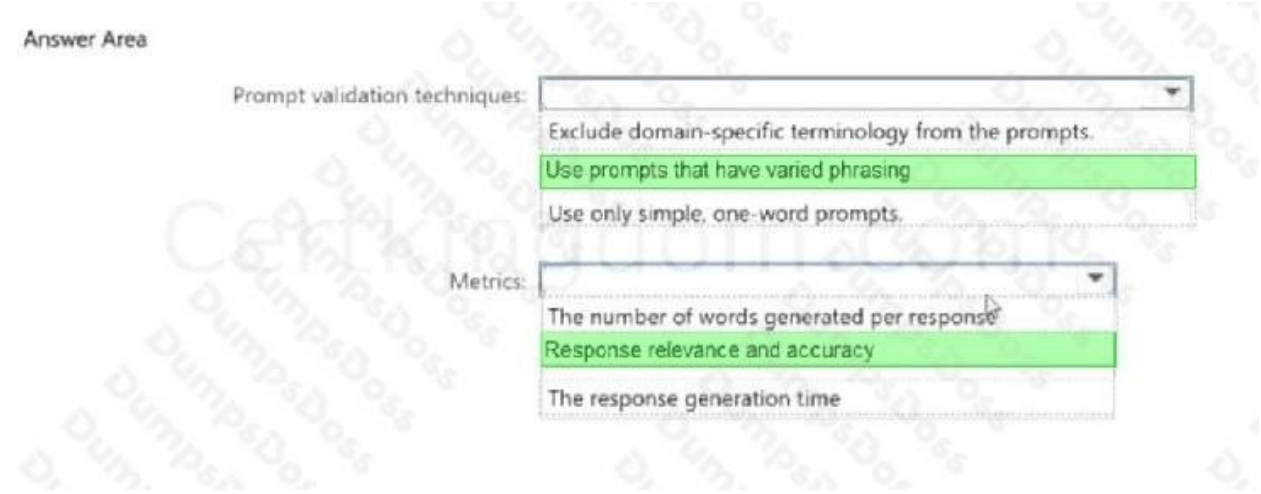
You are designing a testing solution for Microsoft Copilot Studio agents.

You need to validate prompt engineering best practices to ensure that the agents generate accurate and contextually relevant responses. Which prompt validation techniques and metrics should you include in the solution? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.



ANSWER:



Explanation:

To validate prompt engineering best practices for Copilot Studio agents, you want a test approach that mirrors how real users actually interact with the agent and a metric that directly measures whether the agent's output is correct and on-topic. In practice, users rarely ask questions using one fixed sentence. They'll rephrase, use synonyms, omit details, or add extra context. That's why the best prompt validation technique in the dropdown is **Use prompts that have varied phrasing**. This technique checks whether your instructions, grounding content, and conversation design are robust to natural language variability and still produce the intended behavior across multiple utterance styles.

For the metric, the requirement is explicitly about ensuring responses are *accurate* and *contextually relevant*. The metric that directly evaluates that is **Response relevance and accuracy**. This aligns with common LLM evaluation practices where you judge whether the answer addresses the user's intent, is factually correct (especially when grounded on enterprise data), and stays within the right context and constraints.

The other options don't fit the goal. **Exclude domain-specific terminology** is generally the opposite of what you want in business scenarios—domain terms are often necessary for precision and correct grounding. **Use only simple, one-word prompts** is not representative of real usage and won't validate instruction-following or contextual reasoning. On the metrics side, **number of words** is not a quality indicator, and **response generation time** is a performance/latency measure rather than a prompt-quality validation measure.

For more on testing and improving Copilot Studio agent behavior and quality, see Microsoft's Copilot Studio documentation: Microsoft Copilot Studio documentation and guidance on generative answers: Generative answers in Copilot Studio.