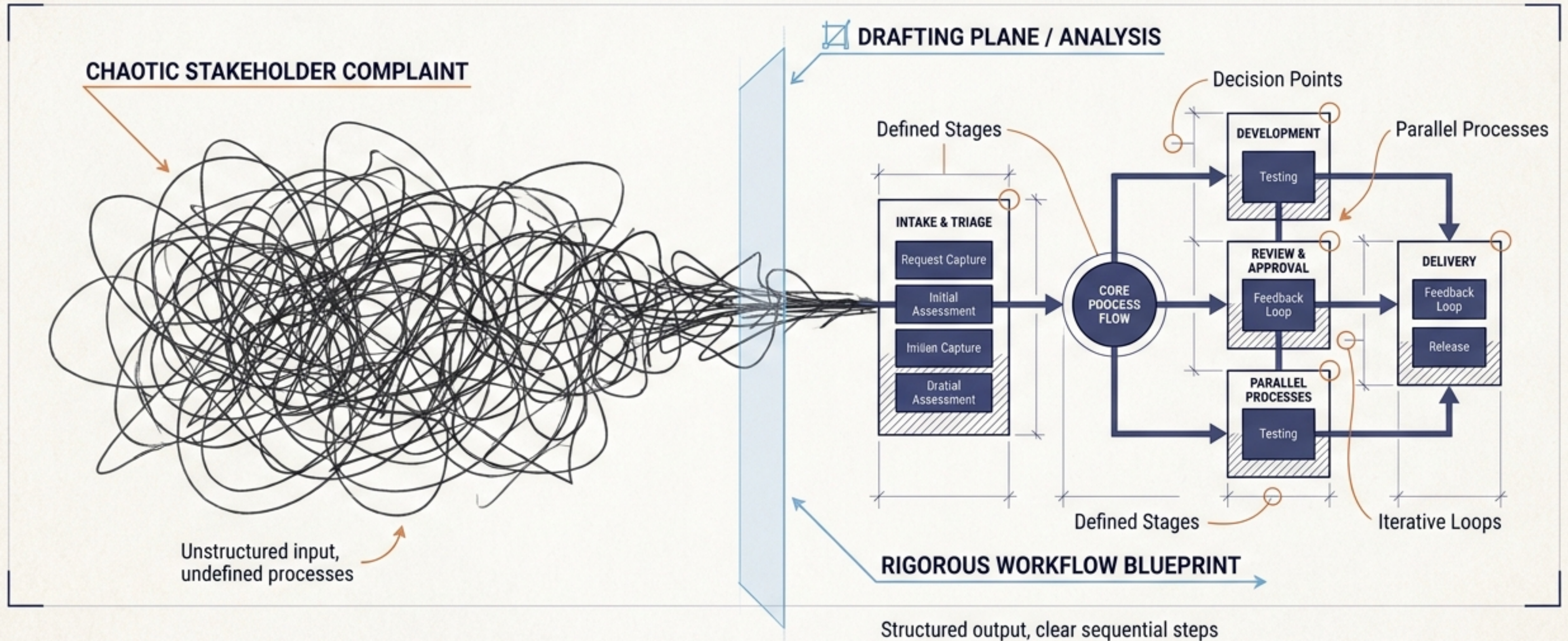


Workflow Discovery

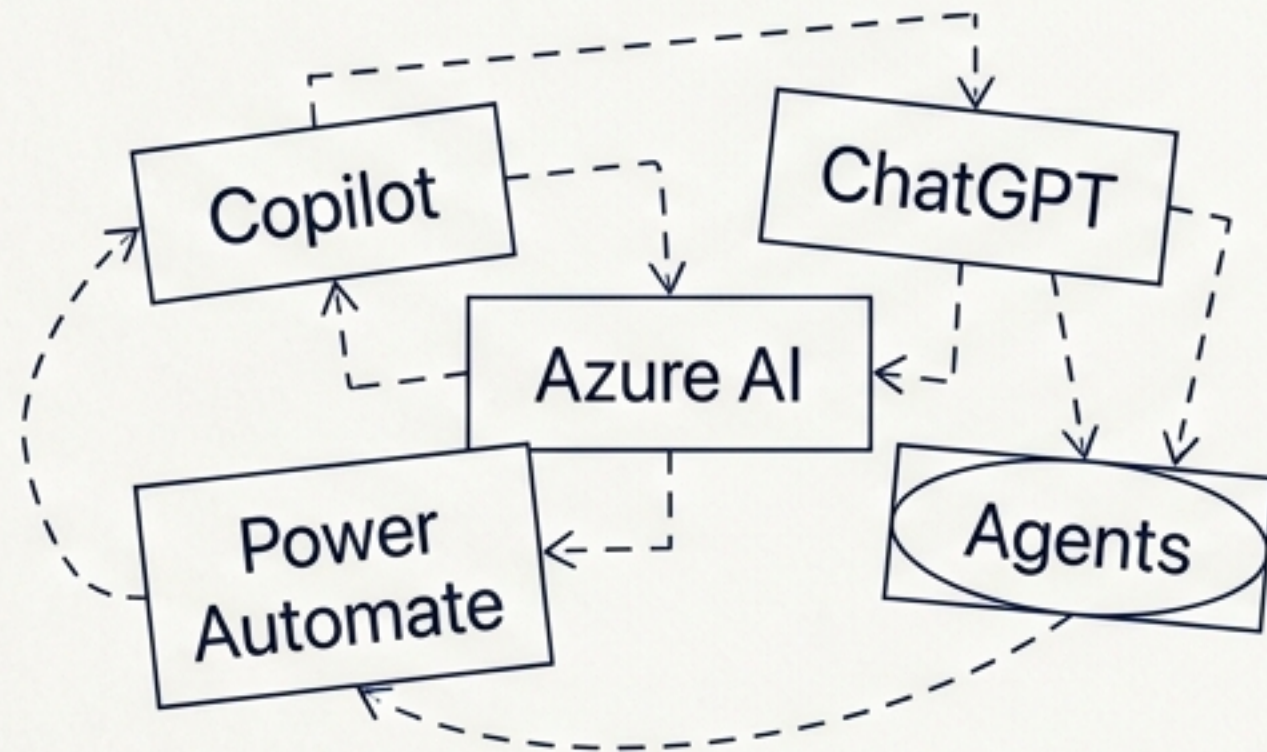
Start with the real work before choosing the tool.



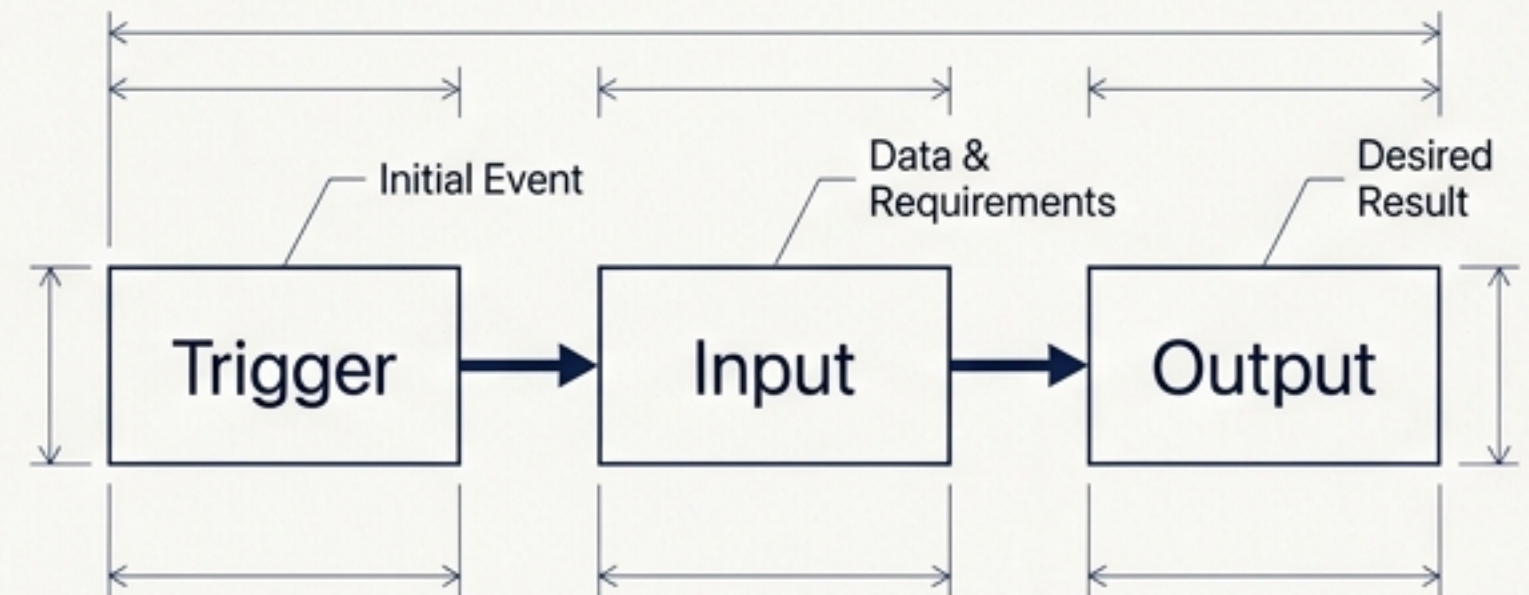
Failed automation is rarely a model issue

Most AI conversations begin with the tool: "Can AI fix this?" Teams must slow down. Failure happens because the team did not understand the work well enough, not because the model was weak.

The Tool-First Trap



The Work-First Approach



Slowing down before solutioning prevents overbuilding

A vague idea must become a clear description before any technical work begins.

The Tool-First Illusion

The Workflow Discovery Approach

Starts with a vague idea or complaint

Describes the current baseline process

Jumps straight to Copilot Studio or Power Apps

Maps pain points and verifies data sources

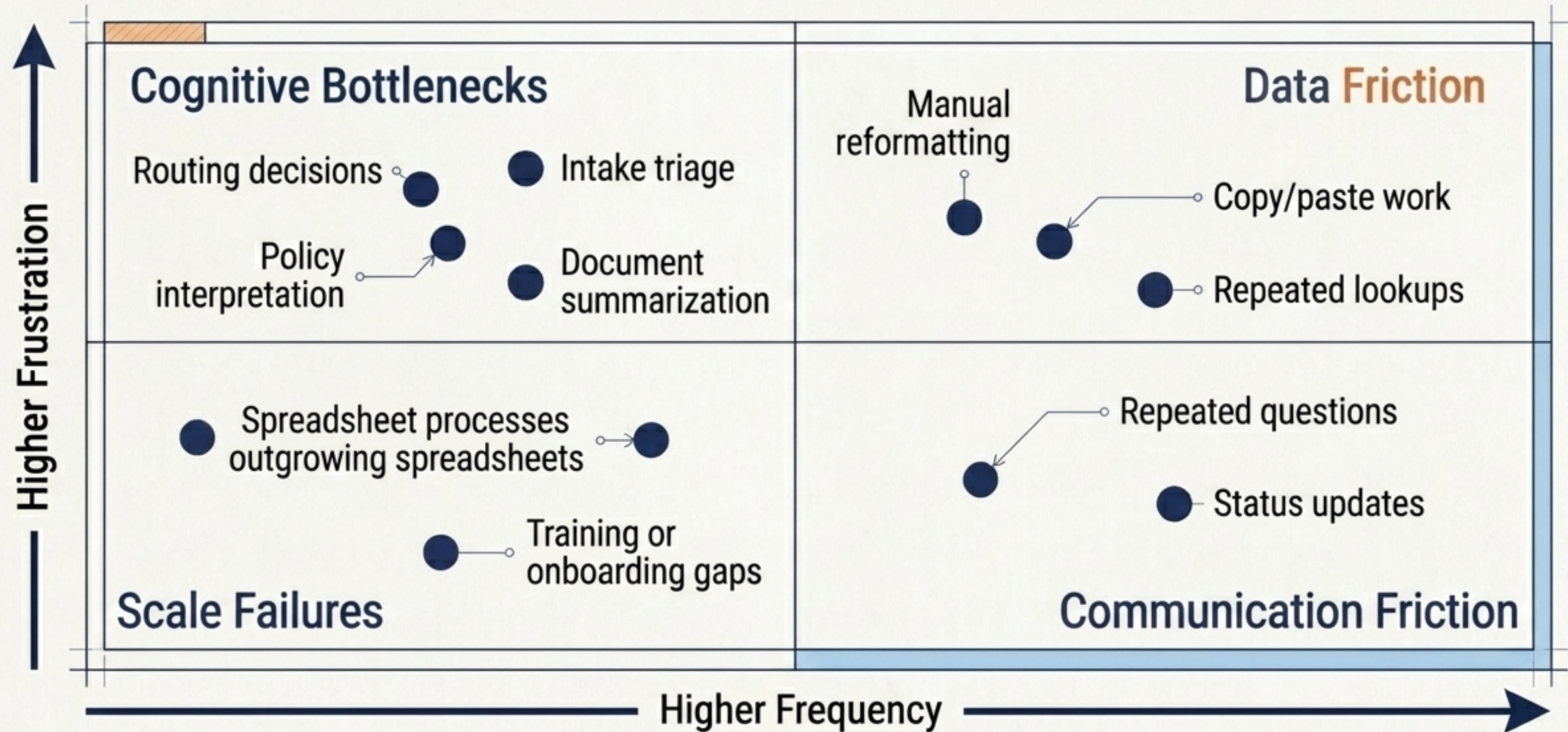
High risk of overbuilding and technical debt

Reduces risk by finding structural flaws first

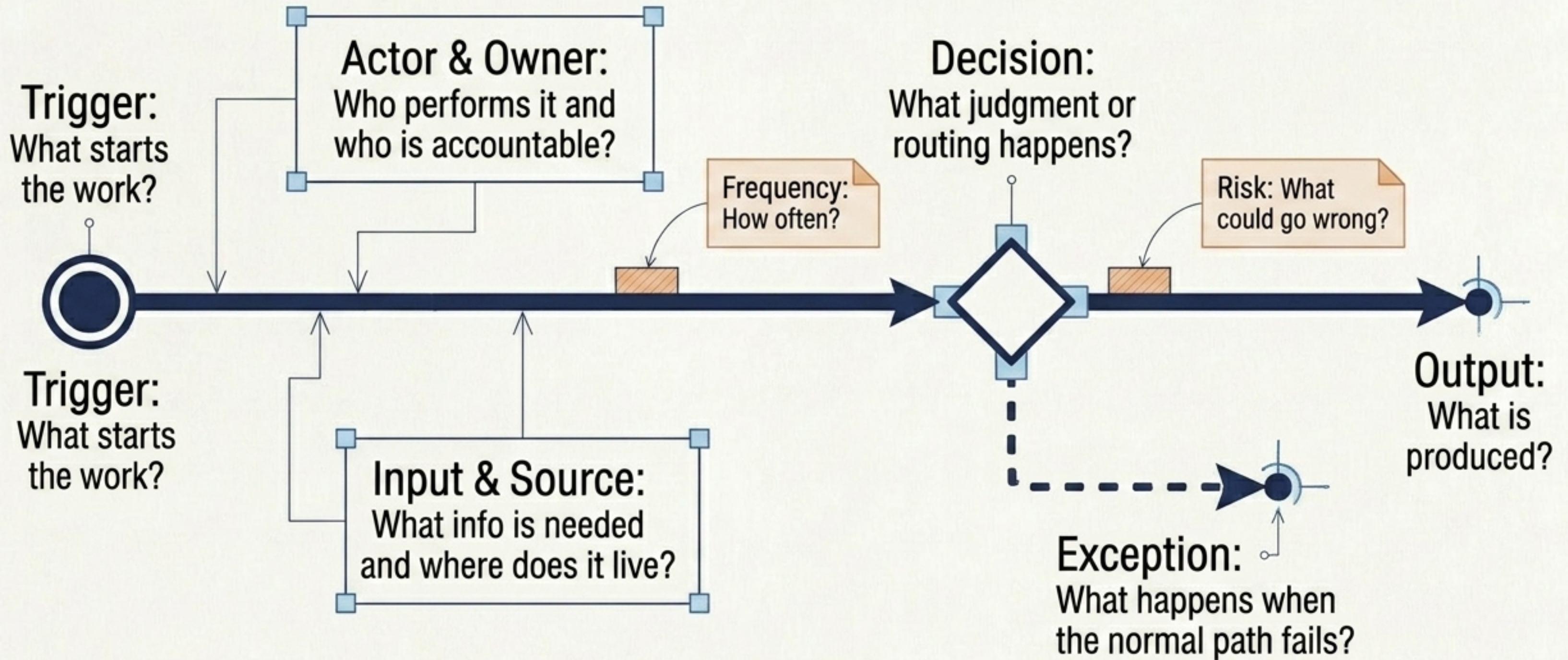
Poor quality requests reach technical teams

Makes employees feel heard and understood

Signals that indicate an opportunity for optimization

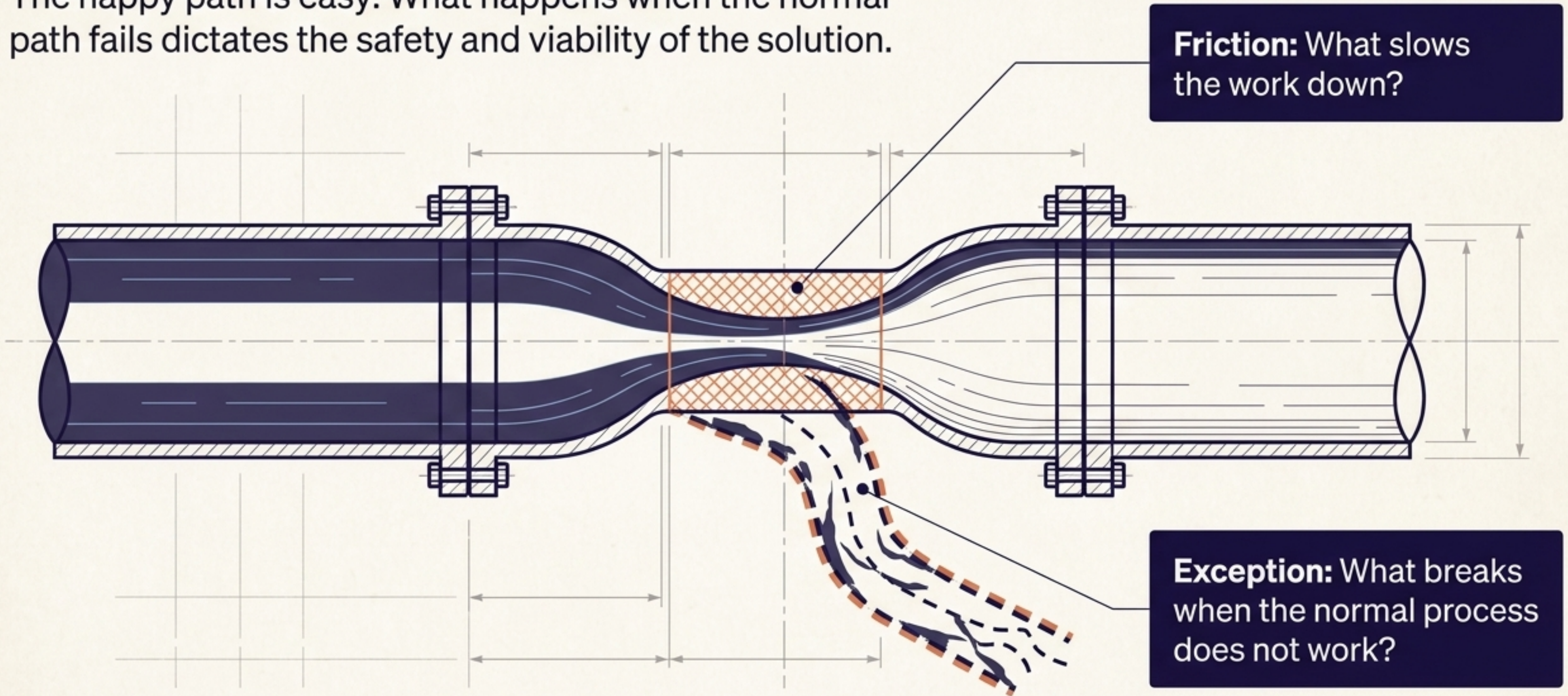


The anatomy of a functional workflow



Exceptions determine whether automation is actually safe

The happy path is easy. What happens when the normal path fails dictates the safety and viability of the solution.



Essential discovery questions to reveal the hidden process

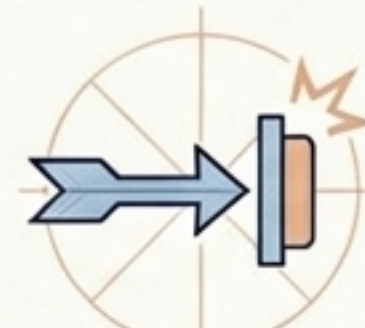
- Keep asking these until the underlying workflow is clear.



What work is actually happening?



Who does it?



What triggers it?



What information is needed and where does it live?



What decisions are made?



What is repetitive, risky, or frustrating?



What would a good outcome look like?

Listen for the hidden workarounds

Discovery is not just an audit. Instead of jumping to a tool, the facilitator helps the team describe what is actually difficult.

Workarounds:

What workarounds have become normal in this process?

Manual Labor:

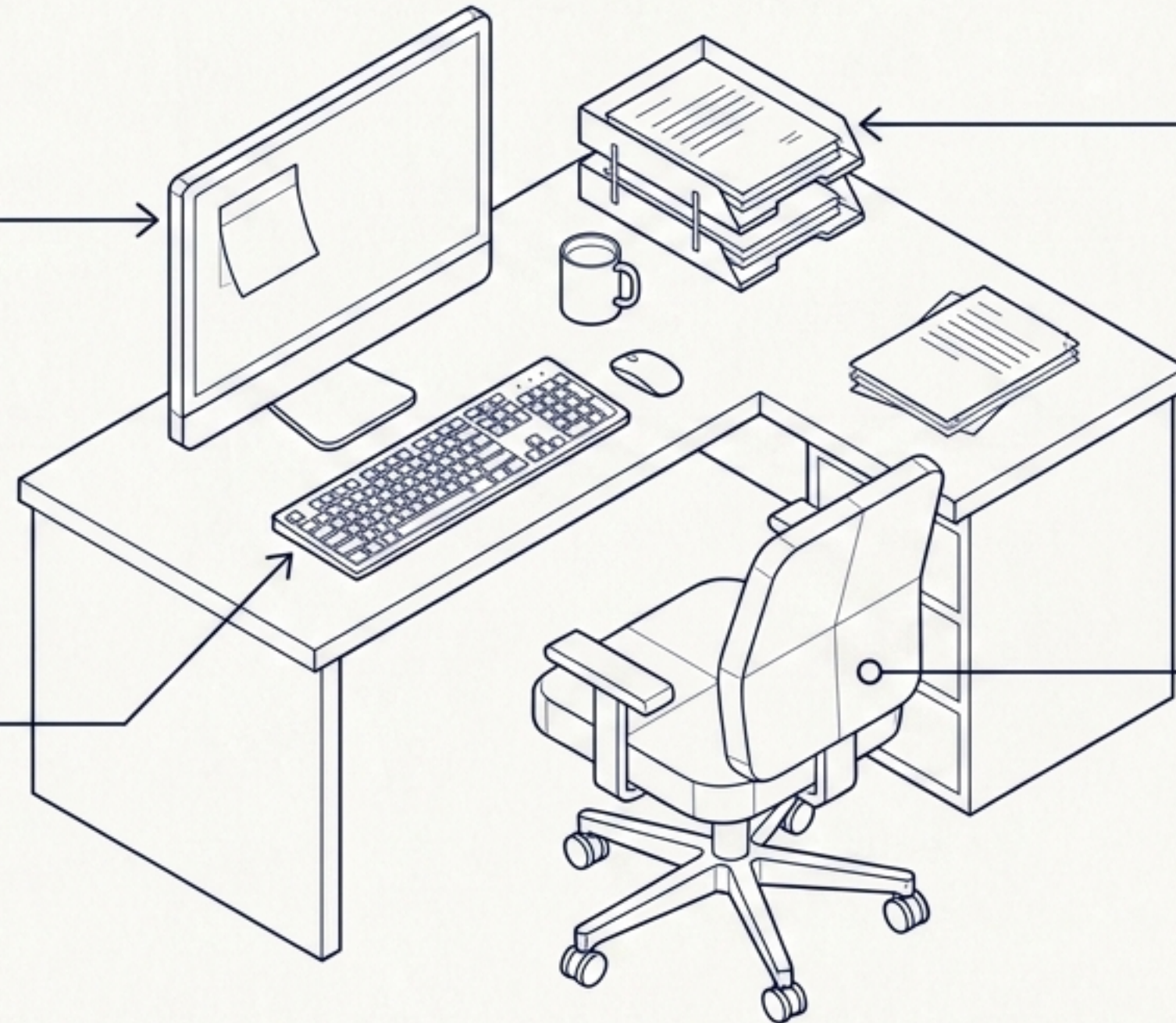
Where do people copy, paste, retype, or reformat?

Repetition:

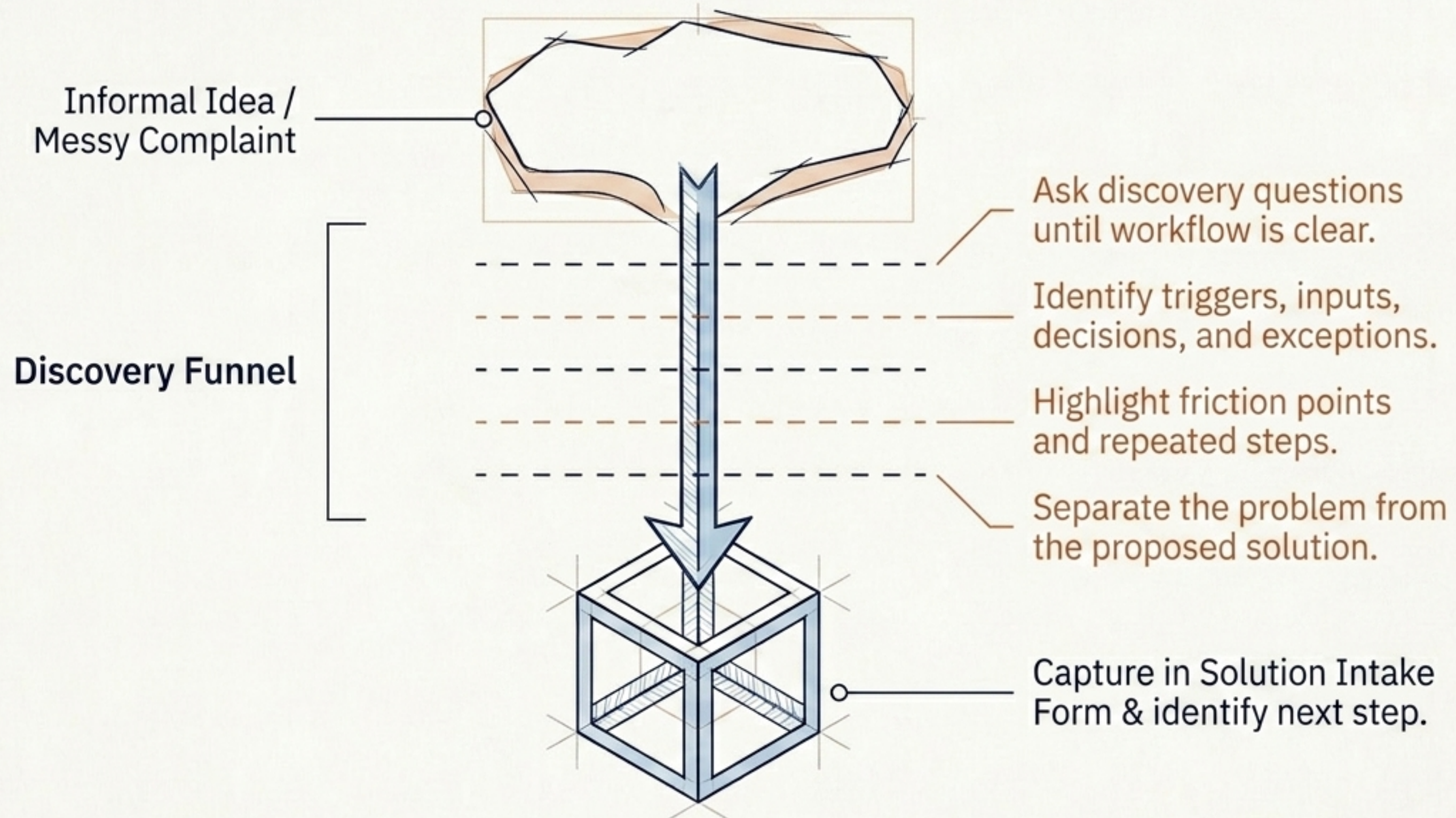
What questions are asked repeatedly?

Ownership:

Who owns the process today, and who should own the improved version?



Refining a messy complaint into a structured blueprint



Separating the problem from the proposed solution

The Complaint

"I need Copilot to handle all my email triage, I spend hours copy-pasting client requests into the tracker."

**Workflow
Discovery**

The Problem Statement

Trigger	New client email arrives
Action	Manual copy/paste into tracker
Exception	Missing client information
Friction	Hours spent transferring data

Misdiagnosing the work leads to the wrong solution.

The discovery output should be useful even if the final answer involves no AI at all.

Repetition $\neq$ Automation

Just because a task happens often does not mean automation is appropriate.

Unclear Ownership = Process Issue

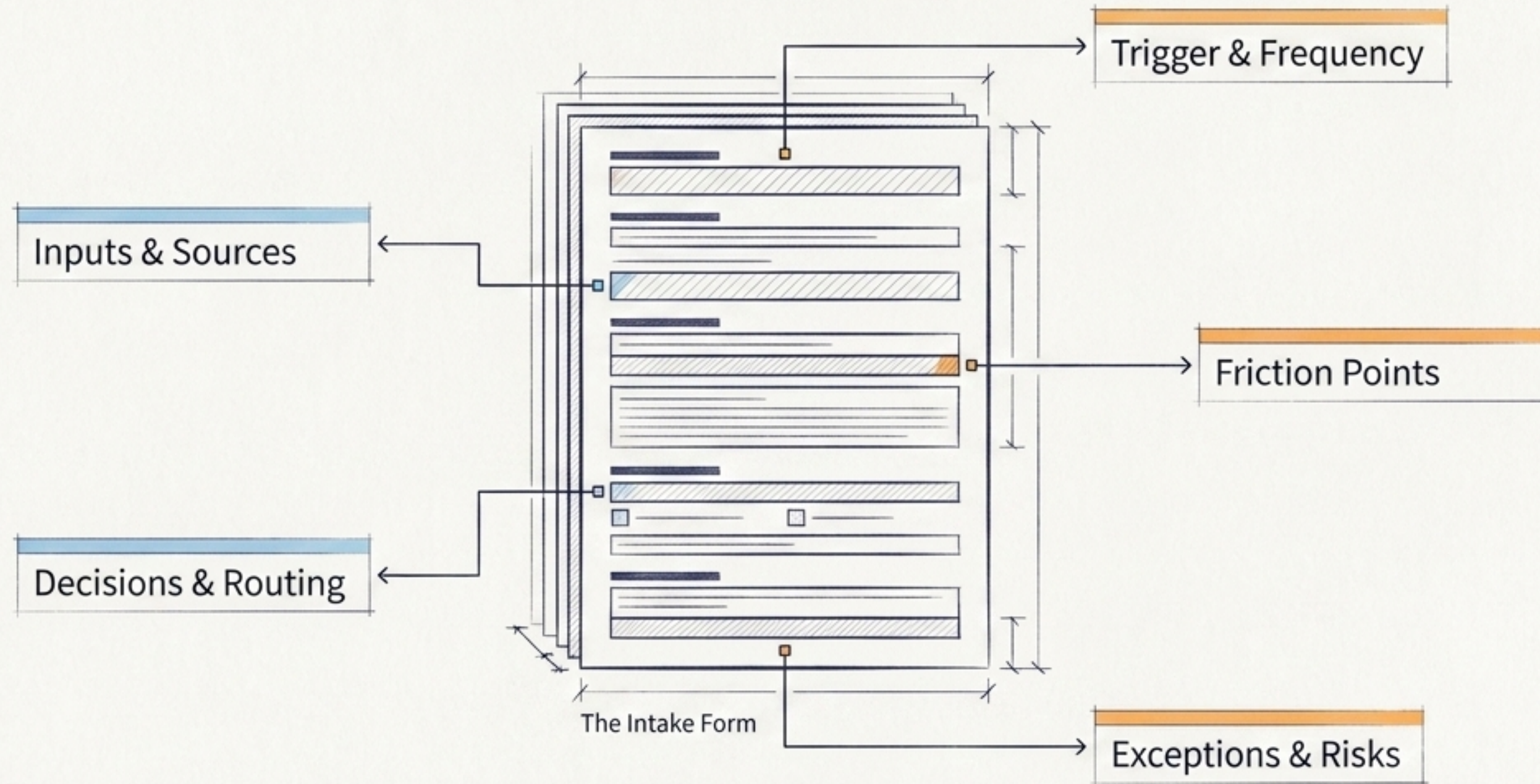
AI cannot fix a broken chain of command. Resolve the process first.

Tool $\neq$ Problem

A proposed tool is not the same as a defined problem.

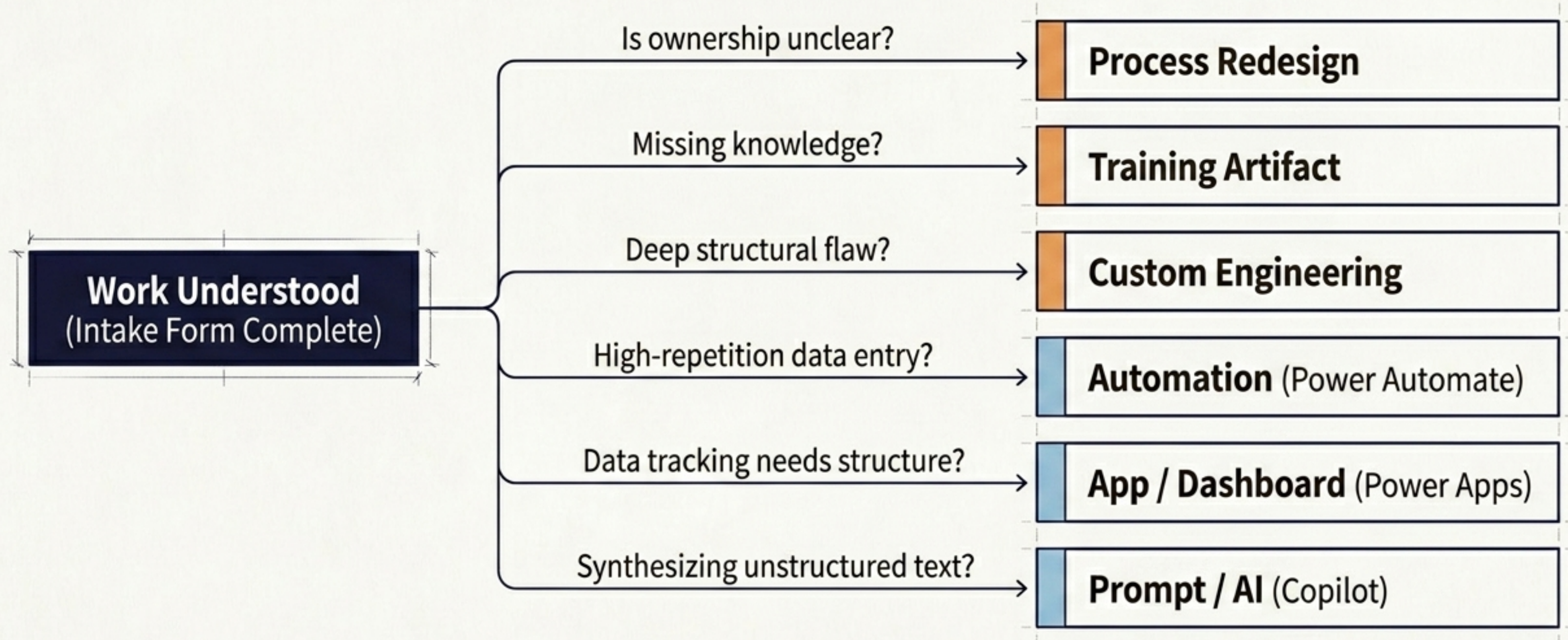
The Solution Intake Form captures the complete picture.

This form translates stakeholder pain points into a structured use case, ready for technical assessment.



Routing the diagnosed problem to the right solution

Only after the work is understood should the team decide the answer. Sometimes, the best answer is no AI at all.



The outputs of a successful discovery session

A structured process prevents overbuilding and ensures that whatever tool you choose actually solves the real problem.

BLUEPRINT SIGN-OFF SHEET	
☒	A clear problem statement.
☒	A simple workflow map or step list.
☒	A list of friction points.
☒	A list of data sources and owners.
☒	A completed draft solution intake form.
☒	A preliminary recommendation for next-step assessment.