

What is Process Discovery?

A clear definition, the categories it gets confused with, and why it matters more now.

BY CARRV.AI

7 MIN READ · MAY 2026

CARRV.AI

ENTERPRISE INTELLIGENCE, CARVED AT THE SOURCE.

One sentence.

The term has been borrowed by enough adjacent categories that anyone coming to it cold has no idea what it means. Three different kinds of software claim it. Only one of them actually does it.

So let me start with a single sentence and then defend it.

DEFINITION

Process discovery is the structured, real-time capture of how work happens inside enterprise software — across the people and roles it passes through, at the screen, while it's happening, by watching.

Everything that follows is what each word is doing, and why this version of "discovery" is a different thing from what the previous decade meant by the word.

Sticky notes and Sharpies.

If you ran a process improvement project in 2015, "discovering the process" looked something like this.

You scheduled a workshop. You pulled eight people into a room. You handed them sticky notes and a Sharpie. Someone drew a flowchart on a whiteboard. A junior consultant photographed it, took it back to the hotel, and rebuilt it in Visio that night.

The output was a BPMN diagram. The diagram was wrong almost immediately — partly because people in workshops describe what they should do, not what they actually do, and partly because the underlying software changed three months later and nobody updated the diagram.

This is how nine out of ten Fortune 500 companies still document their processes. It's also why most enterprise documentation has the truth-content of a horoscope.



Software watches software.

In 2026, process discovery means something narrower and more useful. It means putting a piece of software on the actual workstation where the actual work happens, letting it watch the entire session, and capturing — with structure, not just video — what the person did.

Every click. Every typed value. Every window switch. Every wait. Every correction. With the screen captured at each step, the target element identified, the input field labeled, the decision points marked.

The captured artifact isn't a video. It isn't a transcript. It's a structured representation of one complete way of doing the task — replayable, queryable, generatable into a dozen downstream formats.

The shift, in one sentence: *discovery used to be an interview. Now it's an observation.*

"At the source" means the capture happens where the work is — at the user's interface, while they are working — not somewhere downstream where work has already left its residue.

Mining tools work downstream. They read the logs that underlying systems produced. They reconstruct what likely happened from the trail. That trail is useful, but it's a trail. It's not the thing itself.

Process discovery isn't the trail. It's the work, watched.

Three things people call mining.

The full comparison is its own essay. But quickly, since most people arrive here from a search that conflated the three:

PROCESS MINING

Reads event logs.

GOOD AT

Where are our bottlenecks?

BLIND TO

Anything outside the system of record.

TASK MINING

Aggregates clicks.

GOOD AT

Where does time go?

BLIND TO

One discrete procedure, end to end.

PROCESS DISCOVERY

Watches the screen.

GOOD AT

What is the actual procedure?

BLIND TO

Nothing structured by software.

Mining gives you statistics. Discovery gives you the procedure.

A process has many hands.

Here's what most of the writing about discovery — including, until two paragraphs ago, this essay — quietly gets wrong.

A process is rarely one person at one screen. It's a chain.

Take invoice processing. An AP clerk receives the invoice, enters it into the ERP, codes it, attaches the PO. A line manager reviews and approves. Finance posts it. Treasury releases payment. Four roles. Four sets of screens. Four sessions.

If you capture only the first agent's screen and call that "the process," you have a fragment — one link in a longer chain. You don't have the process.



invoice processing — four hands, one process.

Real process discovery is an assembly problem as much as a capture problem. The platform has to capture each role's session on their own workstation, identify the handoff points, and stitch the segments into one continuous end-to-end walkthrough — with the actors tagged, and the variations preserved across paths.

A process discovered is a process traced through all the hands that touched it. Not one of them.

Why now.

A decade ago, none of these pressures existed at the intensity they do now. Discovery was a nice-to-have. It's becoming the substrate that everything else — training, automation, compliance, AI — depends on.

Continuous shipping.

01

Software ships every two weeks now, not every twelve months. Documentation that takes a quarter to write is documentation that's wrong before it ships.

SaaS sprawl.

02

The average mid-market enterprise runs 130-plus SaaS apps. Documenting each by hand isn't a budget line — it's arithmetic with no solution.

AI needs process data.

03

When you ask an AI to handle invoices or close tickets, it needs the actual workflow — not a Confluence page from 2022. Process data is the substrate.

Compliance shifted.

04

Auditors want evidence, not assertions. "Show me the actual sequence followed, with timestamps and screens" is what an SOP library can't answer. Discovery can.

What you get.

A process discovery tool, run on a session of work, should produce at minimum:

01 **Ordered steps**

Each tagged with the application, the action type, and the target element.

02 **Actor + role tagging**

Identifies who performed each step, where the process passes through multiple hands.

03 **Screen captures**

A screenshot of the screen at the moment of each step.

04 **Bounded regions**

A bounding region indicating where on the screen the action happened.

05 **Annotations**

Context the action alone doesn't reveal — why a value, what the rule was.

06 **Decision points**

Where the flow could have branched, with the variants captured across multiple runs.

From that single artifact, downstream tools generate the deliverables: simulations, written SOPs, video walkthroughs, SCORM packages, structured feeds for AI agents. When the software changes, you recapture. You don't rewrite.

Adjacent, but not the same.

A few categories get confused with process discovery often enough to be worth naming.

IT ISN'T SCREEN RECORDING.

Pixels, not structure.

Screen recording produces video. Video is flat. You can't ask a video "show me only the steps where the user clicked Approve" because the video doesn't know what Approve was. Discovery produces structure. Recording produces pixels.

IT ISN'T BPM.

Bottom-up, not top-down.

BPM (Business Process Management) is top-down: you model the process you want, you orchestrate it with a workflow engine, you measure adherence. Discovery is bottom-up: you watch what actually happens, and represent it after the fact. Different jobs.

IT ISN'T RPA.

Observation, not execution.

RPA (Robotic Process Automation) is execution — a bot performs steps autonomously. Discovery is observation — software watches a human perform steps. They're often used together: discover first, then automate. Different categories.

If you only have ten seconds.

“

The work happens on a screen.
Usually on several screens, across
several roles. Process discovery is
the software that watches each one,
stitches the chain together, and
produces the artifact every
downstream system needs as input.

TRAINING · SIMULATION · AUTOMATION · AI

The decade-ago version of discovery was a workshop.
The 2026 version is an observation.

That's the entire shift.

CARRV.AI

ENTERPRISE INTELLIGENCE, CARVED AT THE SOURCE.

What is Process Discovery?

FOUNDATIONS · MAY 2026