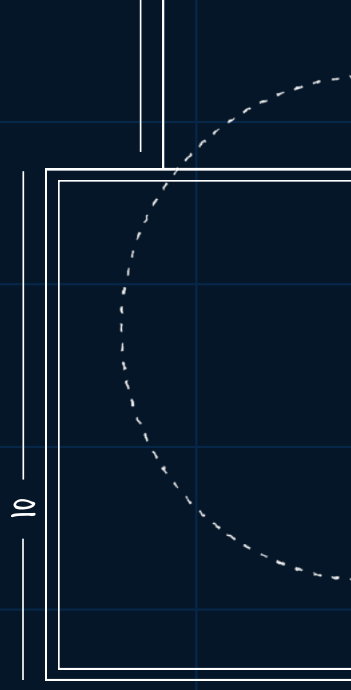
workato

HOW WE MCP | PART 3 OF 4

Building for LLMs



You can have the right scope, the right principles — and still ship tools that break. Here's what actually fails, and how to fix it.

LLMS SEE tool names	LLMS SEE descriptions	LLMS SEE parameters	LLMS SEE outputs

Hard to choose

Overlapping tool names force the model to guess between equally valid options.

BAD
```
search_customers
find_customers
```

GOOD
```
search_customers
get_customer (ID)
```

Make the right choice obvious.

Parameters don't map

Generic inputs force interpretation before action. Each parameter should match how users express intent.

BAD
```
search
(query, filters, opts)
```

GOOD
```
search
(name, date, status)
```

Reflect how people ask.

Too much data

Unbounded results bury signals and waste context. More data doesn't improve reasoning — it makes it harder.

BAD
```
returns 500 rows
no truncation
```

GOOD
```
bounded results
has_more: true
```

Return bounded, usable data.

Outputs lack meaning

Different outcomes require different decisions. Status alone isn't enough — the model needs to know why.

BAD
```
status: success
data: []
```

GOOD
```
nothing_found
permission_denied
```

Communicate meaning, not status.

Outputs aren't usable

Raw IDs and loose structure force reinterpretation before the next step. That's where errors enter.

BAD
```
{ id: "a3f9x" }
```

GOOD
```
structured fields
reusable in next call
```

Enable the next step without guessing.

BEFORE SHIPPING ANY TOOL, ASK

- [ ] Does the name make the right choice obvious?
- [ ] Do parameters map to how users express intent?
- [ ] Are results bounded with clear truncation signals?
- [ ] Do outputs distinguish meaningfully different states?
- [ ] Can the model use the output directly in the next step?

THE THROUGH-LINE

MCP tools aren't just functions. They are interfaces for reasoning.

If the model has to guess at any step — selection, input, output, or interpretation — reliability breaks. Design for the model doing the reasoning, not the developer building the system.