WHITE PAPER · ERP DELIVERY · NOVEMBER 2026

AI-run discovery for ERP implementations

How an agent over MCP turns 6–8 days of fit-gap workshops into a structured, reviewable backlog

192
requirement cards from one scripted order-to-cash discovery

0
"fit" verdicts without a cited rule

4
review gates before a consultant signs the document

Based on a demo build and test run for a Microsoft Dynamics partner (1,400+ staff, Central Europe), September–October 2026. Fictional client data; not a production rollout.

Gergely Racz
PhD Engineering (Cambridge) · AI architect

Lead Flow Automation
Budapest · leadflowautomation.net

Executive summary

Every Business Central (BC, Microsoft's ERP for small and mid-sized companies) rollout starts with discovery: consultants sit with the client's staff, walk through order handling, invoicing and finance, and sort what they hear into **fit** (standard BC covers it after configuration) and **gap** (custom work is needed). For the Microsoft Dynamics partner we built this demo for (1,400+ staff, Central Europe), discovery for one end-to-end process takes **6–8 days of in-person consulting**.

In September 2026 we built a working demo of a different shape: an AI agent that interviews client staff by role, keeps every answer verbatim, turns answers into requirement cards, classifies each card against BC knowledge with a cited rule, and writes the partner's own deliverables — their Fit-Gap Excel template, gap specifications and a reviewed discovery document. The same engine is exposed over **MCP** (Model Context Protocol, an open standard that lets AI assistants call external tools), so a consultant can query a discovery from Claude Desktop or another MCP-capable assistant.

This is a **demo build and a scripted test run on fictional data, not a paid production rollout**. On one end-to-end process (order-to-cash) with three scripted roles and a technical intake, the run produced 192 business requirement cards, classified all of them and wrote 66 gap specifications in 7.4 minutes, with **zero "fit" verdicts without a cited rule**, filled the partner's Excel template without altering a single untouched cell, and produced a 27-page document that passes through four review gates before a consultant approves it. The same run also showed the limits: role coverage of 35–60 %, a 27 % "unknown" share, one process family out of fifteen, and no real client participant yet. Both are reported below.

The point is not fewer consultants. It is one consultant reviewing structured evidence instead of transcribing workshops.

1. The problem: discovery is expensive, slow and uneven

Discovery is where ERP (enterprise resource planning) projects are won or lost, and it is the most manual part of delivery.

- **Consultant time.** The partner described a typical discovery as 6–8 consultant days on site for end-to-end processes such as order-to-cash. That time is spent asking known questions and writing up the answers, not designing solutions.
- **Inconsistency.** The partner was moving its discovery method onto the Microsoft Business Process Catalog, a public taxonomy of end-to-end processes (Microsoft Learn). The catalog file we received holds 8,010 work items across 15 end-to-end flows, but only 11 workshops carry an agenda and key questions. Each consultant fills the rest from experience.

- **The catalog cannot decide fit or gap for BC.** Only 23 of 3,763 catalog scenarios name Business Central, and none of those has a description. Every configuration deliverable in the file targets other Dynamics 365 products. The real fit-gap knowledge lives in the partner's own Fit-Gap Excel: 139 BC functions, a gap register and a list of the partner's add-on products.
- **Hand-off loss.** After discovery, consultants write business user stories, lead developers write technical specifications, and developers build from stories in Azure DevOps. Every hand-off retypes information that was already said once in a workshop.

There was also a human constraint. The partner's staff were uneasy about AI and jobs. The brief was therefore explicit: a force multiplier for consultants, with the consultant in control of every verdict.

2. The approach: interview, capture, classify, specify

The agent runs the questioning; the consultant runs the judgement. The pipeline has five steps.

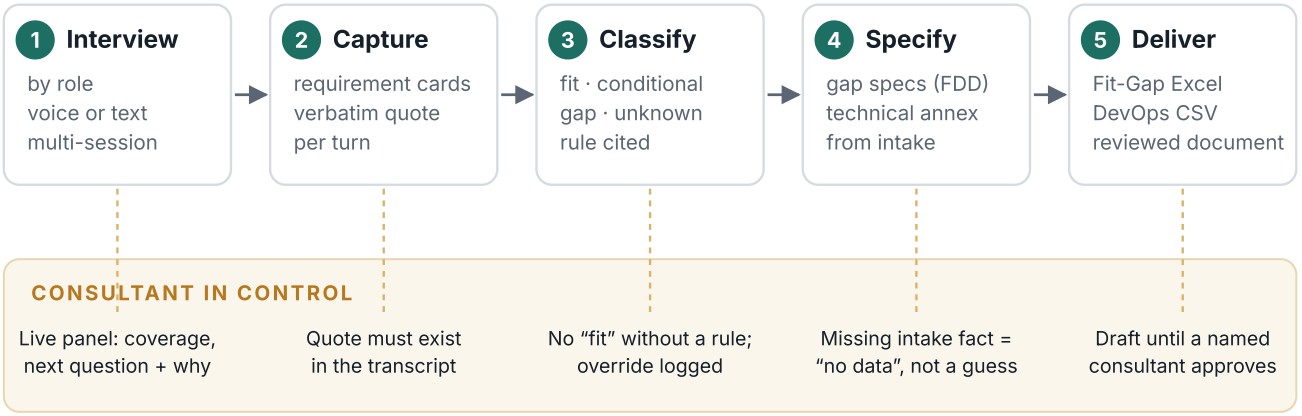


Figure 1. The discovery pipeline. The agent does the asking and the writing; every step ends at a check a consultant can see and overrule.

1. Role-based interview. A participant opens an invitation link, names their role (order desk, accounts receivable clerk, finance lead) and talks to the agent by voice or text, in one sitting or across several sessions. The agent opens with the role and a typical working day, then walks the process areas in catalog order, choosing the next question deterministically: mandatory questions first, then untouched processes for that role. The question bank for order-to-cash holds 259 hand-written questions, translated to Hungarian for the participants; each states which roles it applies to and what a complete answer contains.

2. Structured process capture. Every answer is stored verbatim. In the background a model extracts zero or more **requirement cards**: catalog process ID, a one-sentence statement, the verbatim evidence quote with a link to the turn, volumes, systems, and a first fit/gap guess. Each evidence quote is checked against the transcript word for word; in every measured run, no card failed that check. A live consultant panel shows coverage across the 79 order-to-cash processes, the cards as they arrive, the next question with the reason it was chosen, and "missing facts" the agent has noticed.

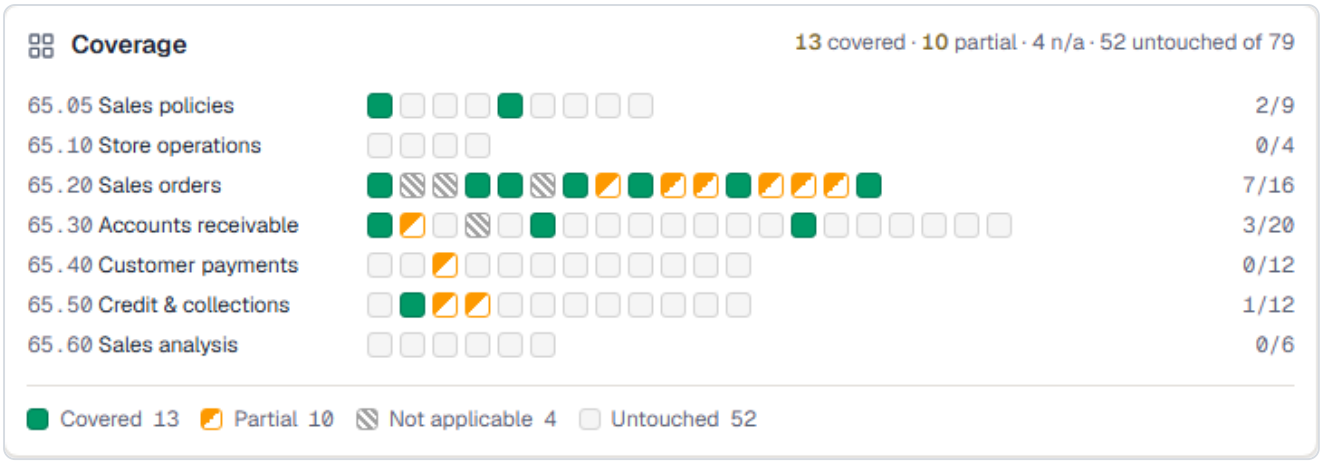


Figure 3. Live coverage panel from the test run (order-desk role, end of interview): the 79 order-to-cash processes in 7 catalog areas, each marked covered, partial, not applicable or untouched.

3. Fit/gap against Business Central. Each card is classified as fit, conditional, gap or unknown. The classifier retrieves the rules for the card's process — 68 rules derived from the partner's Fit-Gap Excel (41 BC functions mapped to order-to-cash processes, every mapping reviewed by hand), and 116 rules derived from Microsoft's public BC documentation (Microsoft Learn), whose 68 fit rules carry a quote verified against the source page. One rule is enforced in code, not in the prompt: **"fit" is only allowed with a cited rule; otherwise the verdict becomes "unknown" and goes to the consultant.** The agent proposes; the rulebook permits.

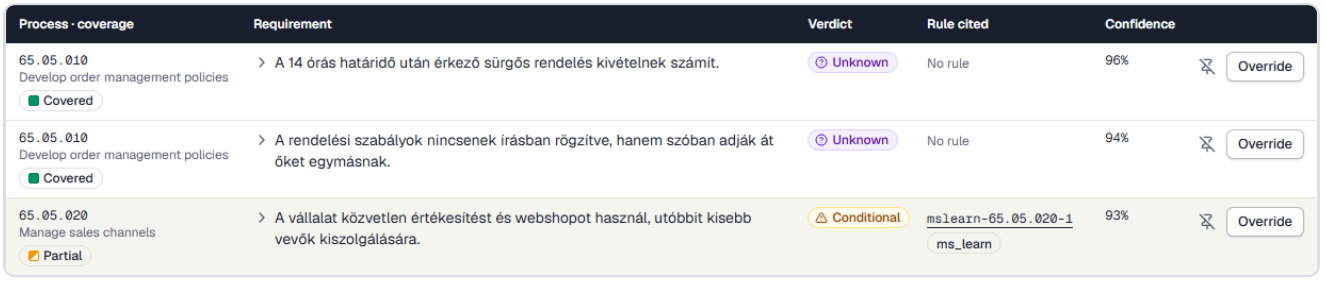


Figure 4. Classified cards in the consultant dashboard (interview language Hungarian). Top two: an order-handling exception and "order rules are passed on verbally, not written down" — governance facts with no matching BC rule, so the verdict is *Unknown*. Bottom: a web-shop channel marked *Conditional* with the cited Microsoft documentation rule. Every row has an Override control.

4. Gap specifications. Each gap and conditional card gets a short functional specification in the shape of an FDD (functional design document): context, current way of working with evidence, required behaviour, why standard BC does not cover it (rule cited), acceptance criteria and open questions. A separate **technical intake** — 40 questions, 20 mandatory, answered by the partner and the client's IT rather than by the order clerk — collects platform, localisation, installed extensions, integrations, data volumes and delivery facts. A technical annex is generated only from those answers; where an answer is missing it says so instead of inventing BC object names.

5. The partner's own deliverables. Outputs are written in the formats consultants already use: their Fit-Gap Excel template (client-specific process text appended to matched functions, new rows in the gap register, add-on product suggestions marked as suggestions), an Azure DevOps CSV of user stories, a full verbatim transcript, and a business and functional discovery document.

3. Architecture

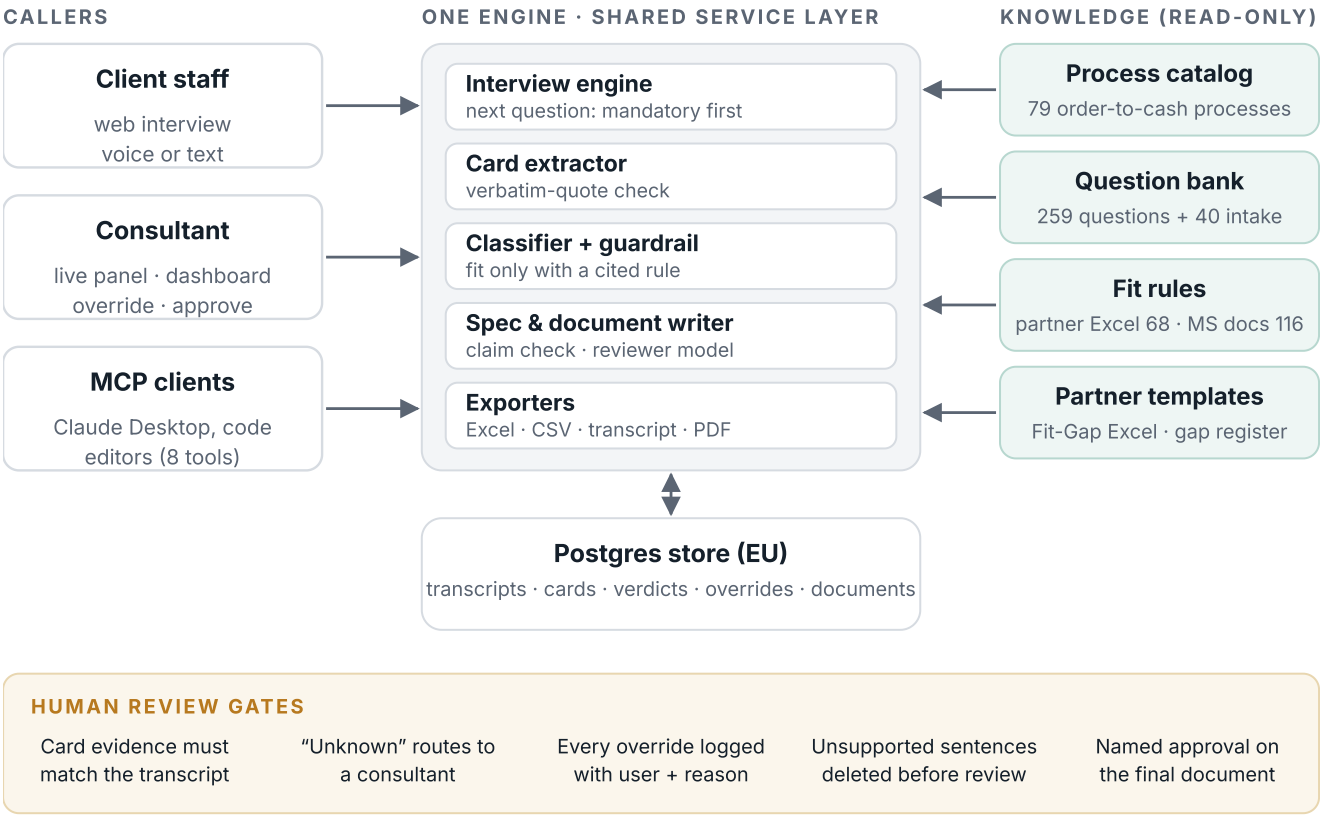


Figure 2. Architecture of the demo build. The web interview, the consultant dashboard and MCP clients all call the same service functions, so the guardrails apply to every caller.

The design choices that matter:

- **One engine, several front doors.** The web interview, the consultant dashboard and the MCP server call the same service functions. An MCP tool is another caller, never a second way in, so there is one place where rules are enforced.
- **MCP as the integration surface.** Eight tools: `list_projects`, `get_coverage`, `list_requirements`, `get_fit_gap_report`, `get_gap_spec`, `export_stories`, `start_interview`, `answer`. The partner's developers already pull Azure DevOps work items into their editor over MCP, so a discovery becomes one more source their assistants can read. In the test, a Claude-Desktop-style client connected through `mcp-remote`, listed all eight tools and fetched the fit-gap report in under a second.
- **No agent framework.** The pipeline is a small, fixed state machine — interview loop, extract, classify, specify, export — written as typed steps with validated data contracts. Each step is testable without a model and each model is replaceable. The production code base had 225 passing automated tests at the end of the build.
- **Model per task, chosen by measurement.** An 11-model bake-off picked a fast, low-cost model for interview turns and classification, Claude Opus for gap specifications, Claude Sonnet for the document draft, and a model from a different family as the document reviewer.

- **Human review gates.** Card overrides (logged with who and why), a draft-only document with a watermark until a consultant approves it, and "unknown" as a first-class verdict that routes to a person.

4. What the test run showed

We ran one complete discovery on a fictional Hungarian trading company: three business roles on order-to-cash (order desk, accounts receivable, finance lead) plus the technical intake. Participants were **scripted**: a model picked each answer line from a fixed answer bank of roughly 30 lines per role and never wrote text itself, and when the bank had no line the answer was "I don't know". The run used the production database and production models.

Measure	Result
Answers per role (of which "don't know")	50 (20) · 92 (53) · 90 (56); intake 40
Wall-clock time per role	4.9 · 9.2 · 10.5 min; intake 3.0 min
Requirement cards	192 business cards + 96 technical intake facts; 18 near-duplicates merged for output
Coverage of 79 order-to-cash processes	26 covered · 33 partial · 11 not applicable · 9 untouched
Role-scoped processes resolved	60 % · 41 % · 35 %; intake 100 %
Partner Fit-Gap functions touched	33 of the 41 relevant to order-to-cash
Verdicts	75 fit · 60 conditional · 6 gap · 51 unknown (27 %)
"Fit" without a cited rule	0 (one guardrail downgrade)
Classification time	441 s for 192 cards and 66 gap specs, 22 batches, 0 failed calls
Fit-Gap Excel	19 function rows extended, 6 gap rows, 3 product suggestions; 0 differences in untouched cells, data validations preserved
Discovery document	27 pages, cost about $1.30 in model usage per document, generated once per classification run

What worked. Every card carried a verbatim quote that exists in the transcript. The guardrail held: no "fit" without a rule. The Excel came back as the partner's own file with their structure intact. The 51 unknowns were not failures to retrieve: every one had rules for its process, but the card described governance or people ("who sets the order rules", "how a new colleague is trained", a 14:00 order cut-off) rather than a BC capability. Routing those to a consultant is the correct outcome.

The document passes four gates before anyone signs it. A code-level claim check drops any sentence whose numbers, systems or object names are not in the cited cards (3 sentences

dropped). A linter fixed 10 of 11 terminology and format findings. A reviewer model from a different model family raised 29 findings; the revision step rewrote 21 passages, deleted 3 repetitions and moved 3 disputed points to the open-questions section for the consultant. A render check confirmed 0 layout defects on 27 pages. The approved PDF records who approved it and when.

Speed. On a benchmark of 20 real turns, the voice path started its reply at a median of 0.98 s (95th percentile 1.3 s); the requirement cards follow about a second later. Through the web interface, text-mode replies took 2.5–3.0 s at the median.

Limits, from our own devil's-advocate review

We reviewed the demo against the partner's original ask before showing it. The honest gaps:

1. **No real participant yet.** All complete runs used scripted answers. Multi-day, multi-role use by real client staff was tested in code, not in a real discovery. As of the review, no person had completed a full voice interview on the production system.
2. **One process family of fifteen.** Only order-to-cash has a question bank and rules. A real discovery also covers purchasing, inventory and finance; the partner's template spans 139 functions across 8 areas.
3. **Resolution, not just completion.** "Interview complete" means the role's question set is exhausted. With scripted banks, 35–60 % of a role's processes were resolved; a real participant answering more questions would raise this, the engine cannot.
4. **The question bank has not been reviewed by a consultant.** It was written from the catalog's workshop pattern and public documentation. A consultant with years of method will find holes.
5. **Functional, not technical.** Without environment facts, a gap specification cannot name BC objects, events or extension points. That is why the technical intake and its cited annex were added; the main document was renamed "business and functional" to match what it is.
6. **Partial Excel fill.** Only functions touched by cards are filled. A consultant marks all relevant rows after discovery, and the agent's judgement of what the client *needs* versus what was merely mentioned is still a guess.
7. **Operational edges.** One extraction call timed out and stopped a session at answer 91 (the answer can be resent). Classification had to be rebuilt as a batched background job before a realistic run of 192 cards could finish inside hosting time limits.

5. Risks and governance

The consultant stays in control. The agent never finalises a verdict on its own authority. "Fit" requires a cited rule; "unknown" is routed to a consultant; every override is logged with user and reason; the document stays a watermarked draft until a named consultant approves it. Every

statement in the outputs traces back to a card, and every card to a verbatim quote and a transcript turn.

Data handling. The demo used fictional data only, stored in an EU-hosted Postgres database, with the web app served from Frankfurt. Model calls ran under our own accounts. A real deployment belongs in the partner's own cloud tenant under a data processing agreement, with transcript retention set by contract; the code was written so the model provider and hosting can be swapped.

Recording and workplace rules. Interviewing employees has legal edges that must be cleared before a pilot, not after. We flagged three: consent before recording spoken words (in Germany, §201 of the Criminal Code); co-determination rights of works councils over tools that could monitor employee performance (German Works Constitution Act, §87(1) no. 6); and the EU AI Act's prohibition of emotion recognition in the workplace (Article 5(1)(f)). The demo therefore captures process facts only and does no sentiment or performance scoring of staff.

Model risk. Models make plausible mistakes. The controls are structural rather than prompt-based: verbatim-quote validation, the rule-citation guardrail in code, a claim check that deletes unsupported sentences, and a reviewer from a different model family.

6. A 30-day pilot outline

A pilot should answer one question: does agent-run discovery save consultant time on a real client without lowering the quality of the fit-gap result?

Week	Work	Exit check
1	Consultant review of the order-to-cash question bank and rule set; import the partner's current Fit-Gap template; deploy in the partner's tenant; consent and works-council text agreed	Consultant signs off the bank; legal check done
2	Interviews with 3–5 real client staff in their roles, in sessions spread over the week; technical intake answered by the partner and client IT	Each role reaches its mandatory question set
3	Classification, gap specs, Excel and document; a consultant reviews every unknown and conditional verdict and records overrides	Override rate and reasons logged
4	Side-by-side comparison with the consultant's own fit-gap for the same scope; consultant hours counted on both paths; go / reshape / stop decision	Written comparison and decision

Scope stays at one end-to-end process. Success thresholds — hours saved, share of verdicts the consultant accepts unchanged, gaps the agent missed — are agreed with the partner in week 1, before any data exists.

7. About the author

Gergely Racz is an AI architect and developer based in Budapest. He holds a PhD in Engineering from Cambridge, has more than ten years of AI experience, and is named on 7 patents. He started in machine learning and now builds AI agent systems for European B2B companies: retrieval-augmented knowledge assistants in production daily use at a $3B/yr multinational, agent workflows for a European industrial trading company, and MCP servers that put company knowledge inside the assistants teams already use.

Lead Flow Automation · leadflowautomation.net · YouTube: AI Lead Flow Automations

---

Sources: the demo specification, implementation plan, build and test reports, and internal review (September–October 2026); the client's name and data are withheld. Public references: Microsoft Business Process Catalog (learn.microsoft.com/dynamics365/guidance/business-processes), Business Central documentation (learn.microsoft.com/dynamics365/business-central), Model Context Protocol (modelcontextprotocol.io), Regulation (EU) 2024/1689 (AI Act), Art. 5(1)(f).