

Technical Documentation Writing Practices



- Course scope: document types, audience analysis, writing principles, structure, style, review workflows, docs-as-code tooling, and maintenance



- Why quality matters for product teams, end users, and AI-assisted discovery in 2026



- Organization: foundational principles through workflows, tooling, collaboration, quality measurement, and a 30-day action plan

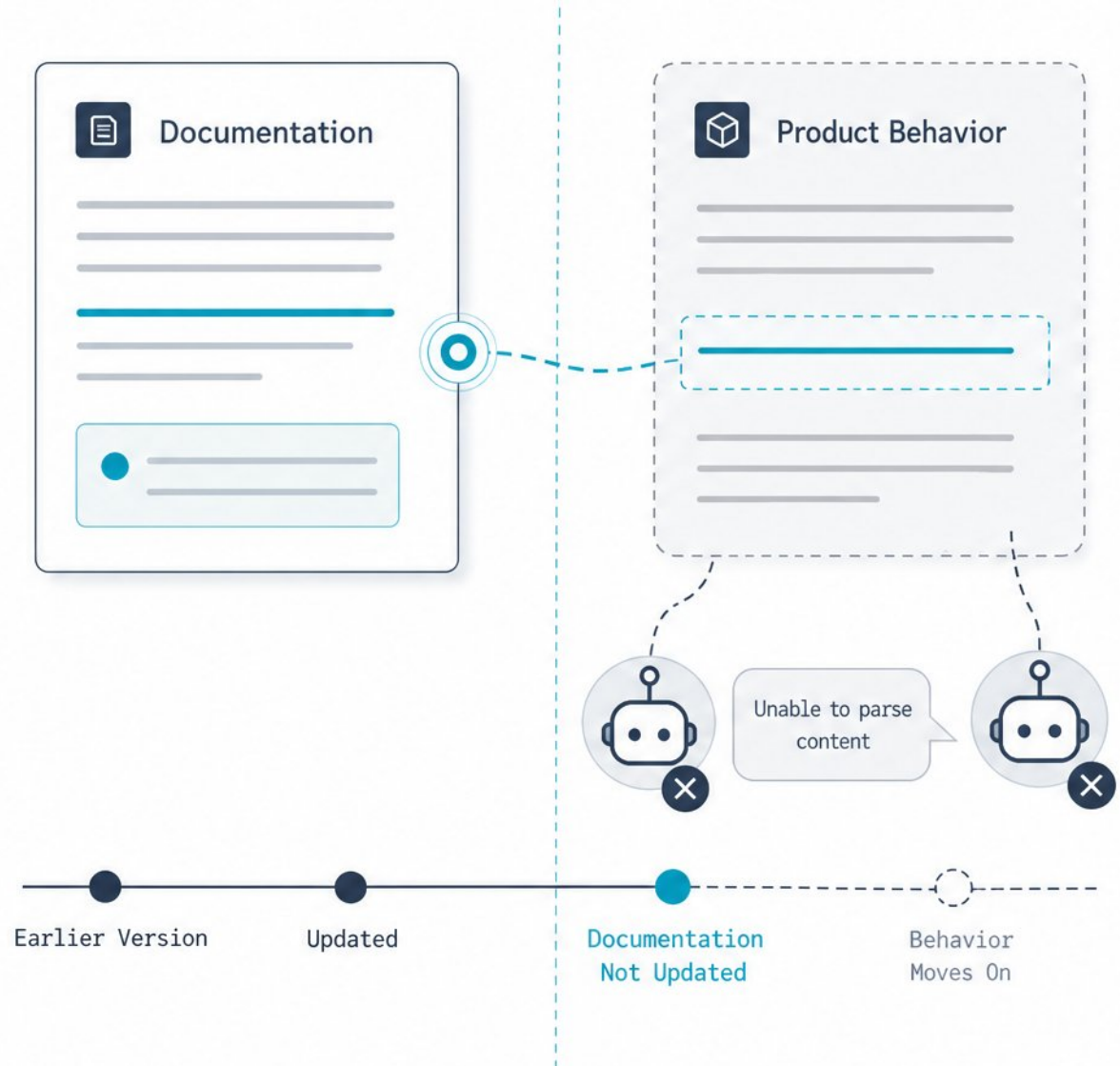


- Primary goal: clear, accurate, maintainable, machine-readable documentation

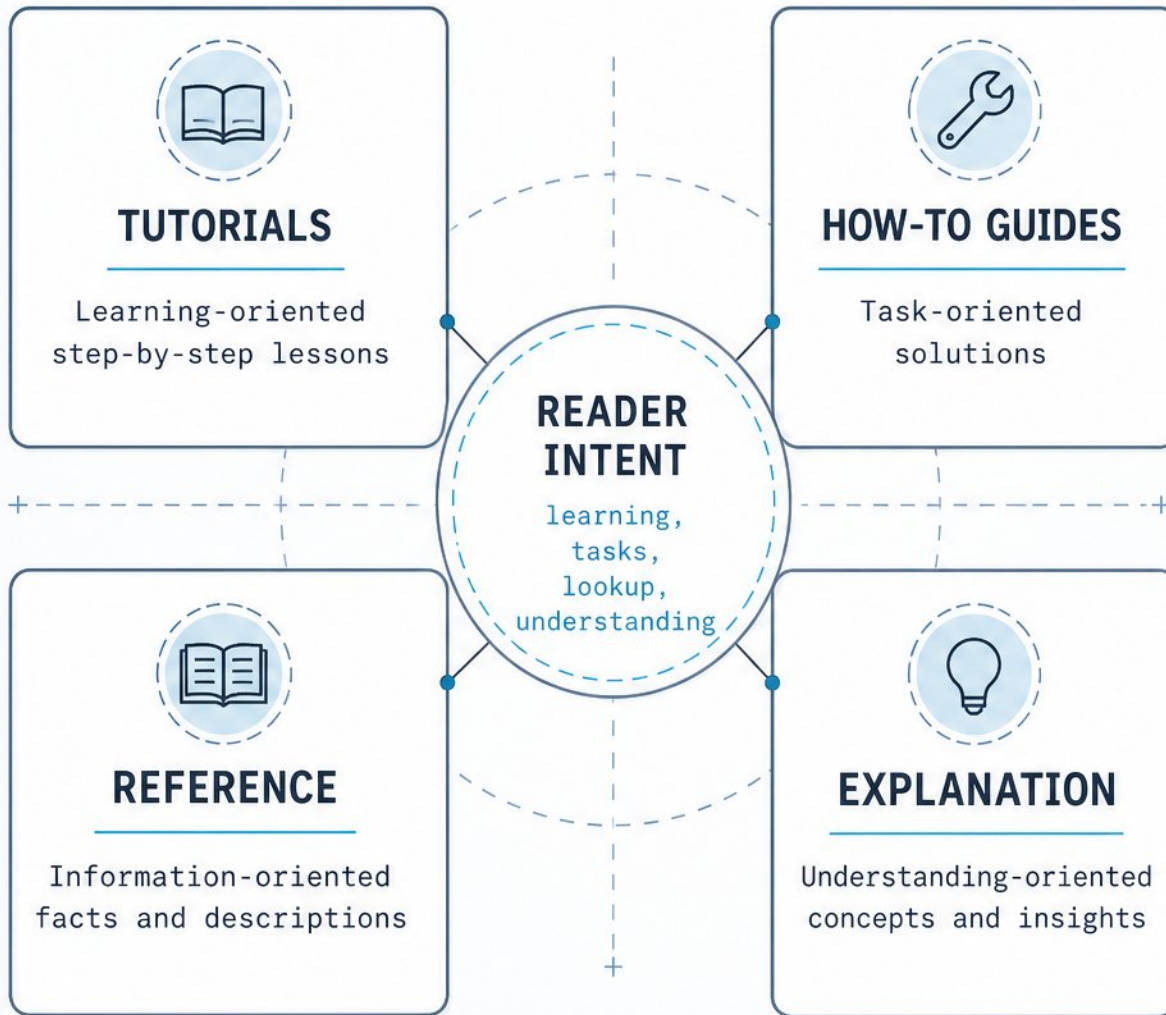


Why Documentation Quality Is a Product Problem

- AI agents and search now parse docs for answers and guidance
- Outdated content increases support load and erodes trust
- Common failure: drift between product behavior and documentation
- Top teams treat docs as part of the definition of done



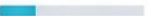
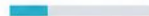
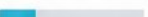
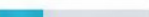
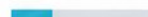
Document Types and the Diátaxis Framework



- Four pillars: tutorials, how-to guides, reference, and explanation
- Each type serves distinct reader intent: learning, tasks, lookup, understanding
- Avoid mixing content types on one page
- Additional types: release notes, runbooks, troubleshooting guides, FAQs

Audience Analysis Before Writing

- Identify primary and secondary audiences by role, expertise, and proximity
- Categories: experts, technicians, implementers, operators, executives, non-specialists
- Map audience characteristics to tone, depth, vocabulary, and examples
- Use role-based navigation or progressive disclosure for multiple audiences
- Avoid one-size-fits-all content and the curse of knowledge

	EXPERTS	TECHNICIANS	IMPLEMENTERS	OPERATORS	EXECUTIVES	NON-SPECIALISTS
						
	Deep domain knowledge	Technical specialists	Project and deployment focus	Day-to-day users	Strategic decision makers	General audience
TONE	Analytical 	Direct 	Practical 	Clear 	Strategic 	Accessible 
DEPTH	Very Deep 	Deep 	Moderate 	Surface to Moderate 	High-Level 	Surface 
VOCABULARY	Domain-Specific 	Technical 	Practical 	Plain Language 	Business 	Simple 
FORMAT & EXAMPLES	Detailed Specs, References 	Procedures, Guides 	Checklists, Templates 	Step-by-Step, How-To 	Summaries, Key Points 	Overviews, Stories 

Core Writing Principles



- Clarity, conciseness, accuracy, and task-oriented structure over flourish



- Active voice, present tense, and direct address to the reader



- Consistent terminology, acronyms defined on first use



- Avoid ambiguous pronouns and jargon



- Free buried verbs, remove empty openers, cut non-informative sentences

BEFORE

Remove empty opener
Free buried verb

~~It is important to note that the report was completed by the team in order to provide information regarding the status of the project.~~

Cut non-informative phrase
Avoid unnecessary words
Remove wordiness

AFTER

The team completed the report and provides the project status.

Active voice
Direct address to reader

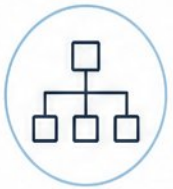
Information Architecture and Readability



- Organize navigation around user tasks and goals, not internal features



- Structure pages for scanning: descriptive headings, short sections, lists, tables, and callouts



- Keep top-level navigation shallow with 5–9 categories; use progressive disclosure for deep hierarchies



- Ensure each page has a single purpose and standalone context with explicit links



Navigation depth

Shallow top-level navigation (5–9 categories)

Descriptive headings

Clear, descriptive headings communicate page purpose

Scannable sections

Short sections with lists help users scan and find information

Structured content

Tables organize information for quick comparison

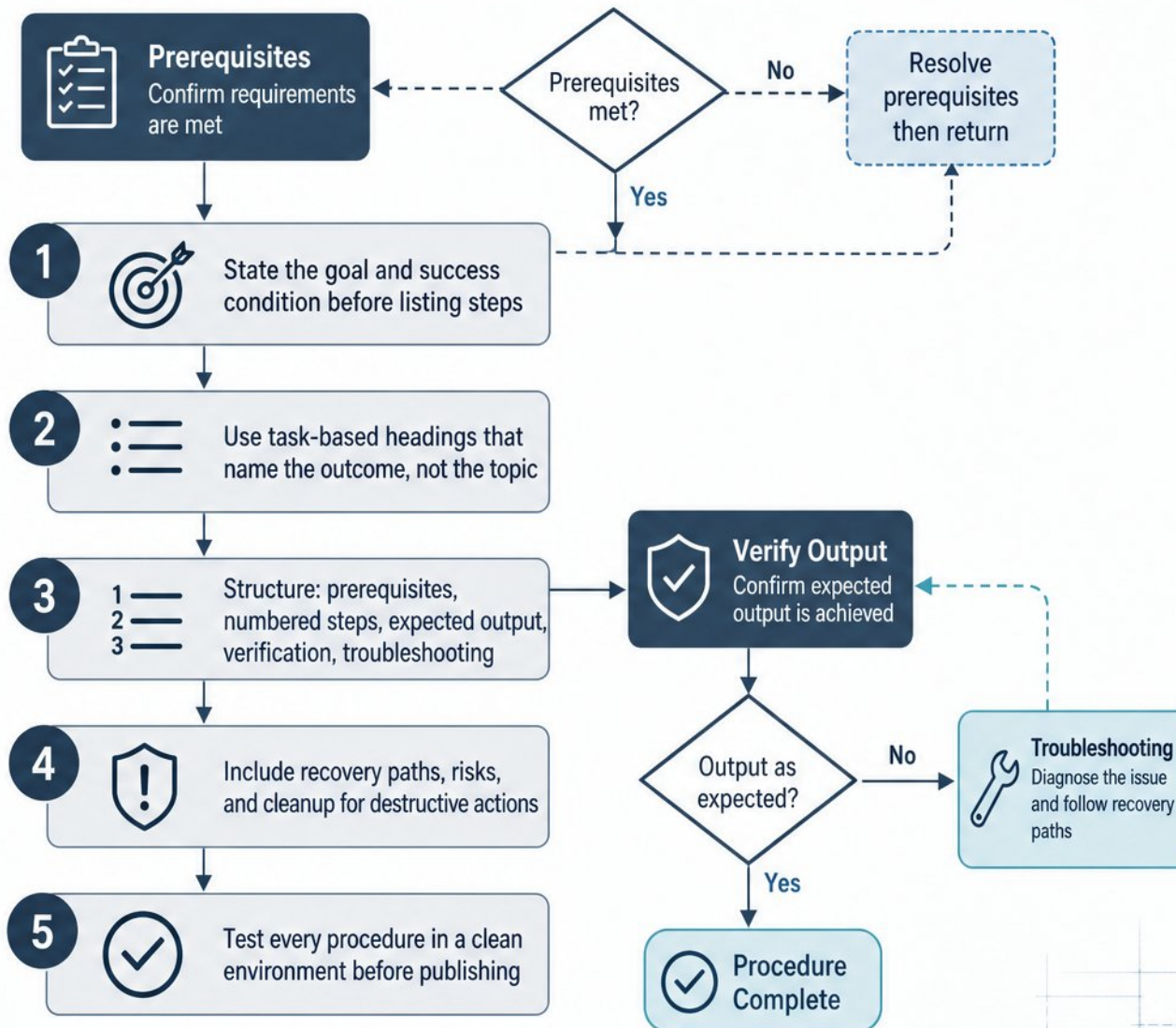
Callouts

Callouts highlight key notes and important context

Explicit links

Clear links provide standalone context and next steps

Writing Procedures and Task-Based Content



- Structure: prerequisites, numbered steps, expected output, verification, troubleshooting
- State the goal and success condition before listing steps
- Use task-based headings that name the outcome, not the topic
- Include recovery paths, risks, and cleanup for destructive actions
- Test every procedure in a clean environment before publishing

API Reference Documentation

- - Consistent endpoint pages: purpose, method, parameters, request, response
- - Document authentication, rate limits, defaults, and edge cases separately
- - Generate reference from OpenAPI to prevent drift from implementation
- - Provide runnable multi-language examples with realistic data



Style Guides, Templates, and Consistency



- Build lightweight style guides covering voice, terminology, and formatting



- Create reusable page templates for tutorials, how-tos, reference, and troubleshooting



- Enforce consistency with linters like Vale and Spectral in CI pipelines



- Balance consistency with readability through intentional, documented exceptions

CONFORMS TO STYLE GUIDE

How to Reset Your Password

Follow these steps to reset your password for your account.

Prerequisites

You need access to the email address associated with your account.

Steps

1. Go to the sign-in page.
2. Select Forgot password?.
3. Enter your email address.
4. Check your email for a reset link.
5. Click the link and choose a new password.

Notes

If you don't receive the email, check your spam folder.

Consistent headings,
tone, and formatting

DEVIATES FROM STYLE GUIDE

HOW TO reset *your* Password!!!

Here's how you can quickly reset your password.

Informal
tone

Requirements

Must have access to the e-mail used for your account.

Terminology
inconsistent

STEPS:

- Open the sign in page
- Click "Forgot Password"
- Type your email
- We'll send a link
- Use the link to set a new password :)

Formatting
differs

Note:

No email? Maybe it's in spam...

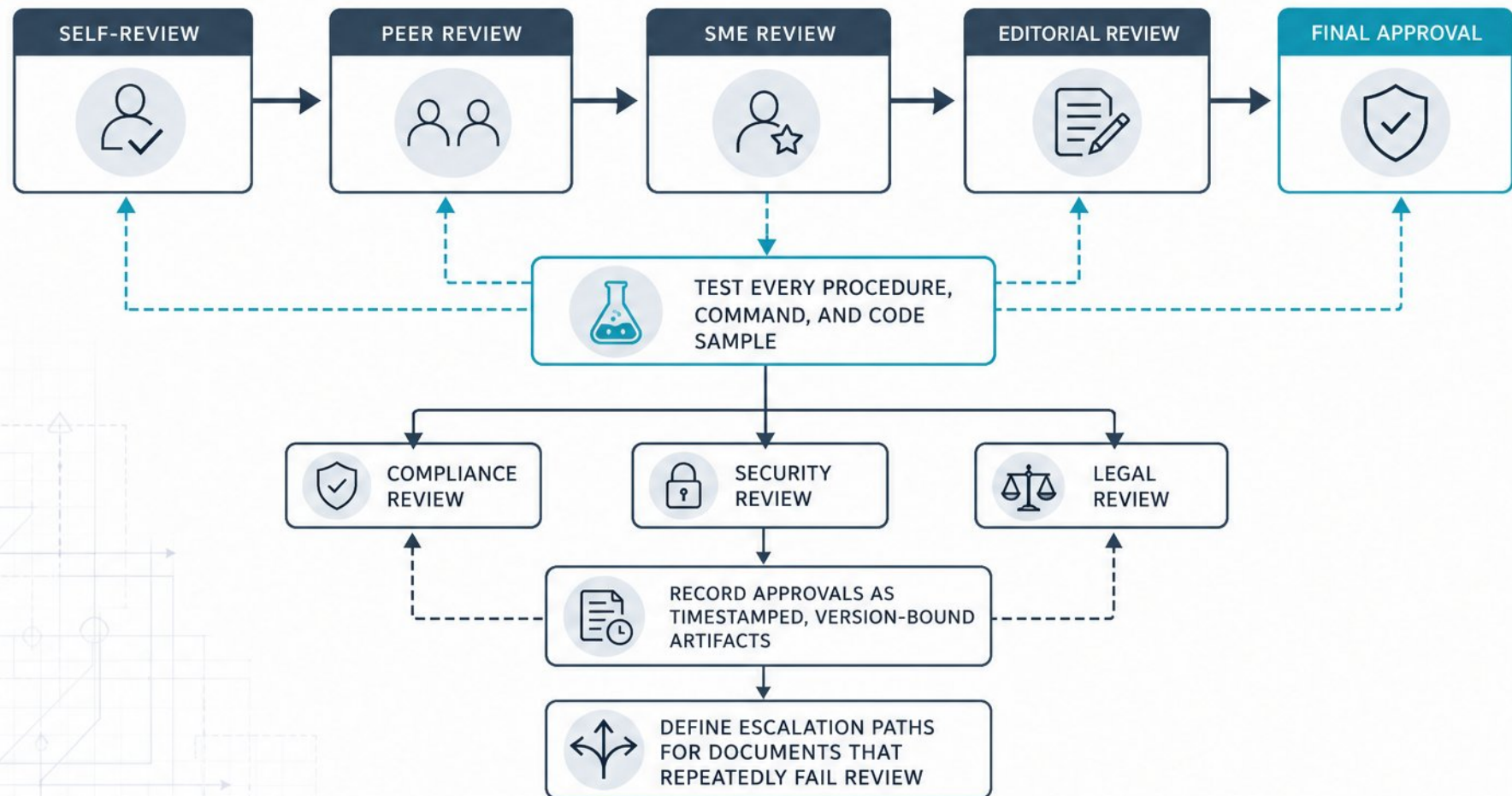
Punctuation
differs

LINTER FINDINGS

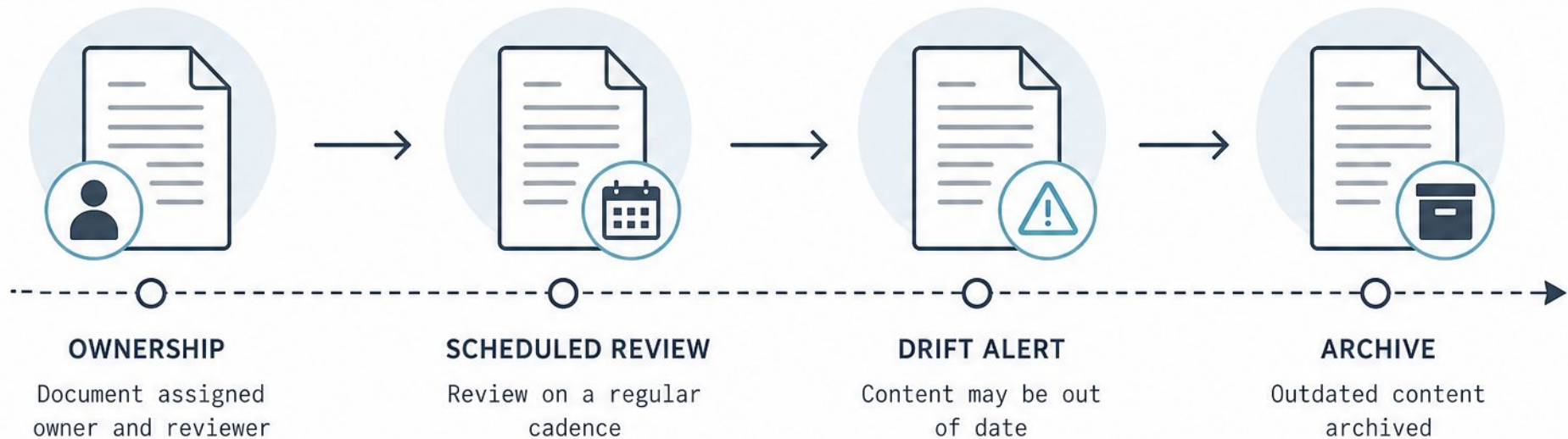
- ✗ Avoid exclamation marks in headings.
- ✗ Use consistent terminology: email (not e-mail).
- ✗ Use sentence case for headings.

Review, Testing, and Approval Workflows

- - Staged review: self, peer, SME, editorial, and final approval
- - Test every procedure, command, and code sample in a clean environment
- - Route risk-sensitive content through compliance, security, or legal review
- - Record approvals as timestamped, version-bound artifacts
- - Define escalation paths for documents that repeatedly fail review

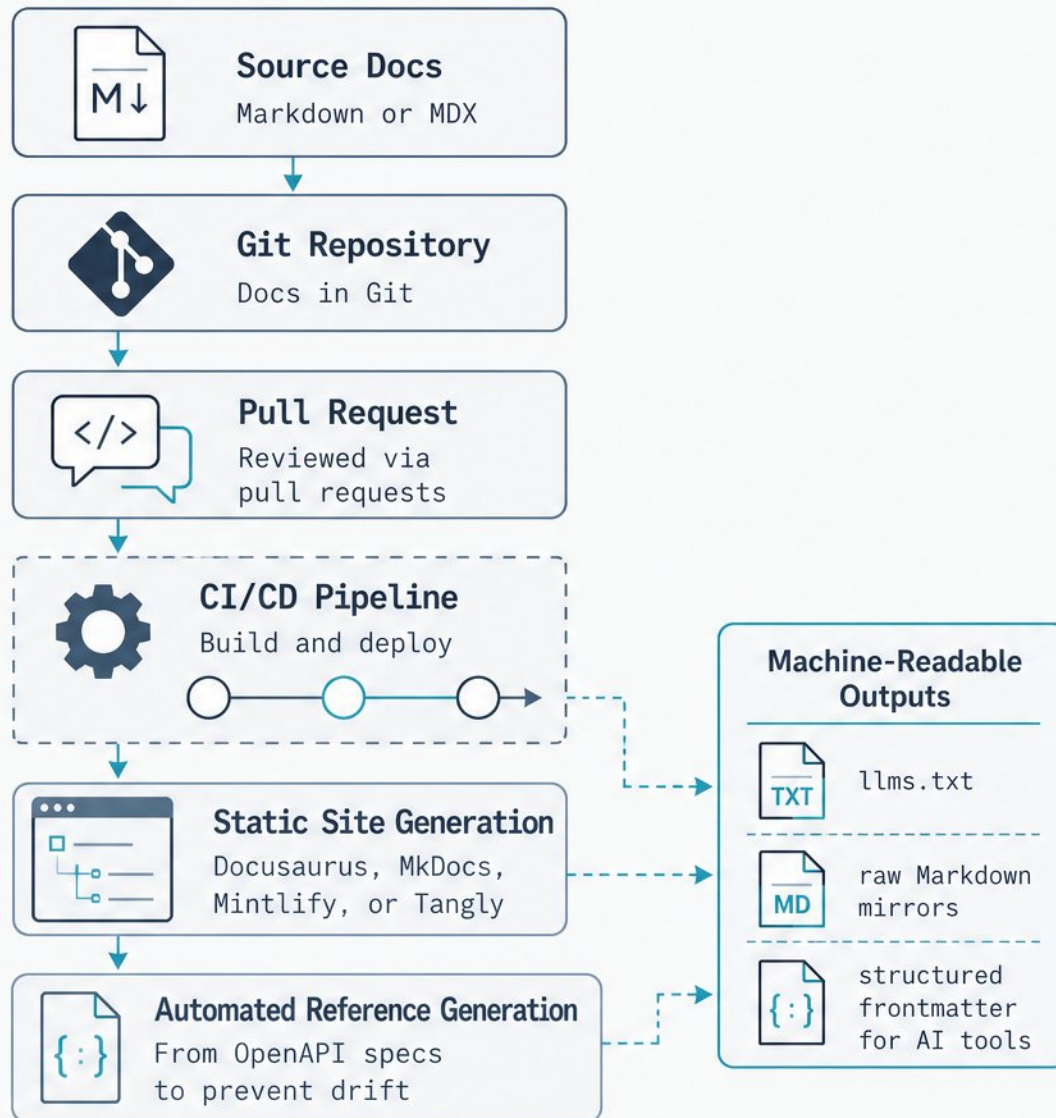


Maintenance, Ownership, and Lifecycle



- Every document needs a named owner and reviewer
- Define update triggers tied to releases and incidents
- Review high-traffic pages more frequently than stable ones
- Archive outdated content instead of leaving conflicting pages
- Use metrics and release checklists to track documentation health

Docs as Code and Tooling



- Treat docs like source code: Markdown or MDX in Git, reviewed via pull requests
- Deploy through CI/CD with static site generators such as Docusaurus, MkDocs, Mintlify, or Tangly
- Automate reference generation from OpenAPI specs to prevent drift
- Add machine-readable outputs like llms.txt, raw Markdown mirrors, and structured frontmatter for AI tools

Practical Strategies for Product Teams



- Engineering owns how the system works; product owns what it does for users



- Assign a named owner and reviewer to every important document



- Embed docs tasks in PRs, release checklists, and launch planning



- Track docs drift, repeated tickets, time-to-first-call, and AI answer accuracy



- Use templates and AI-assisted drafting; humans retain accuracy responsibility

Doc Task	 Engineering	 Product	 Support	 Technical Writing
Define and scope what to document				
Create or update documentation				
Review for accuracy and completeness				
Publish and keep docs discoverable				
Monitor usage and improve content				



Responsible
Does the work



Accountable
Owns the outcome

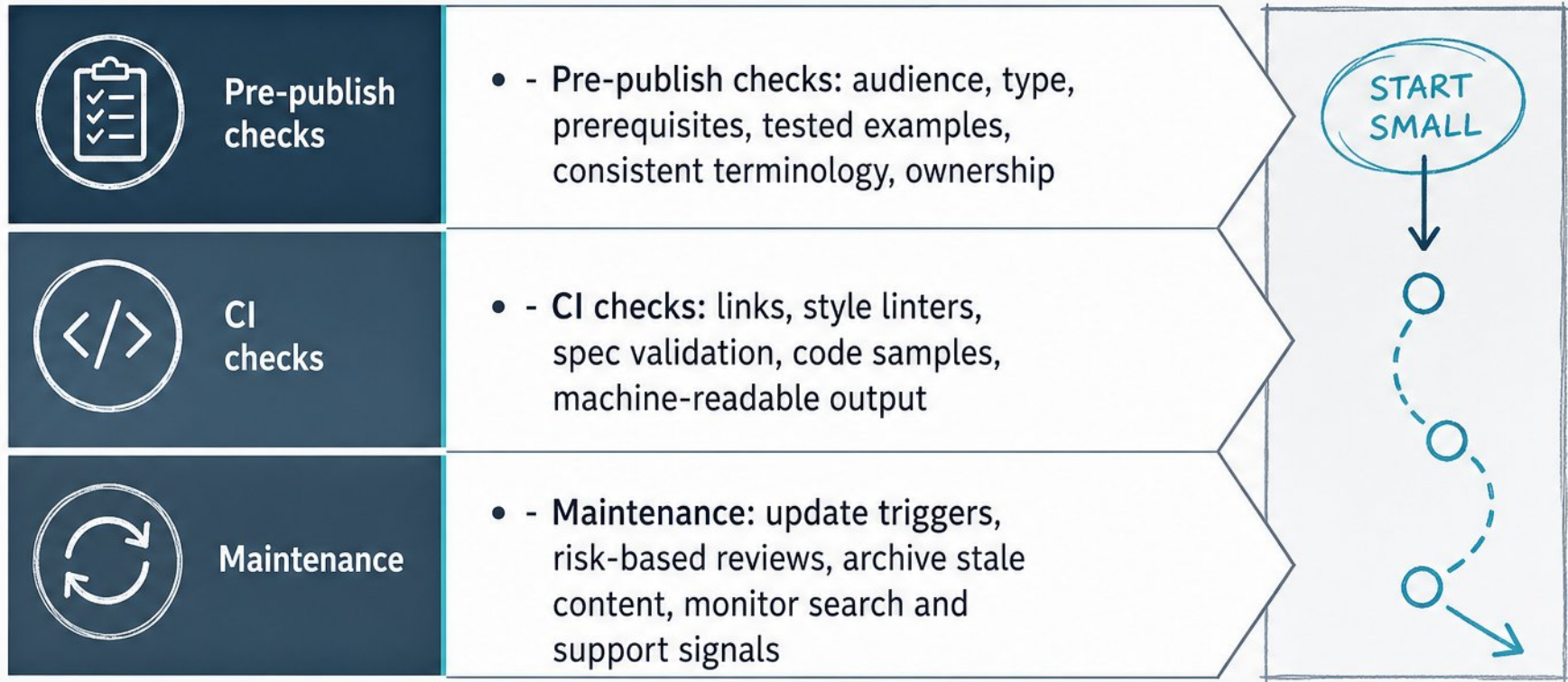


Consulted
Provides input



Informed
Kept in the loop

Checklist and 30-Day Action Plan



Good docs
are built,
not born.



- **Start small:** audit current docs, assign owners, create templates, set review gates, focus on one high-impact area



Iterate.
Improve.
Repeat.

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/9d8a411a/public