



Webinar On



The Blueprint for Quality Documentation

Metrics to Build, Track and Improve



Speaker

Khyati Sharma Karki
Documentation Specialist &
Communication Expert

Overview

1. **The Reframe** — why this is harder than "just writing"
2. **Discovery** — learning the business landscape
3. **Feedback Loops** — the right reviews, not more reviews
4. **Engineering Alignment** — knowledge nobody fully owns
5. **Knowledge Centralization** — building the trusted source
6. **The Finish Line** — deciding what "done" means
7. **Zooming Out** — how this looks across roles and industries

What Are We Actually Struggling With?

Not writing

Process

Ownership

Timing

Proving impact

“Technical writers are rarely struggling with writing. They're struggling with process, ownership, timing, and proving impact.”

Why Documentation Fails

Treated as	Instead of
The last task	Part of product development
One person's responsibility	Shared ownership
A deliverable	A product
Something measured by page count	Something measured by user success

Not because writers aren't skilled. Documentation fails when it's positioned on the left -- and almost everyone in this room has lived there at some point.

THE REFRAME

Writing isn't the difficult part.
Decision-making is.

Before I Write a Single Sentence, I Try to Reduce Documentation Risk

Question	Why I ask it	Risk if unanswered
Who is the audience?	Determines depth, tone, structure	Technically accurate but unusable
What decision does this document help make?	Keeps focus on outcomes, not features	Becomes info instead of a guide
Who owns technical approval?	Ensures technical accuracy	Reviews stall, ownership unclear
Who owns business approval?	Confirms intended business outcome	Late comments change scope
Is documentation in the project timeline?	Helps estimate writing/review effort	Becomes a release-day activity
Do I have access to the product?	Allows validating workflows/screenshots	Reflects assumptions, not reality
What artifacts already exist?	Prevents reinventing work	Time wasted rediscovering info
How will users get help beyond this doc?	Surfaces FAQ/troubleshooting needs	Disconnects from customer experience

Five Kinds of Hidden Work

01

Discovery

Learning the business landscape before trying to document the feature

02

Feedback Loops

Validating conflicting information with SMEs instead of treating disagreement as a review problem

03

Engineering Alignment

Identifying who actually owns knowledge when no single team has the full picture

04

Knowledge Centralization

Reconciling fragmented information into a trusted source

05

The Finish Line

Deciding what's "accurate enough" to publish

SECTION

Discovery

Learning the business landscape before trying to document the feature

- Identify the audience before the content
- Find the best communicator, not just the feature owner

The Situation Most of Us Walk Into

- Some of us join during the first release. Others join after years of product evolution.
- Documentation is expected almost immediately -- before the writer understands the product itself.
- The real challenge isn't learning the technology. It's learning **how the organization thinks about the technology.**

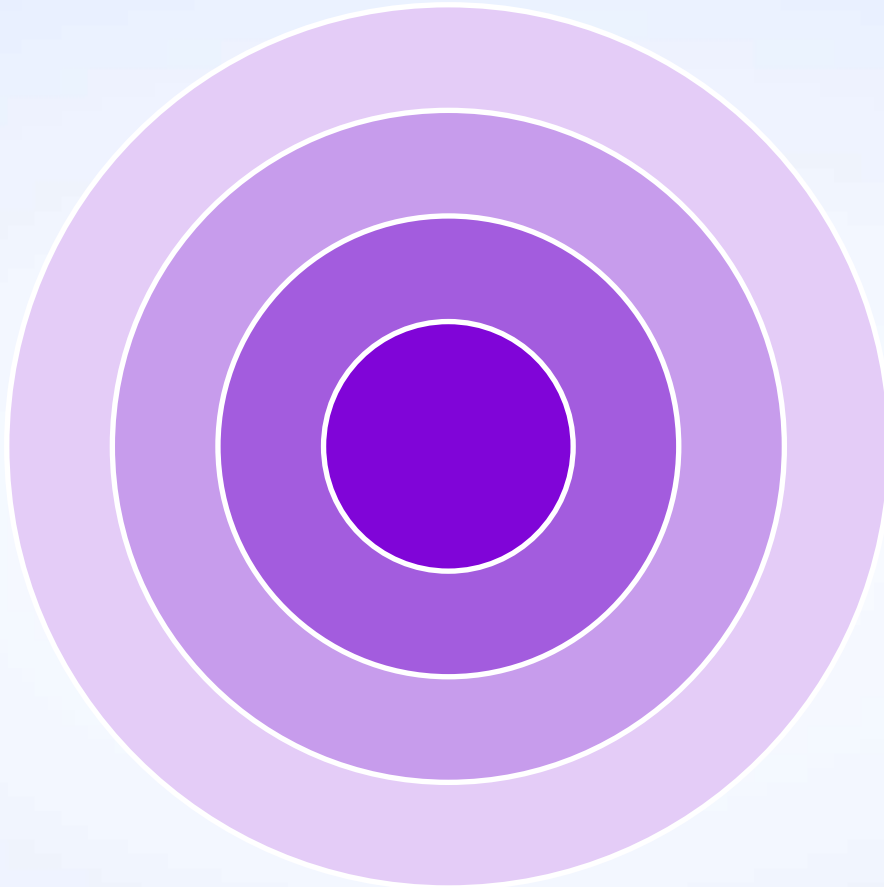
DOCUMENTATION BEGINS WHEN THE WRITER UNDERSTANDS

- Why this feature exists
- Who requested it
- What problem it solves
- What assumptions engineering is making

“The first challenge isn't asking better questions. It's discovering what questions even exist.”

- Which product owns this feature?
- Which version are we talking about?
- Is this feature being replaced?
- Which regions still use it?
- Who makes implementation decisions?
- Who owns customer vs. internal documentation?
- Is this for engineers, customers, support, or future teams?

Discovery Has Layers



-  1. Business discovery
-  2. Product discovery
-  3. Technical discovery
-  4. Documentation discovery

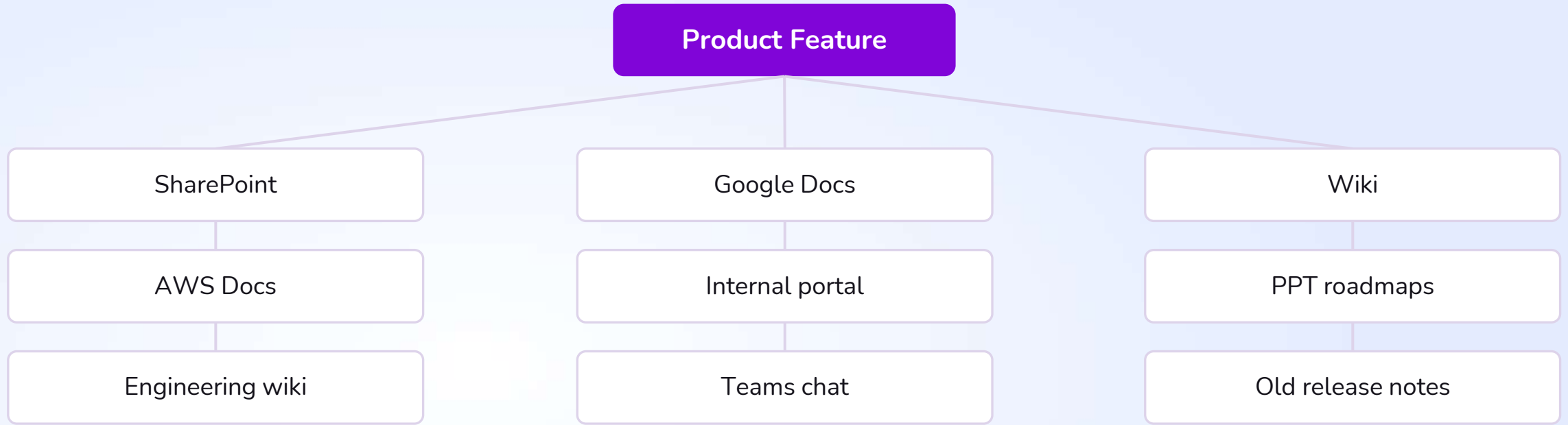
Most organizations jump straight to technical discovery.

“What Is the Actual Audience for This Document?”

- The request said the audience was system administrators.
- Halfway through review, support engineers said the guide was far too technical for the customers they worked with every day.
- **The document wasn't wrong. It was written for the wrong audience.**

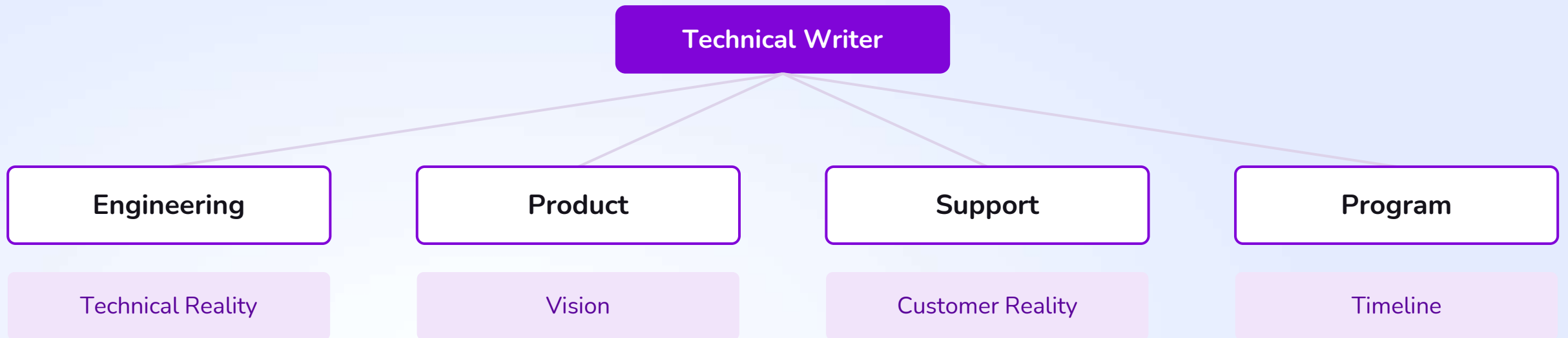
Organizational Archaeology

Where is the documentation?



Which one tells the truth?

Connecting Perspectives, Not Just Collecting Them



Everyone was correct. Nobody had the complete picture.

The challenge wasn't collecting information. It was connecting it.

SECTION

Feedback Loops

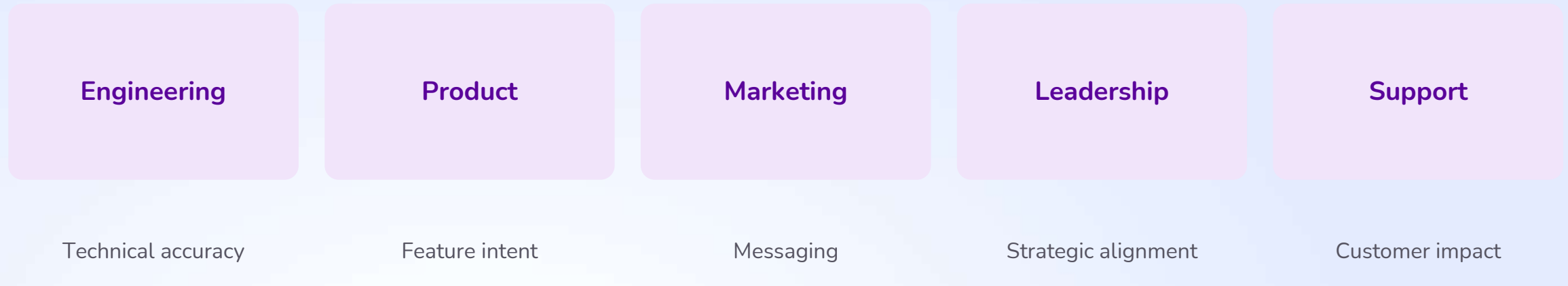
Validating conflicting information with SMEs instead of treating disagreement as a review problem

→ Assign review ownership before requesting reviews

“Who Has the Final Approval?”

- “We'll review it.” Everyone's heard this. It usually means nobody owns the review.
- The document sits untouched while every reviewer assumes someone else will respond.
- **The day one name appeared beside “Approver,” reviews stopped taking weeks. They started taking days.**

Documentation Doesn't Need More Reviews. It Needs the Right Reviews.



Each review is valuable -- the problem starts when everyone reviews everything.

INSTEAD OF “CAN YOU REVIEW THE DOCUMENT?” ASK:

- “Can you verify the workflow accurately reflects the implementation?”
- “Can you confirm this explanation is appropriate for our customer audience?”

Standing by Documentation

Feedback	Priority
Wrong step	High
Missing feature	High
Grammar	Low
Different wording preference	Low

Not all feedback deserves equal weight. This is the triage I use to keep review cycles from stalling on preference instead of accuracy.

SECTION

Engineering Alignment

Identifying who actually owns knowledge when no single team has the full picture

- Ask for test cases, not just specifications
- Make documentation visible in sprint planning

“I Need Access Before I Can Write”

- Specifications rarely tell the whole story.
- Real understanding comes from clicking through the product, making mistakes, reading error messages, seeing what the customer will actually experience.
- **Waiting until review to discover you misunderstood the workflow is one of the most expensive mistakes a documentation team can make.**

Test Cases Tell Stories That Specifications Don't

TEST CASES OFTEN REVEAL:

1

Validation rules

2

Error messages

3

Prerequisites

4

Common user mistakes

5

Recovery paths

6

Troubleshooting scenarios

03 ENGINEERING ALIGNMENT

Documentation Wasn't Late. It Was Never Planned.

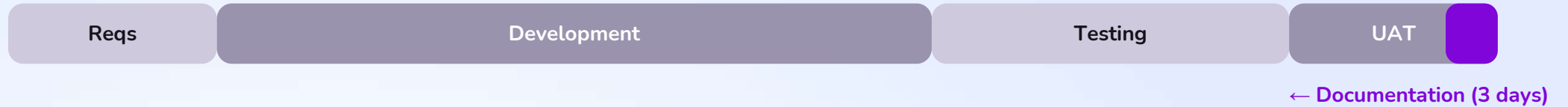
Wk 1

Wk 2

Wk 3

Wk 4

CURRENT — Product Release Plan



REVISED — Documentation Has Its Own Lifecycle



Notice something? Documentation isn't longer. It starts earlier.

SECTION

Knowledge Centralization

Reconciling fragmented information into a trusted source

→ Build a playbook because people forget, not because they're careless

The Decision That Changed Everything

I STOPPED ASKING

“Where is the documentation?”

- Searching for documents
- Copying existing content
- Documenting software

I STARTED ASKING

“Who owns this decision?”

- Searching for decision-makers
- Validating why content existed
- Documenting business decisions

A Good Playbook Doesn't Standardize Writing. It Standardizes Discovery.

Section	Why it exists	What problem it solves
Objective	Aligns everyone on why the document exists	Prevents feature dumps without purpose
Audience	Identifies who the document is for	Prevents rewriting after review
Intake	Captures missing info before writing	Reduces discovery time
SMEs	Identifies knowledge owners	Eliminates endless stakeholder hunting
Approvers	Makes ownership explicit	Prevents stalled reviews
Timeline	Makes documentation visible in planning	Avoids last-minute documentation
Glossary	Creates shared language	Reduces terminology confusion
References	Preserves organizational knowledge	Prevents duplicated research
Appendix	Keeps supporting material accessible	Avoids cluttering the main document

SECTION

The Finish Line

Deciding what's “accurate enough” to publish

→ Decide what “done” means before you start

Measuring Success, Differently

Question	If yes
Can users find it?	Find works
Can they skim it?	Scan works
Can they complete the task?	Read works
Do they come back?	Documentation builds trust

Stop Measuring Documentation. Start Measuring Friction.

Bad metric	Better metric
Pages published	Feature documentation coverage
Word count	Task completion rate
Page views	Search success rate
Number of writers	Time-to-publish
Articles updated	Support ticket reduction

Technical Writers Rarely Finish Learning the Product

- One misconception: expertise comes before documentation.
- In reality, documentation is often the process through which expertise develops.
- Every release: new terminology. Every feature: changed workflows. Every customer: new questions.
- **Experienced writers aren't the ones who know everything. They're the ones who know how to keep learning.**

ZOOMING OUT

There's No Single Framework That Works Everywhere

	SaaS writer	Medical device writer	Manufacturing writer
Source of truth	Product tickets + Slack	Regulatory design file + validation report	Engineering change order (ECO)
Who signs off	Eng lead, informally	Quality / Regulatory Affairs, formally	Change Control Board
Standard followed	Whatever the team agrees on	Good Documentation Practice, IFU requirements	Internal SOP format, sometimes ISO
What “done” means	Merged and published	Reviewed, signed, dated, version-locked	Approved by the board, distributed
Recurring problem	Docs start after code freeze	Getting SME time on a locked process	Change orders arrive after SOP updated

Same five kinds of hidden work. Completely different answers depending on where you sit.

ZOOMING OUT

Documentation Scales as an Ecosystem, Not a Hero Effort

Feature documentation — how the product works

Videos — common workflows

Marketing content — new capabilities

Support articles — recurring questions

Internal documentation — helps engineering

Reference guides — preserves org knowledge

“Good documentation doesn't scale because writers work harder. It scales because organizations stop expecting documentation to solve every problem alone.”

CLOSE

The hidden work is the job.

Discovery. Feedback. Alignment. Centralization. Knowing when it's done.

None of it shows up in a word count -- all of it shows up in whether the documentation actually works.

Questions ?

Thank You!

