Evaluation Test Plan

AI Learning Advisor

10-Question Manual Evaluation Test Set

Generated: April 2026
Developer: Power CAT
Agent Type: Declarative Agent

Table of Contents

Table of Contents 2
1. Overview & Agent Summary 3
Agent Purpose 3
M365 Graph Capabilities 3
Conversation Starters 3
Special Modes (Skills) 3
Key AI Settings 4
2. Test Objectives 4
3. Test Cases 4
Test Case 1: Conversation Starter — Learning Plan 5
Test Case 2: Conversation Starter — Troubleshoot (Behavioral) 5
Test Case 3: Conversation Starter — What To Use When 5
Test Case 4: Conversation Starter — Explain a Concept (Beginner + Maker signals) 6
Test Case 5: Conversation Starter — What's New 6
Test Case 6: Conversation Starter — Compare (Maker default) 7
Test Case 7: Specificity & Precision — Pro Dev + Camp Labs 7
Test Case 8: Multi-Part — Concept + Build (Maker) 8
Test Case 9: Multi-Part — ALM + Governance Checklist 8
Test Case 10: Brainstorm — Marketing Beginner 9
4. Test Coverage & Method Reference 9
Category Distribution 9
Evaluation Approach Distribution 10
Coverage Notes 10
Manual Evaluation Criteria Reference 10
5. Tips for Running Manual Evaluations 11
Before Running 11
During and After 11
Iterating on the Test Set 11

1. Overview & Agent Summary

This evaluation test plan provides a focused 10-question test bank for AI Learning Advisor — a Microsoft 365 Copilot Declarative Agent that teaches, recommends, and troubleshoots across the Microsoft AI + low-code stack (M365 Copilot Agent Builder, Copilot Studio, and Power Platform). The plan is designed to validate the agent's core skills (audience adaptation, what-to-build-where, recommend & build, troubleshoot, learning plan, what's new, and explain-a-concept), its output contract (exact section headers, deep Microsoft Learn links, no fabricated content), and its scope guardrails.

> ⚡ Manual Evaluation: AI Learning Advisor is a Declarative Agent built through Agent Builder. Copilot Studio's automated evaluation tools are not available for declarative agents. All test cases must be evaluated manually by chatting with the agent directly in Microsoft 365 Copilot and reviewing responses against the criteria provided.

Field	Value
Agent Name	AI Learning Advisor
Agent Type	Declarative Agent
Developer	Power CAT
Test Set Size	10 questions
Test Set Type	Manual Single Response Evaluation
Date Generated	April 2026
Testing Channel	Microsoft 365 Copilot (Manual Chat)

Agent Purpose

Teaches and troubleshoots Microsoft's AI and low-code stack — M365 Copilot Agent Builder, Copilot Studio, and Power Platform (Power Apps, Power Automate, AI Builder, Dataverse). Explains concepts in plain language, builds personalized learning plans, recommends the right tool ("what should I build this in?"), gives step-by-step build guides, and troubleshoots errors and misbehaving agents/topics/flows. Grounded in Microsoft Learn and Copilot Developer Camp. Does not handle licensing, pricing, legal, or tenant-specific data.

M365 Graph Capabilities

Capability	Configuration
Web Search (allow-listed)	https://learn.microsoft.com/en-us/ • https://microsoft.github.io/copilot-camp/
Actions / Connectors	None (knowledge-grounded only)
Discourage Model Knowledge	Enabled (responses must ground in allow-listed sites)

Conversation Starters

- Create a Learning Plan — "Create a learning plan to take me from beginner to building my first Copilot Studio agent."
- Troubleshoot — "My Copilot Studio topic isn't triggering — help me troubleshoot it?"
- What To Use When — "What's the difference between Agent Builder and Copilot Studio and which should I use when?"
- Key Concepts — "Explain MCP (Model Context Protocol) like I am new to AI and why should I care as a maker?"
- What's New — "What's new in Copilot Studio and M365 Copilot and what should I try first?"
- Agentic AI Decoded — "Break down agentic AI vs. autonomous agents vs. multi-agent orchestration."

Special Modes (Skills)

- Skill 1 — Audience Adaptation: ask once if level is unstated AND not implied (Beginner / Maker / Pro Developer); default to Maker if skipped; recognize inline signals.
- Skill 2 — What to Build Where: recommend Agent Builder vs. Copilot Studio vs. Power Automate vs. Power Apps vs. AI Builder.
- Skill 3 — Recommend & Build: Recommended Tool → Why Not the Other → Build Steps (real UI names) → References → What's Next.
- Skill 4 — Troubleshoot: gather context first (no guessing); rank Top Fixes as Cause → Fix → Verify.
- Skill 5 — Learning Plan: 4-phase plan + Hands-on (Copilot Developer Camp) section + literal closing phrase.
- Skill 6 — What's New: ask for time window if missing; group items by Product.
- Skill 7 — Explain a Concept: Quick Answer + Why It Matters + (for comparisons) At a Glance table.

Key AI Settings

- Tone: friendly, encouraging, professional; lead with the answer; 150–350 words (expand for depth/troubleshooting).
- Output: Markdown with emoji-prefixed `##` headers; bullets/tables > paragraphs; bold key terms; backticks for products.
- Out of scope: legal, pricing, licensing, tenant-specific data, opinions on Microsoft strategy → use the literal redirect line.
- Link rules: every reply needs ≥1 deep learn.microsoft.com link in `## 🔗 References`; no bare roots, no /search/ URLs, no fabricated URLs/SKUs/limits/error codes.
- Closer: every reply ends with `## 📌 What's Next?` (1–2 follow-ups).

🎯 2. Test Objectives

Since AI Learning Advisor is a Declarative Agent, all testing is performed manually by chatting with the agent in Microsoft 365 Copilot. This evaluation aims to validate:

- Skill routing — the agent picks the right skill (Learning Plan, Troubleshoot, What To Use When, Explain a Concept, What's New, Recommend & Build) based on the user's intent.
- Audience adaptation — the agent honors inline level signals ("new to AI", "as a Maker", "pro code developer") and only invokes the Skill 1 level question when level is genuinely unstated AND not implied.
- Output contract — exact section headers, deep Microsoft Learn links, the literal Skill 5 closing phrase, and the mandatory `## 📌 What's Next?` closer.
- Groundedness — every reply contains ≥1 deep learn.microsoft.com link with a multi-segment path; no bare roots, no /search/ URLs, no fabricated content.
- Scope guardrails — out-of-scope questions (pricing, licensing, legal, tenant data, Microsoft strategy opinions) trigger the literal redirect line.
- Anti-invention — the agent does not fabricate release dates, error codes, SKUs, license tiers, limits, or URLs.
- Diagnostic restraint (Skill 4) — troubleshooting questions gather context BEFORE proposing fixes, rather than jumping to a guessed root cause.
- Multi-source reasoning — the agent can deep-link to both Microsoft Learn and Copilot Developer Camp on the same response.

📝 3. Test Cases

> ⚡ All test cases below must be evaluated manually. Open the AI Learning Advisor agent in Microsoft 365 Copilot, start a new conversation for each test, send the test question, and review the response against the pass criteria.

Test Case 1: 💬 Conversation Starter — Learning Plan

Field	Details
Category	Conversation Starter — Learning Plan
Test Question	Create a learning plan to take me from beginner to building my first Copilot Studio agent.
Expected Response	Recognizes 'beginner' as the implied level (no full Skill 1 prompt) and may ask ONE quick clarifying question (role / goal / focus area) before building the plan. Then delivers a 4-phase plan: Get Hands-On Today → Build Your First Real Thing → Level Up → Go Pro, each with deep-link Microsoft Learn modules, exercises, and self-checks. Includes a `## 🩹 Hands-on (Copilot Developer Camp)` section with deep page links (not the Camp root). Ends with the literal phrase "Your very next step right now: ___". Closes with `## 🔗 References` (≥1 deep Learn link) and `## 📌 What's Next?`.
Evaluation Approach	Overall Quality Assessment
Why This Approach	Open-ended structured plan — review holistically for the four phase names, output contract, link quality, and that the closing 'Your very next step right now: ___' phrase appears verbatim.
Pass Criteria	All four phase names appear in order; ≥1 deep learn.microsoft.com link with multi-segment path (no root or /search/ URLs); Camp section uses deep page links; literal closing phrase present; `## 🔗 References` and `## 📌 What's Next?` sections present.
What to Watch For	Asking a full Skill 1 level question (it shouldn't — 'beginner' is implied). Re-asking level later in the thread. Inventing module names. Bare Camp root URL `https://microsoft.github.io/copilot-camp/` (per agent's link rules this is a fail). Skipping the literal 'Your very next step right now: ___' phrase.

Test Case 2: 🛠️ Conversation Starter — Troubleshoot (Behavioral)

Field	Details
Category	Conversation Starter — Troubleshoot (Behavioral)
Test Question	My Copilot Studio topic isn't triggering — help me troubleshoot it?
Expected Response	Per Skill 4 behavioral path, asks for context BEFORE proposing fixes: (1) the exact user phrasing that failed, (2) expected vs. observed behavior (which topic, if any, triggered instead), (3) current configuration (trigger type — generative vs. 'User says a phrase', trigger phrases, knowledge sources, conditions), and (4) recent changes. Does not invent a root cause. Once gathered, would respond with `## 🛠️ Top Fixes` formatted as up to 3 ranked Cause → Fix → Verify items with a deep learn.microsoft.com troubleshooting link and one prevention tip.
Evaluation Approach	Overall Quality Assessment
Why This Approach	Validates that the agent follows the Skill 4 'gather context first' rule rather than jumping straight to fixes — a structured judgment.
Pass Criteria	Asks for at least 3 of the 4 context items (phrasing, expected vs observed, configuration, recent changes) BEFORE listing any fixes. Does not assert a root cause. If fixes are offered after context, they are formatted as ranked Cause → Fix → Verify with a deep Learn link.
What to Watch For	Skipping context-gathering and proposing 'Top Fixes' immediately is a fail per the agent's own contract. Watch for fabricated specific causes (e.g., 'your trigger phrase is misspelled') without asking. Generic non-Copilot-Studio guidance.

Test Case 3: ⚖️ Conversation Starter — What To Use When

Field	Details
Category	Conversation Starter — What To Use When
Test Question	What's the difference between Agent Builder and Copilot Studio and which should I use when?
Expected Response	Per Skill 2, delivers a direct comparison without asking for experience level. Includes a comparison/At-a-Glance table covering: declarative vs. multi-turn, where each runs (M365 Copilot Chat vs. multi-channel including Teams/web), actions/connectors, auth, and analytics. Recommends Agent Builder for speed when both fit, with Copilot Studio as the upgrade path. Includes 2–3 supporting reasons and at least one deep Microsoft Learn link in `## 🔗 References`. Closes with `## 📌 What's Next?`.
Evaluation Approach	Meaning Comparison
Why This Approach	The comparison can be phrased many ways, but the underlying meaning (when to use each, key differentiators) must match the Skill 2 framing.
Pass Criteria	Both products characterized accurately (Agent Builder = declarative, no-code, in M365 Copilot Chat; Copilot Studio = multi-turn, multi-channel, actions/connectors/auth/analytics). Recommendation present (Agent Builder first when both fit). Deep Learn link included. No experience-level question.
What to Watch For	Asking for experience level (it shouldn't). Conflating the two products. Mentioning licensing/pricing (out of scope). Bare root Learn URLs.

Test Case 4: 🤔 Conversation Starter — Explain a Concept (Beginner + Maker signals)

Field	Details
Category	Conversation Starter — Explain a Concept (Beginner + Maker signals)
Test Question	Explain MCP (Model Context Protocol) like I am new to AI and why should I care as a maker?
Expected Response	Recognizes inline level signals: 'new to AI' → Beginner tone (analogy, plain language, defined acronyms), 'as a maker' → Maker framing for relevance. Does NOT re-ask level. Per Skill 7, outputs `## 🎯 Quick Answer` (1–2 plain-language sentences with an analogy), `## 💡 Why It Matters to You` (2–3 bullets framed for a Maker), `## 🔗 References` with at least one deep learn.microsoft.com link, and `## 📌 What's Next?`.
Evaluation Approach	Overall Quality Assessment
Why This Approach	Tests inline level inference + Skill 7 output contract + groundedness — a holistic judgment is the best fit.
Pass Criteria	Beginner-friendly analogy in Quick Answer; 'as a Maker' framing reflected in Why It Matters; exact section headers `## 🎯 Quick Answer`, `## 💡 Why It Matters to You`, `## 🔗 References`, `## 📌 What's Next?`; ≥1 deep Learn link; no experience-level question.
What to Watch For	Re-asking experience level despite inline signals. Section headers renamed (e.g., 'Explain It Like I'm New to AI' instead of `## 🎯 Quick Answer`). Fabricated MCP claims with no Learn link. Heavy jargon for a beginner.

Test Case 5: 🆕 Conversation Starter — What's New

Field	Details
Category	Conversation Starter — What's New
Test Question	What's new in Copilot Studio and M365 Copilot and what should I try first?
Expected Response	Per Skill 6, may ask ONCE for a time window (e.g., last 30/90 days, current wave) if missing — NOT for experience level. Then returns 5–10 grouped 'What's New' items by product, each formatted as **Headline · 1-sentence summary · Why it matters · Reference link** (every link a

Field	Details
	deep learn.microsoft.com URL). Adds `## 🚀 Try First` with 1–2 highest-impact items defaulting to Maker level. Closes with `## 📌 What's Next?`.
Evaluation Approach	Overall Quality Assessment
Why This Approach	Open-ended; structure, grounding, and absence of fabrication matter more than exact wording.
Pass Criteria	If a clarifying question is asked, it is for time window (not level). Items grouped by product in the **Headline · summary · Why it matters · Reference link** format. Every Reference is a deep learn.microsoft.com URL. `## 🚀 Try First` section present. No invented release dates or features.
What to Watch For	Wrong clarifying question (level instead of time window). Fabricated specific release dates (e.g., asserting an exact 'Wave 2 2025' date) without a confirming Learn link. Bare root Learn URLs. Missing `## 🚀 Try First`.

Test Case 6: 🤖 Conversation Starter — Compare (Maker default)

Field	Details
Category	Conversation Starter — Compare (Maker default)
Test Question	Break down agentic AI vs. autonomous agents vs. multi-agent orchestration.
Expected Response	Defaults to Maker level (per Skill 7) and delivers immediately — does NOT ask for experience level. Includes `## 🎯 Quick Answer` (1–2 sentences calibrated to Maker), `## 💡 Why It Matters to You` (2–3 bullets), and `## 🏛️ At a Glance` table with one row each for **agentic AI**, **autonomous agents**, and **multi-agent orchestration** (Concept · What it is · When to use). Closes with `## 🔗 References` (deep Learn link) and `## 📌 What's Next?`.
Evaluation Approach	Overall Quality Assessment
Why This Approach	Concept comparison with strict output contract — best evaluated holistically against the Skill 7 + comparison rules.
Pass Criteria	All three concepts defined and clearly distinguished; comparison table present with the three required rows; exact section headers; ≥1 deep Learn link; no level question; closes with `## 📌 What's Next?`.
What to Watch For	Asking experience level (it should default to Maker). Conflating the three terms. Surface-level overview instead of a real breakdown. Missing the At-a-Glance table.

Test Case 7: 📏 Specificity & Precision — Pro Dev + Camp Labs

Field	Details
Category	Specificity & Precision — Pro Dev + Camp Labs
Test Question	I am a pro code developer, which labs should I complete in the Copilot Developer Camp and why?
Expected Response	Recognizes 'pro code developer' → Pro Dev level (skips Skill 1). Recommends a focused set of Copilot Developer Camp labs/pages relevant to a Pro Dev (e.g., Extend M365 Copilot pages, Copilot Studio extensibility, declarative + custom engine agents, API plugins/MCP). For each recommended lab: a deep page URL on `microsoft.github.io/copilot-camp/...` (NOT the Camp root) and a 1-sentence 'why' tied to Pro Dev concerns (APIs, ALM, SDKs, custom connectors). Includes `## 🔗 References` with at least one deep learn.microsoft.com link AND deep Camp page links. Closes with `## 📌 What's Next?`.
Evaluation Approach	Key Content Check
Why This Approach	The response must contain specific Copilot Developer Camp deep page links (not the root URL) and explicitly Pro Dev-relevant reasoning — easiest validated by checking for required elements.

Field	Details
Pass Criteria	≥3 specific Copilot Developer Camp lab/page recommendations, each with a 1-sentence 'why'. Every Camp URL is a deep page (multi-segment path on `microsoft.github.io/copilot-camp/...`), NOT the root. ≥1 deep learn.microsoft.com link. No experience-level question. `## 📌 What's Next?` present.
What to Watch For	Bare root URL `https://microsoft.github.io/copilot-camp/` is a fail per the agent's link rules. Asking for experience level despite the inline 'pro code developer' signal. Generic 'do all the labs' answers without specific page targeting. Vague non-Pro-Dev rationale.

Test Case 8: 🛠️ Multi-Part — Concept + Build (Maker)

Field	Details
Category	Multi-Part — Concept + Build (Maker)
Test Question	What is an autonomous agent? How do I create one if I am already a Maker?
Expected Response	Recognizes 'as a Maker' → Maker level (skips Skill 1). Two-part response combining Skill 7 + Skill 3. Skill 7 portion: `## 🎯 Quick Answer` defining an autonomous agent (proactive, trigger/event-driven, can take actions without per-turn human prompting), `## 💡 Why It Matters to You` (2–3 Maker-relevant bullets). Skill 3 portion: `## 🎯 Recommended Tool` (Copilot Studio with autonomous agent triggers), `## ⚖️ Why Not the Other` (2–3 bullets — Agent Builder lacks triggers/actions for autonomous behavior), `## 🛠️ Build Steps` (5–8 numbered steps using real Copilot Studio UI names — create agent, add trigger, configure actions/connectors, add knowledge, test, publish). `## 🔗 References` deep Learn link. `## 📌 What's Next?`.
Evaluation Approach	Overall Quality Assessment
Why This Approach	Compound question requiring two skills (concept explanation + build path) executed correctly with the right output contract — needs holistic evaluation.
Pass Criteria	Both halves answered: an accurate definition of autonomous agent AND Maker-level Copilot Studio build steps using real UI names. Exact section headers used. ≥1 deep Learn link. No experience-level question. Closes with `## 📌 What's Next?`.
What to Watch For	Answering only the 'what is' half and skipping the build path (or vice versa). Generic build steps that don't use real Copilot Studio UI names. Recommending Agent Builder (it doesn't support autonomous triggers). Inventing trigger names or features.

Test Case 9: ☑️ Multi-Part — ALM + Governance Checklist

Field	Details
Category	Multi-Part — ALM + Governance Checklist
Test Question	IT wants me to do some research on ALM and Governance for Power Platform before building an agent. Can you help me understand what ALM is and provide me with a governance checklist that I can show the IT team?
Expected Response	Skill 7 explainer for ALM tailored to Power Platform: `## 🎯 Quick Answer` (Application Lifecycle Management — managing solutions across dev/test/prod environments with source control, automated deploys, and governance), `## 💡 Why It Matters to You` (2–3 bullets). Then a `## ☑️ Checklist` for IT covering Power Platform-specific governance constructs: environment strategy (dev/test/prod), DLP (Data Loss Prevention) policies, Managed Environments, Center of Excellence (CoE) Starter Kit, solution-based ALM (managed vs. unmanaged solutions), Pipelines for Power Platform, Power Platform admin center monitoring, security roles & sharing, and audit logs. `## 🔗 References` with deep learn.microsoft.com links to Power Platform ALM and governance/admin docs. Closes with `## 📌 What's Next?`.
Evaluation Approach	Key Content Check

Field	Details
Why This Approach	The checklist must include specific Power Platform governance constructs (DLP, Managed Environments, CoE, Pipelines) — easiest validated by checking required items are present.
Pass Criteria	ALM defined accurately and tied to Power Platform. Checklist includes ≥6 of: environments, DLP, Managed Environments, CoE Starter Kit, solution-based ALM, Pipelines for Power Platform, admin center monitoring, security roles, audit logs. ≥1 deep Learn link to Power Platform ALM/admin docs. `## 📌 What's Next?` present.
What to Watch For	Generic 'governance checklist' that isn't Power Platform-specific. Missing DLP, Managed Environments, or CoE — these are core. Bare root learn.microsoft.com URLs. Mentioning licensing/pricing (out of scope).

Test Case 10: 💡 Brainstorm — Marketing Beginner

Field	Details
Category	Brainstorm — Marketing Beginner
Test Question	I am in Marketing and am new to building agents. Can you help me brainstorm some ideas for agents?
Expected Response	Recognizes 'new to building agents' → Beginner tone (plain language, no jargon, defined acronyms). Does NOT re-ask level. Provides 4–6 marketing-relevant agent ideas — for each: a 1-sentence purpose, a recommended tool (Agent Builder vs. Copilot Studio, justified per Skill 2), and a 'what it would learn from' note (knowledge source hint). Examples could include: brand-guidelines lookup, campaign brief Q&A, social copy review, competitor news digest, event FAQ, content-approval routing. Uses an emoji-prefixed `##` header for each idea or a clean numbered list. `## 🔗 References` with at least one deep learn.microsoft.com link. `## 📌 What's Next?` offering to deep-dive on one idea.
Evaluation Approach	Overall Quality Assessment
Why This Approach	Open-ended brainstorm — best judged holistically on audience-appropriate tone, marketing relevance, tool recommendations grounded in Skill 2, and output contract.
Pass Criteria	≥4 marketing-specific ideas (not generic productivity ideas). Each idea has a tool recommendation grounded in Skill 2 (Agent Builder for declarative Q&A, Copilot Studio for multi-turn/actions). Beginner-friendly language. ≥1 deep Learn link. `## 📌 What's Next?` invites a deeper dive.
What to Watch For	Generic, non-marketing ideas (e.g., 'an HR onboarding agent'). Heavy technical jargon for a beginner. Missing tool recommendations. Re-asking experience level. Bare root Learn URLs.

📊 4. Test Coverage & Method Reference

Category Distribution

Category	Count	Question #s
Conversation Starter — Learning Plan	1	1
Conversation Starter — Troubleshoot (Behavioral)	1	2
Conversation Starter — What To Use When	1	3
Conversation Starter — Explain a Concept	1	4
Conversation Starter — What's New	1	5
Conversation Starter — Compare (Maker default)	1	6
Specificity & Precision — Pro Dev + Camp Labs	1	7

Category	Count	Question #s
Multi-Part — Concept + Build (Maker)	1	8
Multi-Part — ALM + Governance Checklist	1	9
Brainstorm — Marketing Beginner	1	10

Evaluation Approach Distribution

Evaluation Approach	Count	Question #s
Overall Quality Assessment	7	1, 2, 4, 5, 6, 8, 10
Meaning Comparison	1	3
Key Content Check	2	7, 9

Coverage Notes

- All 6 conversation starters from declarativeAgent_0.json are tested verbatim as Q1–Q6, in order.
- Q7–Q10 broaden coverage with: Pro Dev / Copilot Developer Camp link discipline (Q7), multi-part concept + build (Q8), Power Platform ALM + governance checklist (Q9), and a Beginner-tone marketing brainstorm (Q10).
- Audience adaptation is exercised across all three levels: Beginner (Q1, Q4, Q10), Maker (Q6 default, Q8), and Pro Developer (Q7).
- Skill routing coverage: Skill 5 (Q1), Skill 4 (Q2), Skill 2 (Q3), Skill 7 (Q4, Q6, Q9), Skill 6 (Q5), Skill 7+3 multi-part (Q8), Skill 5/Camp (Q7), Skill 3/brainstorm (Q10).
- Anti-invention guardrails are stress-tested in Q5 (no fabricated release dates) and Q7/Q9 (deep page URLs only — no bare roots).
- Out-of-scope handling is not directly tested in this 10-question set. Refer to the 20-question extended Excel workbook for additional coverage including pricing, licensing, and opinion-based out-of-scope tests.
- All tests are performed manually in Microsoft 365 Copilot — Copilot Studio's automated evaluation tools are not available for declarative agents.

Manual Evaluation Criteria Reference

Evaluation Approach	What to Evaluate	How to Score	When to Use
Overall Quality Assessment	Holistic review of relevance, groundedness, completeness, structure, and abstention. The reviewer reads the full response and judges it against the expected response.	Score 0–100% mentally OR mark Pass if ≥70% of expected criteria are met. Use a structured scorecard.	Open-ended responses where multiple correct phrasings are valid (most learning, troubleshooting, what's new, and brainstorm questions).
Meaning Comparison	Check whether the agent's answer reflects the same meaning as the expected response, even if phrased differently.	Pass if the underlying meaning matches; Fail if the agent omits or distorts a key concept.	Comparison and definition questions where the structure is fixed but wording varies.
Data Source Verification	Manually confirm the agent referenced and grounded its answer in the expected knowledge sources (Microsoft Learn, Copilot Developer Camp).	Pass if links/citations point to the expected sources; Fail if the agent fabricates or omits required sources.	Tests where citation quality and source grounding are the primary success criteria.
Key Content Check	Manually verify the response contains specific required keywords, phrases, URLs, or section names.	Pass if all required content elements are present; Fail if any are missing.	Responses that must include specific section headers, link types (deep page URLs only), or named constructs (e.g.,

Evaluation Approach	What to Evaluate	How to Score	When to Use
			DLP, Managed Environments).
Response Similarity	Compare how closely the wording of the response matches the expected answer.	Pass if the wording is substantially similar; Fail if it diverges significantly in structure or terminology.	Responses with a strongly preferred phrasing (e.g., the literal Skill 5 closing phrase or the out-of-scope redirect line).
Exact Match	Confirm the response contains a specific exact string, character-for-character.	Pass if the exact string appears; Fail otherwise.	Short fixed strings — e.g., the literal phrase "Your very next step right now: ___" or the literal scope-redirect sentence.

5. Tips for Running Manual Evaluations

Before Running

- Open the AI Learning Advisor agent in Microsoft 365 Copilot (m365.cloud.microsoft) — confirm the agent appears in your Agents list and is the latest published version.
- Verify your account has access to the agent and to the allow-listed web search domains (learn.microsoft.com and microsoft.github.io/copilot-camp).
- Use a realistic test account — preferably one that mirrors the persona you are testing (Beginner, Maker, Pro Developer). Personalization signals can subtly affect responses.
- Prepare a scorecard (a simple notes file or the included 20-question .xlsx) with columns for Question, Expected, Actual, Pass/Fail, and Suggested Changes.
- Test during business hours when M365 Copilot service performance is most representative; avoid peak update windows.

During and After

- Start a new conversation for each test case — don't let prior context (especially a previously declared experience level) bleed across tests. The agent is designed to remember level within a thread.
- Copy the full response into your scorecard, including emoji-prefixed section headers, bullet structures, and every reference link.
- Click every reference link to confirm it (a) opens, (b) is a deep multi-segment URL on learn.microsoft.com or microsoft.github.io/copilot-camp, and (c) is relevant to the answer. Bare root URLs and /search/ URLs are fails per the agent's own contract.
- Score each test against the Pass Criteria and capture concrete Suggested Changes — call out the specific fix (e.g., "missing literal Skill 5 closing phrase" or "asked for level despite inline signal") rather than vague comments.
- Document edge cases and unexpected agent behaviors — useful inputs for prompt refinement.

Iterating on the Test Set

- Start with this 10-question set, then expand using the 20-question extended workbook for broader regression coverage.
- Track results over time — keep dated copies of the scorecard so you can spot regressions when the prompt or knowledge configuration changes.
- Re-test after every prompt change, conversation starter edit, or web-search allow-list change. Even small instruction edits can shift skill routing.
- Build new tests from real production usage — patterns of fails (e.g., wrong skill triggered, fabricated dates) are the highest-signal seeds for new test cases.

- When the agent's prompt is materially updated, refresh the Expected Response cells to reflect the new contract before scoring.

— End of Document —