Agency Delivery-Fit Checklist

For boutique agencies deciding whether a client AI, data, analytics, or automation request needs a specialist technical delivery partner.

Purpose

This checklist helps an agency decide whether a client request belongs inside the agency team, inside a no-code workflow, or with a technical delivery partner.

Use it before promising AI, personalization, reporting, data activation, attribution, lifecycle automation, client portals, enrichment, or backend integration work.

Quick fit rule

Bring in technical delivery when at least three are true:

- The project depends on data from more than one system.
- The workflow needs branching logic or transformations.
- The client expects the output to run repeatedly without manual repair.
- A failed automation would create client-visible risk.
- The implementation touches CRM, warehouse, product analytics, BI, APIs, or event data.
- The agency would need to hire or subcontract outside its core capability to deliver it well.

1. Client promise clarity

Before scoping technical work, clarify the actual promise.

Questions:

- What will the client be able to do after this is live?
- Is the promise strategic, creative, analytical, operational, or technical?
- Does the client need a campaign asset, a dashboard, a workflow, a data product, or a production automation?
- What manual work disappears?
- Who uses the output every week?

Risk signal:

If the client says "AI" but cannot name the workflow, user, decision, or output, start with feasibility rather than build.

2. Data dependency

Most agency projects become technical when data is part of the promise.

Questions:

- Which systems contain the needed data?
- Is the data in CRM, product analytics, billing, support, spreadsheets, warehouse, email tools, ad platforms, or event streams?
- Is there a reliable API or export path?
- Does the client trust the source data?
- Is identity resolution needed across systems?

Risk signal:

If the agency needs to reconcile customer, campaign, lead, product, or revenue data across systems, treat it as a technical delivery project.

3. Automation complexity

Simple no-code workflows are fine when the logic is simple. Technical delivery is needed when reliability matters.

Questions:

- Does the workflow need retries, error handling, logging, or alerting?
- Are there more than two steps?
- Are there conditional branches?
- Does data need cleaning, joining, deduping, scoring, or transformation?
- Does the workflow update important business records?

Risk signal:

If silent failure would hurt the client relationship, do not rely on a fragile no-code setup without monitoring and ownership.

4. Handoff and ownership

The client should be able to operate what gets delivered.

Questions:

- Who owns the workflow after launch?
- What documentation will they need?
- What happens when a source field changes?
- How will they know if the workflow fails?
- Is the agency expected to support the system indefinitely?

Risk signal:

If nobody owns the system after delivery, the agency may inherit support work it did not price.

Recommended collaboration model

Agency owns:

- Client relationship
- Strategy and creative/product framing
- Account communication
- Brand, UX, campaign, or lifecycle direction
- Commercial relationship

MLDeep owns:

- Technical feasibility
- Data/system inspection
- Backend workflow scope
- AI/data/analytics implementation
- Runbook and technical handoff

Best first scopes

Strong first scopes include:

- Lead routing or enrichment workflow
- Weekly client KPI brief
- Campaign performance reconciliation
- Lifecycle segmentation logic
- CRM cleanup automation
- Client-facing reporting layer
- Product usage to CRM sync
- AI-assisted support or sales triage with human review

NEXT STEP

Bring one client scenario to a partner fit call.

20-min call. Bring the client situation, the technical question, and the decision that needs to happen next. Leave with a written go/no-go — no slide deck.

Book a partner fit call → or email info@mldeep.io