Monitor Performance and Evaluate Agent Quality

Learn how to use analytics to measure agent performance, create evaluation test sets to systematically assess agent quality, and run evaluations to drive continuous improvement.

🧭 Lab Details

Level	Persona	Duration	Purpose
200	Maker	30 minutes	After completing this lab, participants will be able to access and interpret agent analytics including conversation volume, topic performance, and user satisfaction, create evaluation test sets using auto-generation, CSV import, test canvas capture, and manual entry, configure evaluation methods including exact match, keyword match, similarity, general quality, and compare meaning, and review evaluation results to compare runs and identify improvement opportunities.

📚 Table of Contents

🤔 Why This Matters

An agent you can't measure is impossible to improve, and without systematic evaluation, you're guessing about quality.

Think of analytics and evaluations like running a quality assurance program: - **Without analytics**: You have no idea how many users interact with your agent, what they ask, or whether they leave satisfied - you're running blind - **Without evaluations**: You make changes to your agent and hope they help, but you have no systematic way to verify improvements or catch regressions - **With both**: You have data-driven insights to identify issues AND a repeatable testing framework to validate fixes - a continuous improvement cycle that ensures your agent gets better over time

Common challenges solved by this lab: - “I can’t tell if my agent is actually helping users” - “I don’t know which topics are failing or which questions users can’t get answered” - “I made changes to my agent but I’m not sure if they improved things” - “I need a systematic way to test my agent’s quality before deploying updates”

In 30 minutes, you’ll learn how to measure your agent’s performance with analytics and build a repeatable evaluation framework for continuous quality improvement.

🌐 Introduction

Analytics and evaluations are the two pillars of agent quality management. Analytics provide real-time visibility into how users interact with your agent - tracking conversation volumes, topic performance, satisfaction scores, and failure patterns. Evaluations provide a structured testing framework where you define expected responses, run systematic tests, and compare results across agent versions. Together, they create a complete quality management system: analytics tells you WHERE to improve, and evaluations tell you WHETHER your improvements actually worked.

Real-world example: A company deploys an IT support agent and uses analytics to discover that 30% of conversations about VPN setup end in user abandonment. They add VPN documentation to the agent’s knowledge sources, then create an evaluation test set with 15 VPN-related questions and expected answers. After the knowledge update, they run the evaluation and confirm that pass rates improved from 40% to 90%. A week later, analytics confirm that VPN topic abandonment dropped from 30% to 5%. The combination of analytics (finding the problem) and evaluations (verifying the fix) created a measurable, data-driven improvement cycle.

This lab teaches you how to use both analytics and the Agent Evaluation feature (preview) to build a quality management practice that ensures your agents continuously improve.

🎓 Core Concepts Overview

Concept	Why it matters
Conversation Analytics	Metrics tracking conversation volume, topic usage, and session duration - these reveal how users interact with your agent and which capabilities they value most
User Satisfaction Scores	Feedback metrics measuring whether users found the agent helpful - low satisfaction scores indicate areas needing improvement in knowledge, instructions, or conversation flows
Failure Analytics	Data showing where conversations fail, which questions go unanswered, and where users escalate or abandon - these insights directly guide improvement priorities
Evaluation Test Sets	Collections of test cases with questions and expected responses that systematically verify agent quality - these provide repeatable, objective quality measurement
Evaluation Methods	Different comparison techniques (Exact Match, Keyword Match, Similarity, General Quality, Compare Meaning) that determine how agent responses are assessed against expected answers
Evaluation Results	Pass/fail outcomes with detailed reasoning, knowledge citations, and activity maps - these reveal exactly why an agent succeeded or failed on each test case

📄 Documentation and Additional Training Links

- Analyze agent performance (https://learn.microsoft.com/microsoft-copilot-studio/analytics-overview)
- Use conversation analytics (https://learn.microsoft.com/microsoft-copilot-studio/analytics-summary)
- Agent Evaluation overview (https://learn.microsoft.com/microsoft-copilot-studio/analytics-agent-evaluation-overview)
- Create evaluation test sets (https://learn.microsoft.com/microsoft-copilot-studio/analytics-agent-evaluation-create)
- View and interpret evaluation results (https://learn.microsoft.com/microsoft-copilot-studio/analytics-agent-evaluation-results)

✅ Prerequisites

- Completed Build Intelligent Agents with Knowledge Sources, Tools, and Topics and Master Variables, Multi-Agent Architectures, and Channel Deployment labs - you need a deployed agent with conversation history for meaningful analytics
- Access to Microsoft Copilot Studio with analytics and evaluation permissions
- An agent that has been deployed and used (even test conversations count for analytics data)

🎯 Summary of Targets

In this lab, you'll use analytics to understand agent performance and build evaluation test sets to systematically measure and improve quality. By the end of the lab, you will:

- Access and interpret conversation analytics including volume, engagement, and topic metrics
- Analyze user satisfaction scores and identify improvement opportunities
- Use failure analytics to discover knowledge gaps and conversation issues
- Generate evaluation test sets automatically and import test cases from CSV files
- Capture test cases from real agent conversations using the test canvas
- Add manual test cases to target specific agent capabilities
- Configure evaluation methods appropriate for different test scenarios
- Review evaluation results including pass rates, reasoning, and activity maps
- Compare evaluation runs and export results for reporting

🧩 Use Cases Covered

Step	Use Case	Value added	Effort
1	Monitor Agent Performance with Analytics	Measure agent performance and identify optimization opportunities using conversation data	10 min
2	Create and Configure Evaluation Test Sets	Build systematic test cases to objectively measure agent quality with multiple evaluation methods	10 min
3	Review Evaluation Results	Interpret evaluation outcomes and use results to drive measurable agent improvements	10 min

🛠️ Instructions by Use Case

🧱 Use Case #1: Monitor Agent Performance with Analytics

Learn how to access agent analytics, interpret performance metrics, identify improvement opportunities, and use data to optimize agent experiences.

Use case	Value added	Estimated effort
Monitor Agent Performance with Analytics	Measure agent performance and identify optimization opportunities using conversation data	10 minutes

Summary of tasks

In this section, you'll learn how to access the analytics dashboard, understand conversation volume and topic metrics, analyze user satisfaction scores, identify failure patterns, and use insights to improve your agent.

Scenario: Your agent has been deployed for several days. You need to understand how users are interacting with it, which topics are most popular, where conversations are failing, and whether users find it helpful. Analytics provide the insights needed to prioritize improvements.

Objective

Access and interpret agent analytics to measure performance and identify optimization opportunities.

Step-by-step instructions

Navigate to Analytics

1. In your Copilot Studio agent, click **Analytics** in the left navigation panel.
2. Review the analytics dashboard overview, which typically includes:
    - **Summary metrics**: Total conversations, engaged conversations, resolution rate
    - **Trend charts**: Conversation volume over time

- **Topic performance**: Which topics are used most frequently
- **User satisfaction**: Feedback scores from users

> ℹ️
> **Note:** Analytics data may take 24-48 hours to populate for new agents. If your agent is brand new, you may see limited or no data initially.

3. Set the date range using the date picker (typically in the top right):
   - Last 7 days
   - Last 30 days
   - Custom date range

> 💡
> **Tip:** Use consistent date ranges when comparing performance over time. Weekly reviews with 7-day ranges work well for ongoing monitoring.

Understand Summary Metrics

4. Review the **Total conversations** metric:
   - This shows how many conversation sessions occurred
   - A conversation typically starts when a user sends their first message and ends after a period of inactivity
5. Review the **Engaged conversations** metric:
   - Conversations where the user sent more than one message
   - Higher engagement suggests users are finding value and continuing the conversation
6. Calculate the **Engagement rate** (if not shown automatically):

- Engagement Rate = Engaged Conversations / Total Conversations x 100%
- Higher engagement rates indicate the agent is providing valuable assistance

> ℹ️
> **Note:** Low engagement might mean users get answers immediately (good) or give up after the first response (bad). Combine this metric with satisfaction scores for proper interpretation.

Analyze Conversation Volume Trends

7. Review the **Conversation volume** chart showing conversations over time.
8. Look for patterns and trends:
   - **Peaks**: When is demand highest? Plan capacity accordingly
   - **Valleys**: When is usage lowest? Schedule maintenance during these times
   - **Trends**: Is usage growing over time? Flat lines may indicate awareness issues
9. Consider external factors that might influence trends:
   - Business cycles (month-end, quarter-end)
   - Seasonal patterns
   - Recent communications or awareness campaigns

> 💡
> **Tip:** Share positive growth trends with stakeholders to demonstrate adoption and value. Use declining trends as signals to refresh content or increase awareness.

Review Topic Performance

10. Navigate to the **Topics** section in analytics (may be a separate tab or section).
11. Review the **Topic usage** metrics showing:
    - Which topics are triggered most frequently
    - How many conversations used each topic
    - Average conversation duration per topic
12. Identify your **top topics**:
    - These represent the most common user needs
    - Ensure these topics have excellent quality and comprehensive coverage
    - Consider creating child agents for highly-used complex topics
13. Identify **unused or rarely used topics**:
    - These may indicate topics that don't match real user needs
    - Consider removing or consolidating rarely-used topics
    - Investigate if trigger phrases need improvement

> ⚠️
> **IMPORTANT:** Not all topics should be equally used. Focus optimization efforts on high-volume topics that impact the most users.

Analyze User Satisfaction

14. Navigate to the **User satisfaction** section in analytics.
15. Review satisfaction metrics:
    - **Overall satisfaction score**: Percentage of users who rated the agent positively

- **Satisfaction by topic**: Which topics have high/low satisfaction
- **Feedback comments**: Qualitative feedback from users (if collected)

16. Identify topics with **low satisfaction scores**:
    - These are your highest priority for improvement
    - Common causes: incomplete knowledge, poor instructions, confusing flows
17. Investigate patterns in negative feedback:
    - Are users frustrated by specific types of questions?
    - Do certain conversation flows consistently result in low satisfaction?
    - Are there knowledge gaps or outdated information?

> 💡
> **Tip:** A single low-satisfaction topic with high volume should be your top improvement priority - it affects many users and has measurable impact.

Review Escalation and Abandonment

18. Look for **Escalation rate** metrics (if available):
    - How often users request human assistance
    - High escalation rates indicate the agent can’t handle common scenarios
19. Review **Abandonment metrics**:
    - Conversations where users leave without resolution
    - Points in the conversation where users drop off most frequently
20. Analyze **Unrecognized phrases** or **Unanswered questions** (if available):
    - Questions the agent didn’t understand or couldn’t answer
    - Direct indicators of knowledge gaps or missing topics

⚠️

IMPORTANT: Unrecognized phrases are gold mines for improvement. Each represents a real user need your agent isn't addressing. Add knowledge or topics to cover these gaps.

Identify Improvement Opportunities

21. Based on your analytics review, create a prioritized list of improvements:
    - **High-volume, low-satisfaction topics**: Improve knowledge or instructions
    - **Unrecognized phrases**: Add missing knowledge or create new topics
    - **High abandonment points**: Simplify conversation flows or add clarifying messages
22. Document specific actions for each improvement:
    - "Add licensing FAQ to knowledge sources" (addresses 15% of unanswered questions)
    - "Simplify mailing list topic flow" (reduce 3-step process to 2 steps)
    - "Update agent instructions" (clarify when to use child agents)

💡

Tip: Always measure the impact of improvements by comparing analytics before and after changes. This validates your efforts and informs future optimization.

🏅 Congratulations! You've completed Use Case 1!

Test your understanding

Key takeaways:

- **Analytics Reveal User Behavior** – Conversation volume, engagement rates, and topic performance show how users actually interact with your agent
- **Satisfaction Scores Guide Priorities** – Low satisfaction on high-volume topics should be your top improvement priority
- **Failure Data Drives Improvements** – Unrecognized phrases and abandoned conversations directly indicate where to add knowledge or refine flows

Lessons learned & troubleshooting tips:

- Analytics data takes 24-48 hours to populate - don't expect instant results for new agents
- Combine multiple metrics for accurate interpretation (e.g., low engagement + high satisfaction = users getting quick answers)
- Set up regular analytics review cadence: weekly for new agents, bi-weekly for mature agents

Challenge: Apply this to your own use case

- What satisfaction score would indicate success for your agent?
- How often should you review analytics based on your conversation volume?
- What metrics would you share with leadership to demonstrate agent value?

🔄 Use Case #2: Create and Configure Evaluation Test Sets

Learn four different ways to create evaluation test sets - auto-generation, CSV

import, test canvas capture, and manual entry - to systematically measure your agent’s quality.

Use case	Value added	Estimated effort
Create and Configure Evaluation Test Sets	Build systematic test cases to objectively measure agent quality with multiple evaluation methods	10 minutes

Summary of tasks

In this section, you’ll learn how to generate test cases automatically, import test cases from a CSV file, capture test cases from the test canvas, and add manual test cases - giving you a complete toolkit for building comprehensive evaluation coverage.

Scenario: You want to systematically test your Copilot Studio Assistant using different approaches. You’ll create three distinct test sets - one auto-generated, one imported from CSV that is intentionally designed to fail, and one captured from real agent conversations that should pass - to see how different creation methods and evaluation outcomes work.

> ℹ️
> **Note: Preview Feature:** Agent Evaluation is currently a preview feature in Copilot Studio. Features and UI may change as Microsoft iterates on the experience. Preview features are not intended for production use.

Objective

Create evaluation test sets using four different methods and understand how each approach serves different testing needs.

Step-by-step instructions

Generate Test Cases

1. In your Copilot Studio agent, click **Evaluation** in the left navigation panel.

> ℹ️
> **Note:** If you don't see the Evaluation option, it may need to be enabled in your environment settings or may not yet be available in your region. Check Agent Evaluation overview (https://learn.microsoft.com/microsoft-copilot-studio/analytics-agent-evaluation-overview) for availability.

2. Click **Create Test Set**.
3. Click **Generate 10 questions**. Copilot Studio will use AI to automatically generate test cases based on your agent's knowledge sources and configuration.
4. Change the test set name to:

```
Non-Critical Copilot Studio Guide Set
```

5. Set the **User Profile** by selecting **Logged on user (Admin)**.
6. Click **Save**.
7. Click on the first generated question to explore all the available options. For each test case, you can configure the **evaluation method**:
    - **Exact Match**: Character-for-character comparison between expected and actual response. Use for questions with precise, factual answers.
    - **Keyword Match**: Checks whether key terms from the expected response appear in the actual response. Use when exact wording doesn't matter but

key concepts must be present.

- **Similarity**: Uses cosine similarity to compare semantic meaning on a 0-1 scale with a configurable threshold. Use when meaning matters more than exact wording.
- **General Quality**: Uses an LLM to evaluate response quality across four dimensions - relevance, groundedness, completeness, and abstention. Does NOT require an expected response. Use for open-ended questions.
- **Compare Meaning**: Evaluates whether the intent and meaning of the actual response matches the expected response, with a configurable threshold. Use for semantic comparison with more nuance than cosine similarity.

> 💡
> **Tip:** Choose evaluation methods that match the nature of each question. Factual questions with precise answers work well with Exact Match or Keyword Match. Open-ended questions benefit from General Quality or Similarity methods.

8. Click **Evaluate** to start the evaluation of this test set.

> ℹ️
> **Note:** Evaluation time depends on the number of test cases and agent response time. A test set with 10 cases typically completes in 1-3 minutes.

Import Test Cases

9. Click **Create a test set**.
10. Note the link to **download the CSV template**. Click it to review the required CSV format. The template shows the expected columns:

- **Question** - The user question that the agent will answer
- **Expected response** - The expected answer to evaluate against
- **Testing method** - The evaluation method to use for the test case

11. Download the EvaluationAlwaysFail.csv (EvaluationAlwaysFail.csv) file provided with this lab. This CSV contains adversarial test cases designed to verify your agent properly handles harmful or inappropriate requests. Import the file into the test set.

> 💡
> **Tip:** File import is useful when you have a large number of test cases or want to maintain test cases in a spreadsheet. You can import up to 100 test cases per test set. Questions can be up to 1,000 characters.

12. Change the test set name to:

```
Always Fail Copilot Studio Guide Set
```

13. Click **Save**.

14. Click **Evaluate** to run the evaluation. These adversarial test cases use the General Quality method to assess how the agent handles harmful requests.

Manually Create Test Cases from Test Canvas

15. Open the **Test Canvas** (the test chat panel on the right side of the screen).

16. Send the following message to your agent:

```
I want to get notified of the latest news about Copilot Studio.
```

17. The agent should trigger the mailing list topic and ask for your email address. Enter the email address of your lab account (e.g., `user@yourlabdomain.com`) when

prompted.

> ℹ️
> **Note:** For privacy reasons, use your lab account email rather than a personal email address. Your name is fine to use. All lab environment content is cleared 2 weeks after the event is over.

18. When prompted for your first name, enter your first name.
19. When prompted for your last name, enter your last name.
20. Now send the following questions to your agent one at a time, waiting for a response between each:

```
If I have a M365 Copilot license can I publish to Teams in Copilot Studio?
```

```
What are Tools in Copilot Studio?
```

```
What are all the topic triggers in Copilot Studio?
```

```
What kind of connected agents does Copilot Studio support?
```

21. Click **Evaluate**. This captures all the questions and agent responses from your test canvas session as a new test set, using the agent's actual responses as the expected responses.
22. Change the test set name to:

```
Always Pass Copilot Studio Guide Set
```

> ℹ️
> **Note:** Since the expected responses are captured directly from the agent's own answers, this test set should pass when evaluated - the agent should give the same (or very similar) answers when asked again.

23. Click on the first test case (the mailing list question). Change the evaluation method to **Text Match** and set the expected response to:

```
Please enter your email address to join the Copilot Studio announcements mailing list.
```

24. Delete the follow-up response entries for email, first name, and last name. These conversational follow-ups are not needed as individual test cases since the mailing list flow is already tested by the first question.

Add a Manual Test Case

25. Click **Add Case Manually**.

26. Enter the following question:

```
Where can I set DLP policies for Copilot Studio?
```

27. Click **Save Test Case**.

28. Click **Evaluate** to run the evaluation on the updated test set.

> ℹ️
> **Note:** Only one test set can run at a time. If an evaluation is already in progress from a previous step, you can wait for it to complete or move on to Use Case #3 and come back later.

⚠️

IMPORTANT: Key limits to remember: Each test set supports a maximum of 100 test cases. Questions can be up to 1,000 characters. Evaluation results are retained for 89 days.

🏅 Congratulations! You've completed Use Case 2!

Test your understanding

Key takeaways:

- **Four Creation Methods** – Auto-generation, CSV import, test canvas capture, and manual entry each serve different needs. Auto-generation provides quick baseline coverage, CSV import enables bulk management, test canvas captures real conversations, and manual entry targets specific scenarios.
- **Evaluation Methods Matter** – Different question types require different evaluation approaches. Factual questions need Exact/Keyword Match; open-ended questions benefit from General Quality or Similarity.
- **Test Set Strategy** – Creating test sets that are intentionally designed to pass or fail helps you understand how evaluations work before applying them to real quality measurement.

Lessons learned & troubleshooting tips:

- Start with auto-generated test cases for baseline coverage, then add manual cases for critical scenarios
- Use the test canvas approach to capture test cases from real conversations - the agent's own responses make reliable expected answers

- Review auto-generated test cases before relying on them - they may include irrelevant or poorly worded questions

Challenge: Apply this to your own use case

- What are the 10 most important questions your agent must answer correctly?
- Which creation method would you use for ongoing regression testing?
- How would you organize test sets for different agent capabilities?

🧱 Use Case #3: Review Evaluation Results

Analyze evaluation results across your test sets to understand pass/fail outcomes, examine detailed reasoning, and use insights to drive agent improvements.

Use case	Value added	Estimated effort
Review Evaluation Results	Interpret evaluation outcomes and use results to drive measurable agent improvements	10 minutes

Summary of tasks

In this section, you'll learn how to review evaluation results from the test sets you created in Use Case #2, examine pass rates and individual test case details, compare results across your "Always Pass" and "Always Fail" test sets, filter results, and use insights to identify improvement opportunities.

Scenario: You've created and evaluated three test sets in Use Case #2 - an auto-generated set, an intentionally failing set, and one captured from real agent conversations. Now you'll review the results to understand how evaluations work and how to use them for continuous quality improvement.

Objective

Review and interpret evaluation results, compare outcomes across test sets, and identify actionable improvements for your agent.

Step-by-step instructions

Review the Auto-Generated Test Set Results

1. Navigate to the **Evaluation** page in your Copilot Studio agent.
2. Select the **Non-Critical Copilot Studio Guide Set** to view its evaluation results.
3. Review the **pass rate** displayed in the results. This shows the percentage of auto-generated test cases that met their evaluation criteria (e.g., “7/10 passed - 70%”).
4. Click on an individual test case to view its detailed results:
    - **Question**: The original test question
    - **Expected response**: What the AI generated as the correct answer
    - **Actual response**: What the agent actually responded with
    - **Result**: Pass or Fail
    - **Reasoning**: An explanation of why the test passed or failed
5. For any failed test cases, review the **activity map** to see the step-by-step conversation flow showing the agent’s decision path, including which knowledge sources, tools, and topics were used.

> 💡
> **Tip:** The activity map is especially valuable for debugging failures. It shows exactly which knowledge sources, tools, and topics the agent used (or failed to use) when generating its response.

Review the “Always Fail” Test Set Results

6. Select the **Always Fail Copilot Studio Guide Set** to view its results.
7. Review the test case results. These adversarial questions test whether your agent properly refuses harmful or inappropriate requests using the **General Quality** evaluation method.
8. Click on a test case and review:
    - The **actual response** (how the agent handled the adversarial question)
    - The **reasoning** explaining why the evaluation determined it was a pass or failure
    - Whether the agent appropriately declined to answer or redirected the conversation

> ℹ️
> **Note:** The “Always Fail” test set uses adversarial questions to verify your agent’s safety guardrails. This is valuable for understanding how evaluations assess responsible AI behavior.

Review the “Always Pass” Test Set Results

9. Select the **Always Pass Copilot Studio Guide Set** to view its results.
10. Review the pass rate. Since expected responses were captured from the agent’s own answers in the test canvas, most test cases should pass.

11. Check the first test case (mailing list question) which you configured with **Text Match**. Verify that the agent's response matches the expected text:

```
Please enter your email address to join the Copilot Studio announcements mailing list.
```

12. Check the DLP policies test case you added manually. Review whether the agent was able to answer this question and what the evaluation result was.

> 💡
> **Tip:** If the DLP test case failed, it likely indicates a knowledge gap. This is exactly how evaluations help you discover areas where your agent needs additional knowledge or improved instructions.

Filter and Compare Results

13. Use the filter options to focus on specific subsets:
    - **All**: Show all test cases
    - **Pass**: Show only passing test cases
    - **Fail**: Show only failing test cases
14. Filter to **Fail** across your test sets to quickly identify exactly where your agent needs improvement.
15. If you have multiple evaluation runs for the same test set, use the **Compare with** dropdown to compare two runs side by side:
    - **Green arrows** show test cases that improved (failed before, pass now)
    - **Red arrows** show test cases that regressed (passed before, fail now)
    - **No change** indicators show consistent results

> 💡
> **Tip:** The comparison feature is one of the most powerful aspects of Agent Evaluation. It turns agent improvement from guesswork into a measurable, data-driven process.

Provide Feedback and Export Results

16. For individual test case results, use the **thumbs up/thumbs down** feedback options to indicate whether the evaluation's pass/fail determination was accurate:
    - **Thumbs up**: The evaluation correctly assessed the response
    - **Thumbs down**: The evaluation's assessment was wrong (false positive or false negative)
17. Click **Export test results** to download the results as a CSV file for stakeholder reporting or documentation.

> 💡
> **Tip:** Exported results are valuable for stakeholder reporting, compliance documentation, and tracking quality trends over time. Consider exporting results after each major agent update.

🏅 Congratulations! You've completed Use Case 3!

Test your understanding

- Why is creating an "Always Fail" test set a useful learning exercise?

- How does the activity map help you debug failed test cases?
- What does it tell you when a test case in the “Always Pass” set unexpectedly fails?

Challenge: Apply this to your own use case

- What pass rate would you set as a quality gate before deploying agent updates?
- How would you integrate evaluation runs into your agent development workflow?
- What stakeholders would benefit from seeing exported evaluation results?

🏆 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 analytics and evaluations in Copilot Studio:

- **Measure Continuously with Analytics** – Regular analytics review identifies issues early and guides improvement priorities. Use data, not assumptions, to optimize agent performance.
- **Focus on High-Impact Improvements** – Prioritize high-volume, low-satisfaction topics over rarely-used ones. Improve what affects the most users first.
- **Build Comprehensive Test Sets** – Use Quick Question Set for baseline coverage and manual entries for critical scenarios. Cover your agent’s most important capabilities.
- **Choose Evaluation Methods Wisely** – Match evaluation methods to question types. Factual questions need strict matching; open-ended questions need semantic evaluation.
- **Compare Runs to Verify Improvements** – Never assume a change helped. Run evaluations before and after changes and use the comparison feature to verify measurable improvement without regressions.

- **Close the Feedback Loop** – Analytics identifies problems, evaluations verify fixes, and continuous iteration ensures your agent gets better over time.

Conclusions and recommendations

Analytics and evaluations golden rules:

- Review analytics weekly for new agents, bi-weekly for mature agents
- Prioritize improvements based on conversation volume multiplied by satisfaction impact
- Track unrecognized phrases and unanswered questions - these reveal knowledge gaps
- Create evaluation test sets that cover your agent’s most critical capabilities
- Run evaluations before and after every significant agent change
- Make one improvement at a time to clearly attribute results
- Export and share evaluation results with stakeholders to demonstrate quality commitment
- Use the comparison feature to ensure improvements don’t cause regressions elsewhere

By following these principles, you’ll build a data-driven quality management practice that ensures your agents continuously improve - delivering measurable business outcomes through systematic monitoring, evaluation, and optimization.
