BA TOOLKIT · FREE TEMPLATE

Prebuilt AI agent fit-gap matrix

Six gates, a first-match decision table and a filled Northstar example for deciding whether to adopt an off-the-shelf AI agent

By **Prachit Parikh**, business analyst · github.com/prachitparikh/ba-toolkit

A decision table and scoring matrix for one question: **should we adopt this prebuilt AI agent for this process, and what has to change first?**

Grain: one row = one step of your process, as your team runs it today.

Why this exists

Vendors now sell AI agents off the shelf by the thousand (IBM and Microsoft announced a catalog of "more than 6,000 AI agents and assets" on 10 November 2026, see IBM Newsroom). A prebuilt agent is a packaged product with a process baked in, so the old BA question for packaged software comes back: where does it fit our process, where does it need configuring, and where is there a gap? The difference is that an agent acts. A gap in an ERP screen means a workaround. A gap in an agent that issues refunds means money out the door. This matrix runs the classic fit-gap analysis with the agent-specific checks added, and it makes the decision before the contract, not after go-live.

How to use it

1. **List your process steps** as the team runs them today, one per row. Write the step, not the system ("decide whether the return is inside the window", not "returns module").
2. **Check the six gates first.** Gates are pass/fail, not scored. If any gate fails, stop: the answer is "don't adopt yet", whatever the scores say.
3. **For each step, write what the agent does out of the box** in one sentence, from a run on your own cases, not from the demo.
4. **Classify each step with the decision table** below. First match wins.
5. **Fill in effort, undo and owner** for every row that isn't a plain Fit.
6. **Apply the overall decision rule** at the bottom.
7. **Put the matrix in the requirements pack.** Every Configure and Extend row becomes a requirement with acceptance criteria. Every "Change our process" row needs a sign-off from the process owner.

The six gates (all must pass)

#	Gate	Passes when	Evidence to ask for
G1	One named owner	A person on your side (not "IT", not the vendor) owns the agent's behavior and can stop it	Name and role in the matrix header
G2	Never-rules enforced outside the prompt	Every "the agent must never..." rule is enforced by permissions, limits or validation the agent can't talk its way past	The vendor shows where each never-rule lives (tool permission, API limit, schema check)
G3	Off switch with names	Who can stop it, where work in flight goes, who signs it back on	A tested stop in the trial, not a slide
G4	End state you can check	Every action leaves a record in your system you can query (not only the agent's own transcript)	An export or a query of the actions taken in the trial
G5	Your cases, before the contract	You can run your own golden set against the agent during evaluation	Trial access with your data, or a sandbox with realistic data
G6	Data stays where policy says	Customer data, logs and prompts stay in the locations and retention periods your policy allows	Written answer from the vendor, checked by your data protection owner

Fit levels

Level	Meaning	What happens next
Fit	Does the step as you run it today, proven on your cases	Test it in UAT like any other step
Configure	Does it after settings, rules or prompts you control, no code	Write the configuration as a requirement with acceptance criteria
Extend	Needs a custom tool, integration or code	Estimate it; it's a build, so it gets a design and tests
Change our process	The agent's way is fine and yours is the odd one out	The process owner signs off the change and the training it needs
Gap	Can't do it, or can't do it safely	Keep a person on the step, or don't adopt

Decision table for each step (first match wins)

#	If...	Classify as
1	The step can't be undone, the agent would do it alone, and you have no evidence from your own cases	Gap
2	The agent would break a never-rule on this step, or the rule isn't enforced outside the prompt	Gap

#	If...	Classify as
3	It does the step correctly on your golden cases with no changes	Fit
4	It does the step correctly after settings, rules or prompt changes you control	Configure
5	It needs a new tool, an integration or code to do the step	Extend
6	Your step differs, the agent's way meets the business rule, and changing costs less than extending	Change our process
7	None of the above	Gap

Rule 1 sits at the top on purpose. A demo on the vendor's data isn't evidence for a step you can't take back.

The matrix (copy this)

Also available as a CSV: `prebuilt-ai-agent-fit-gap-matrix.csv`.

Step ID	Process step (as we run it today)	What the agent does out of the box (from our cases)	Evidence	Undo possible?	Fit level	Effort (S/M/L)	Requirement or decision	Owner
			Our cases / vendor demo / none	Yes / No / Partly				

Overall decision rule

Count the rows by level, then take the first line that matches:

1. Any gate failed → **Don't adopt yet.** Say which gate and what would fix it.
2. Any Gap on a step the agent would do alone → **Don't adopt yet**, or redesign so a person does that step.
3. More than a third of steps are Extend or Gap → **Pilot behind a person**: the agent recommends, a person acts, and you log where they disagree.
4. Every Extend row is estimated and every Change-our-process row is signed → **Adopt with changes.**
5. Otherwise → **Adopt**, and test it in UAT with pass marks set per step before the first run.

Filled example (fictional)

Client: Northstar Systems (fictional), online retail. **Agent under evaluation:** a prebuilt "returns and refunds agent" from a vendor catalog (fictional vendor, called Vendor R here). **Owner (G1):** Head of customer

operations. **Golden set:** 120 past return requests, anonymized, with the decision the refunds lead says was right.

Gates

Gate	Result	Note
G1 One named owner	Pass	Head of customer operations
G2 Never-rules outside the prompt	Fail	"Never refund above $200 without approval" is written in the agent's instructions only. Vendor R says an API limit is "on the roadmap"
G3 Off switch with names	Pass	Tested in the trial: ops lead paused it, 4 cases in flight went back to the human queue
G4 End state you can check	Pass	Every refund the agent issues appears in Northstar's refunds table with an agent ID
G5 Your cases before the contract	Pass	120-case golden set run in the vendor sandbox
G6 Data stays where policy says	Pass	Vendor's written answer checked by the data protection owner

Steps

Step ID	Process step	What the agent did on our cases	Evidence	Undo?	Fit level	Effort	Requirement or decision	Owner
S1	Read the customer's return request and find the order	Found the right order in 118 of 120 cases; 2 misses were orders placed as a guest	Our cases	Yes	Configure	S	Match guest orders by email + order number (AC-RET-01)	Ops lead
S2	Check the return is inside the 30-day window	Used ship date; Northstar's policy counts from delivery date	Our cases	Yes	Configure	S	Window counts from delivery date (AC-RET-02); test the 30th and 31st day	Refunds lead
S3	Check the item is returnable (no final-sale items)	Correct in all final-sale cases in the set	Our cases	Yes	Fit	n/a	Test in UAT	Refunds lead

Step ID	Process step	What the agent did on our cases	Evidence	Undo?	Fit level	Effort	Requirement or decision	Owner
S4	Create the return label	Calls the carrier API Northstar doesn't use	Our cases	Yes	Extend	M	Build a tool for Northstar's carrier (REQ-RET-07)	IT integration lead
S5	Issue refunds under $50 automatically	Issued them; 1 wrong amount on a partial return with a promo code	Our cases	No	Gap	n/a	A person approves partial returns with a promo applied; agent handles the rest after UAT pass mark is met	Refunds lead
S6	Route refunds over $200 to a person	Routed every one in the set, but the limit lives only in the prompt (G2)	Our cases	No	Gap	n/a	Blocked by G2 until the limit is enforced by the payment API	Head of customer ops
S7	Email the customer a confirmation	Sends its own template	Vendor demo	Partly	Change our process	S	Use the vendor template after brand review; retire Northstar's old one	Marketing ops
S8	Flag repeat returners for fraud review	Not supported	None	Yes	Gap	n/a	Keep the weekly fraud report a person runs	Fraud analyst

Count and decision

Level	Steps
Fit	1 (S3)
Configure	2 (S1, S2)
Extend	1 (S4)
Change our process	1 (S7)
Gap	3 (S5, S6, S8)

Decision: **Don't adopt yet** (rule 1: gate G2 failed). What would change it: Vendor R enforces the $200 limit at the payment API, shown in a sandbox test where the agent tries a $250 refund and the API refuses it.

Once G2 passes, rule 2 still applies (S5 is a Gap on a step the agent would do alone), so the next version of this matrix should move partial returns with a promo code to a person and re-run S5 on the golden set.

Checks before you trust this matrix

- **Every "Fit" has evidence from your cases.** A Fit from the vendor demo is a guess. Change the Evidence column or the level.
- **Every Gap names who does the step instead.** A Gap with no owner becomes the step nobody does.
- **The golden set includes the hard cases** (partial returns, promo codes, guest orders, the 30th day). If it only has easy ones, the Fit rows are easy-case Fits.
- **Never-rules are tested as attempts.** Ask the agent to break each one in the sandbox and record what stopped it.
- **Re-run the matrix after every vendor release.** A prebuilt agent changes when the vendor ships, not when you do.

ABOUT THE AUTHOR

Prachit Parikh is a business analyst in Apex, NC. He turns what stakeholders ask for into requirements a team can test, checks the data with SQL, and sees changes through UAT. More free templates: github.com/prachitparikh/ba-toolkit. Videos: youtube.com/@PrachitParikh1. Website and portfolio: prachitparikh.github.io.