MCP Server Accuracy Rubric

A five-dimension scoring method for measuring AI answer quality with and without an MCP connection.

What This Rubric Is

Scoring an AI answer as "good" or "bad" does not produce a number you can compare across runs. This rubric breaks quality into five independent dimensions, each scored 0 to 10, for a maximum of 50 points per question. Multiply by the number of questions to get your total scale, so five questions gives a 250 point maximum.

Run the same set of questions twice. Once with no MCP connection, using only what the model already knows, and once with your MCP server connected. Score both answer sets against the documented ground truth. The percentage difference is your accuracy lift.

How to Use It

Step 1 Choose 5 questions	Each must have one unambiguous answer explicitly stated in your documentation. No best-practice questions and nothing where "it depends" is a fair answer. Version-pinned facts and exact commands work best.
Step 2 Establish ground truth	Before running either condition, write down the correct answer for each question from the current documentation, and record the date.
Step 3 Run Condition A	Ask all five questions in a fresh chat session with no MCP connection and no pasted context. Save the full output.
Step 4 Run Condition B	Ask the same questions in a fresh session with your MCP server connected, and prompt the model to use it.
Step 5 Score both outputs	Score each answer on all five dimensions using the tables below. Score against ground truth only, not whether the answer sounds right.
Step 6 Calculate the lift	Divide the improvement in total score by the Condition A total to get the relative accuracy improvement.

The Five Dimensions

Each dimension is scored independently from 0 to 10. A score of 10 requires that the criterion is fully met according to the current documentation, not that the answer reads as reasonable.

1. Factual Accuracy

Is every verifiable claim correct per the current documentation?

Score	What it means
10	Every fact matches the current documentation exactly
7–9	One minor detail is slightly off or paraphrased, but everything substantive is correct
4–6	Core answer is right, but at least one specific fact such as a version number, flag or variable name is wrong
1–3	Directionally right but the specific values are wrong
0	Confidently and substantively incorrect

2. Specificity

Are exact values given, such as commands, variable names and version numbers, rather than vague guidance?

Score	What it means
10	Concrete, usable values given for every part of the answer
7–9	Most values given; one part is hedged or paraphrased
4–6	Right topic covered but without the specific detail needed to act on it
1–3	Vague throughout
0	No useful specifics at all

3. Completeness

Does it cover all parts of the question, including the caveats needed to act safely?

Score	What it means
10	All sub-questions answered, and all important prerequisites and safety caveats mentioned
7–9	One minor sub-point or caveat missing
4–6	Main answer is there but a meaningful piece is missing
1–3	Significant gaps, so a reader would need to look elsewhere to fill them in
0	Answer is essentially incomplete

4. Currency

Does the answer reflect the current release, not a superseded one?

Score	What it means
10	All version-specific details match the current release
7–9	Mostly current with one minor dated reference
4–6	Some details reflect an older release
1–3	Answer primarily describes outdated behaviour
0	Information is from an old version and substantively wrong today

5. Source Groundability

Can every claim be traced to a specific statement in the documentation, rather than something that merely sounds plausible?

Score	What it means
10	Every claim traceable to a specific page and section
7–9	Most claims traceable; one slightly paraphrased but clearly originating from the documentation
4–6	Core facts are real but at least one specific claim has no origin in the documentation
1–3	Multiple claims exist nowhere in the documentation
0	Answer contains invented specifics throughout

Score Sheet

Fill in one row per question, then repeat the sheet for the second condition.

Server name: _____ Date: _____

Condition: [ ] No MCP [ ] With MCP

Question	Factual	Specific	Complete	Currency	Ground.	Total
Q1						
Q2						
Q3						
Q4						
Q5						
Total						

Maximum: 50 points per question. Five questions gives a 250 point maximum.

Calculating the Improvement

Relative accuracy improvement

(With MCP total - No MCP total) ÷ No MCP total × 100

Worked example: (245 - 139) ÷ 139 × 100 = 76.3% improvement

Tips for Good Test Questions

- One unambiguous answer per question, explicitly stated in the current documentation.
- Prefer version-pinned facts such as minimum versions and deprecated flags. These are where grounding matters most.
- Include exact commands and syntax-sensitive questions, since models frequently invent flags that do not exist.
- Avoid questions where "it depends" is a legitimate answer, because those are hard to score fairly.
- Establish ground truth before running either condition. Looking the answer up afterwards biases scoring.
- Confirm the documentation topic is actually indexed in your MCP server before testing it.

Put This Rubric to Work

The most useful version of this test is the one you run against documentation you own. Building the MCP server side of it takes about two minutes: paste a documentation URL, a GitHub repository or any website, and MCP Studio handles indexing and serving. The free tier covers everything this rubric requires.

Build your MCP server free »
appatools.com/mcp-studio/studio

Read the full test and results »
The MCP Server Rubric, scored

Browse the documentation »
docs.appatools.com/mcp-studio

Once your server is live, the MCP Studio analytics dashboard shows which pages were retrieved for every answer, so each score on this sheet can be traced back to a real source.