

Why Documentation Decays.

Documents are written once and rot from that day onward.
Why simulations captured at the source are the only
training artifact that survives the next release.

BY CARRV.AI

6 MIN READ · DEC 2025

CARRV.AI

ENTERPRISE INTELLIGENCE, CARVED AT THE SOURCE.

A new hire, a stale screenshot.

A new hire on her first day. Procurement role. She has the onboarding SOP open on her left monitor, the actual application on her right. Page four of the SOP, step seven. The screenshot shows a button in the top-right corner of the screen labeled Submit for Approval.

The screen in front of her does not have that button. The button is now in a menu under Settings, three clicks away, labeled Send to Reviewer. The application was upgraded four months ago. The SOP was last updated nine months ago. Nobody has had the bandwidth to reconcile the two.

She does what every new hire does. She emails her manager. Her manager forwards it to L&D. L&D adds it to a backlog of forty-three similar reports from this month alone. The SOP doesn't change. The next new hire next month will hit the same problem.

This is how enterprise documentation actually fails. Not catastrophically. Quietly. One stale screenshot at a time, multiplied across thousands of procedures.

The clock starts on day one.

The moment a piece of enterprise documentation is finished, the clock starts on its decay.

This isn't a complaint about lazy writers or under-resourced teams. It's a structural fact. Documentation is a static description of a moving target. The screenshots show a UI that will redesign in March. The steps reference a button that will be renamed in April. The terminology matches a workflow that will be re-engineered by the end of Q2.

I've watched companies spend a meaningful share of their L&D budget keeping documentation accurate. None of them win the fight. The documentation rots faster than the team can rewrite it.

What you encode, what moves.

When you write down a procedure, you're encoding several things at once: the names of the applications involved, the look and feel of their interfaces, the labels on the buttons, the terminology your company uses for the work, the integrations between systems, the compliance requirements in effect at the time, the roles involved, and the names of the teams owning each step.

Every one of those is a moving piece. Vendors rename features. Companies restructure teams. Compliance regimes update. Buttons move. Terminology drifts. The aggregate effect is that a procedure written in January is partially wrong by March and substantially wrong by September.

The half-life of corporate documentation is not a year. It is closer to a quarter.

Why the maintenance approach fails.

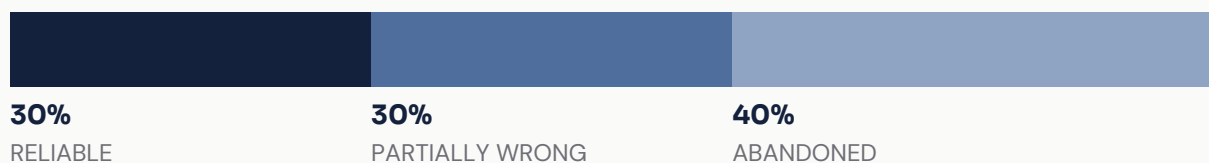
The standard response to documentation decay is to invest in maintenance. Quarterly review cycles. Owner assignments. Confluence tags. SLAs on documentation freshness.

This works for the highest-priority documents — the ones tied to compliance or critical onboarding. It fails everywhere else, because the math doesn't work.

Consider a typical mid-market enterprise. Two hundred documented procedures. Each one takes six hours to review and update properly. That's 1,200 hours of skilled reviewer time per quarter — about five FTEs of effort to do it right across the year.

Almost no L&D team gets that headcount. So they triage. The top twenty procedures stay current. The next eighty get reviewed annually. The remaining hundred rot quietly.

BY THE THIRD YEAR, A TYPICAL DOC LIBRARY LOOKS LIKE



The L&D team knows this. The auditors don't, until they ask.

Capture, don't write.

The structural problem with documentation is that it's a representation of a procedure, not the procedure itself. When the procedure changes, the representation has to be hand-updated.

What if the document and the procedure were the same artifact?

This is what captured simulations do. You record an expert performing the procedure once — the actual clicks, the actual screens, the actual decision points. The capture becomes the source of truth. The training material, the SOP, the video walkthrough, the SCORM module — all of them are generated from that capture, not written separately.

When the underlying software changes, the procedure changes. So you recapture. Twenty minutes of an expert doing the new version replaces a week of writer time updating Confluence. The outputs regenerate from the new capture.

This is not a marginal improvement. [It's a different operating model.](#)

Five FTEs become one.

A documentation team running on captured simulations doesn't budget for maintenance the same way. The labor profile shifts from writing to capturing and curating. The unit of work shifts from hours per page to minutes per procedure.

THE WRITTEN-DOC MODEL

1,200

hours per quarter

5 FTEs

THE CAPTURED-SIMULATION MODEL

200

hours per year

1 FTE

This is the real efficiency unlock the documentation category is heading toward. It isn't AI-generated copy or auto-translated SOPs. It's the realization that the document was never the right primitive in the first place.

Why the industry hasn't moved.

Three reasons.

01

The category vocabulary is wrong.

“Documentation” is what enterprises buy. “Captured simulations” sounds adjacent — like a training tool, not a documentation replacement. Buyers with the L&D budget look for documentation tools, find writers and editors and Confluence-adjacent products, and stay in that lane.

02

The perceived transition cost is high.

Replacing a 200-procedure document library with a 200-procedure capture library feels like a large project. In practice it isn't — you don't migrate all at once; you recapture procedures as they need to be re-trained or audited. The transition is incremental. But the perceived switching cost keeps companies in the rotting-document equilibrium.

03

Capture tools used to be bad.

Five years ago, screen capture for documentation produced clunky video files with no structure, no branching, and no way to update incrementally. The technology has changed. The mental model hasn't caught up.

Documents rot because they describe a thing instead of being the thing.

Simulations captured at the source don't rot, because when the source changes, you recapture and regenerate.

The cheapest documentation is the documentation you don't have to maintain.

CARRV.AI

ENTERPRISE INTELLIGENCE, CARVED AT THE SOURCE.

Why Documentation Decays

AN OPINION ESSAY · DEC 2025