

Introduction to the Technical Writing Process: Stages, Roles, and Deliverables

- ▶ Documentation moves from idea to published asset, staying current
- ▶ Core themes: structured stages, clear role ownership, deliverables
- ▶ Shared workflow avoids delays, rework, and stale content
- ▶ Actionable templates, checklists, and handoff criteria for sprints
- ▶ Defined process turns docs from bottleneck to product enabler



The Business Case for a Structured Documentation Workflow



- Unclear processes cause delayed releases, repeated tickets, and slow onboarding



- Structured stages cut review cycles and prevent documentation debt



- Documentation is a product requirement that drives adoption and trust



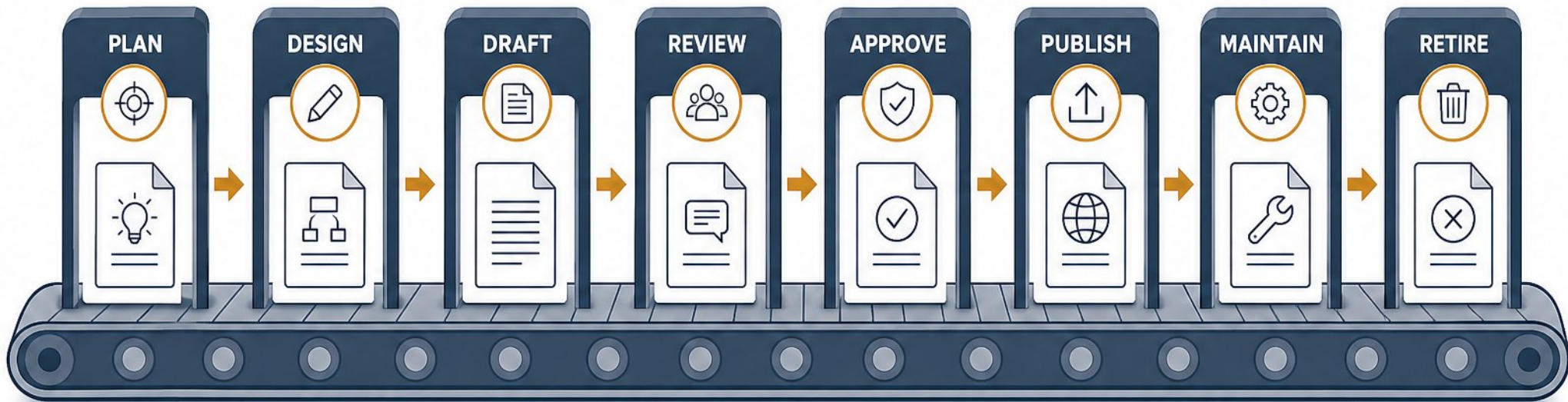
- Success criteria: clarity, consistency, verifiability, maintainability, task success



- Quantified impact: ticket deflection and onboarding time reduction



Core Stages of the Document Development Life Cycle (DDLC)



- Repeatable pipeline: plan, design, draft, review, approve, publish, maintain, retire
- Maintenance and retirement are formal stages, not afterthoughts
- Lightweight vs. regulated workflows: more gates for high-risk docs
- Map DDLC stages to product development phases
- Each stage is a gate: advance only when safe

Planning: From Audience Need to Documentation Scope



- Identify users, tasks, and decisions documentation must support



- Translate product scope into documentation scope, including explicit non-goals



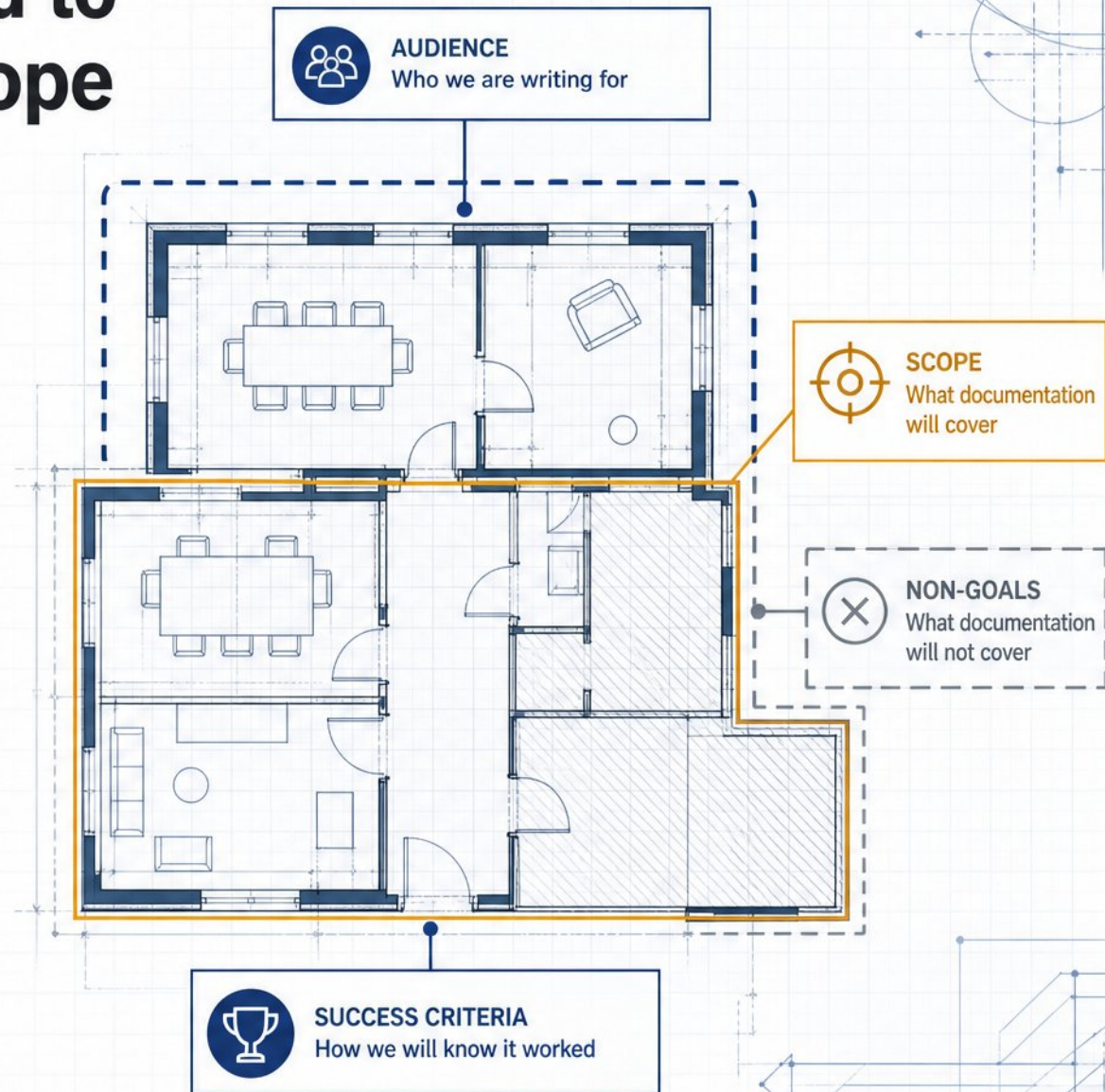
- Define measurable success criteria: task completion, ticket deflection, time-to-first-success



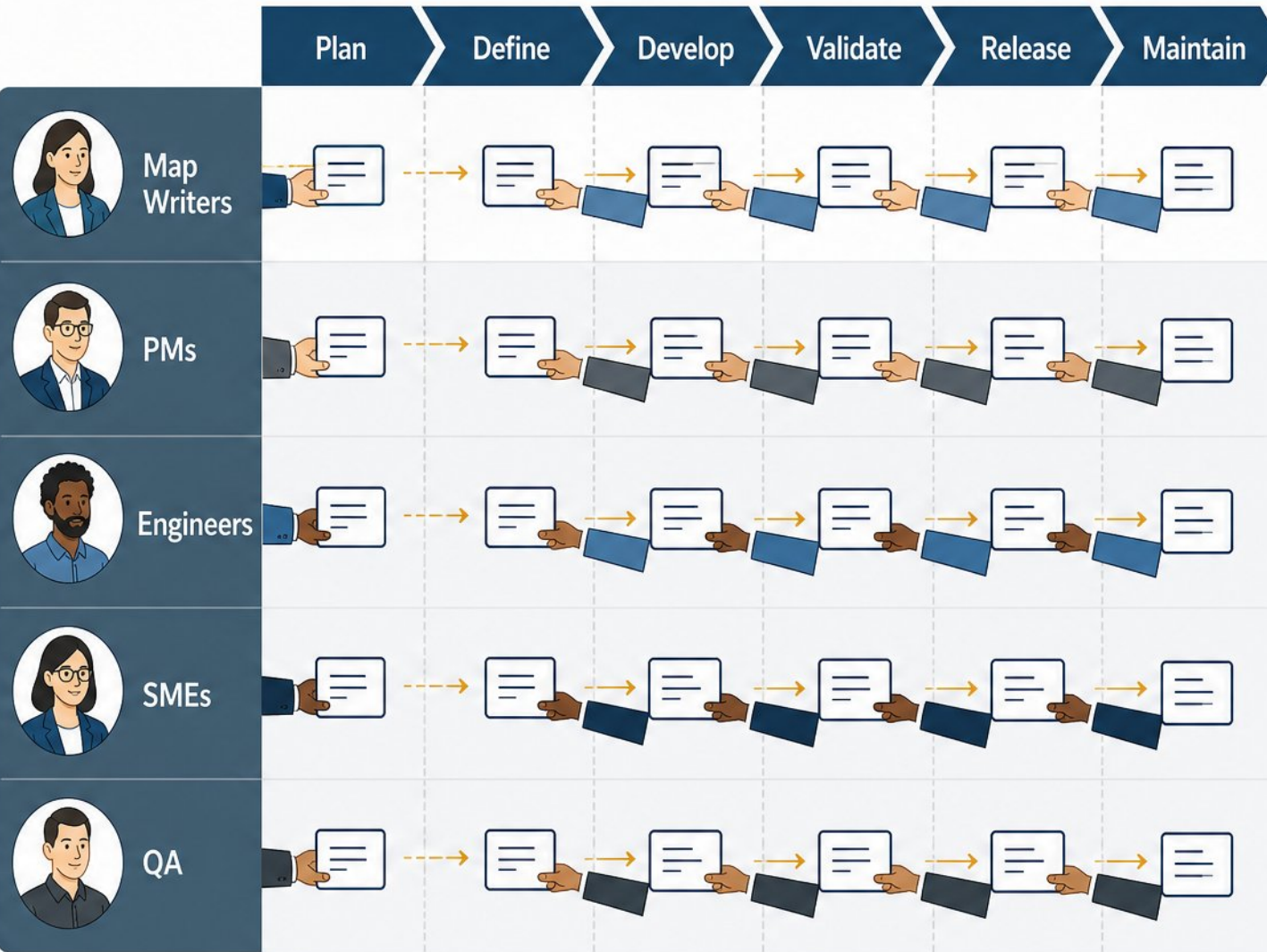
- Create a lightweight documentation plan for quick cross-functional review and approval



- Use support tickets and search gaps as planning signals for what matters most



Roles and Responsibilities: Who Does What, When



- Map writers, PMs, engineers, SMEs, and QA to DDLC stages



- Use RACI: one Accountable owner per document and stage



- Avoid vague ownership, SME bottlenecks, unclear approvals



- Name an owner for every stage and document



- Use scoped SME requests and predefined escalation paths

Handoffs and Reviews: The Artifacts That Move Work Forward



- Stage completion signaled by drafts, sign-offs, and baselines



- Handoff artifacts reduce ambiguity and prevent missed feedback



- Name a reconciliation authority for conflicting feedback



- Checklists let the receiving role proceed without meetings



- Handoff completes only when the next owner has a clear action



Deliverables at Each Stage: From Plans to Published Docs

-  Define outputs per stage: audience profiles, plans, drafts, review packages, release notes, and maintenance logs.
-  Distinguish source docs (version control) from published deliverables.
-  Vary deliverables: user guides, API references, release notes, and troubleshooting content.
-  Map style guides, knowledge bases, and version control conventions.
-  Scale: lightweight versus formal deliverables based on content risk.



Review Workflows: Separating Accuracy, Clarity, and Compliance



- Separate passes: self, peer, SME, editorial, compliance



- Verify technical accuracy before polishing language



- Send scoped review requests with specific questions



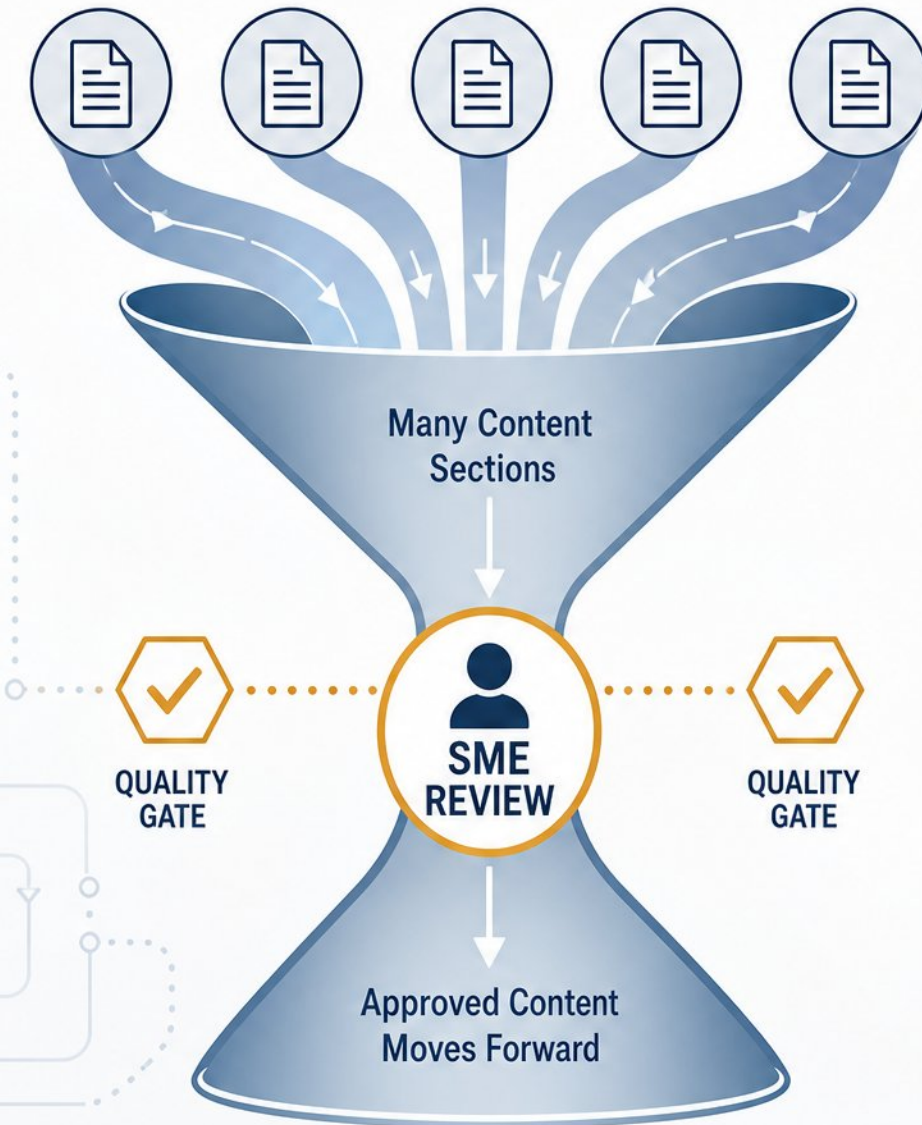
- Categorize feedback: must-fix, should-fix, nice-to-have



- Capture approvals as timestamped records tied to version



Managing the SME Review Bottleneck



- SME review is where documentation workflows stall most often



- Label content gaps as TBD-SME with a named owner before review



- Send SMEs only the sections they need to verify, with clear questions



- Tier review depth: small updates skip it; new articles require full validation



- Establish escalation paths for missed deadlines and conflicting feedback

Publication, Versioning, and Release Readiness

- ▶ Confirm all comments resolved and accuracy verified before publishing
- ▶ Keep URLs stable and label lifecycle states clearly
- ▶ Treat each supported version as a distinct contract
- ▶ Align versioning with breaking, feature, and fix changes
- ▶ Verify published docs against production behavior



Maintenance, Deprecation, and Retirement as First-Class Stages



- Define update triggers: feature changes, defects, support trends, and deprecations.



- Assign a named owner to every page; schedule reviews based on traffic and risk.



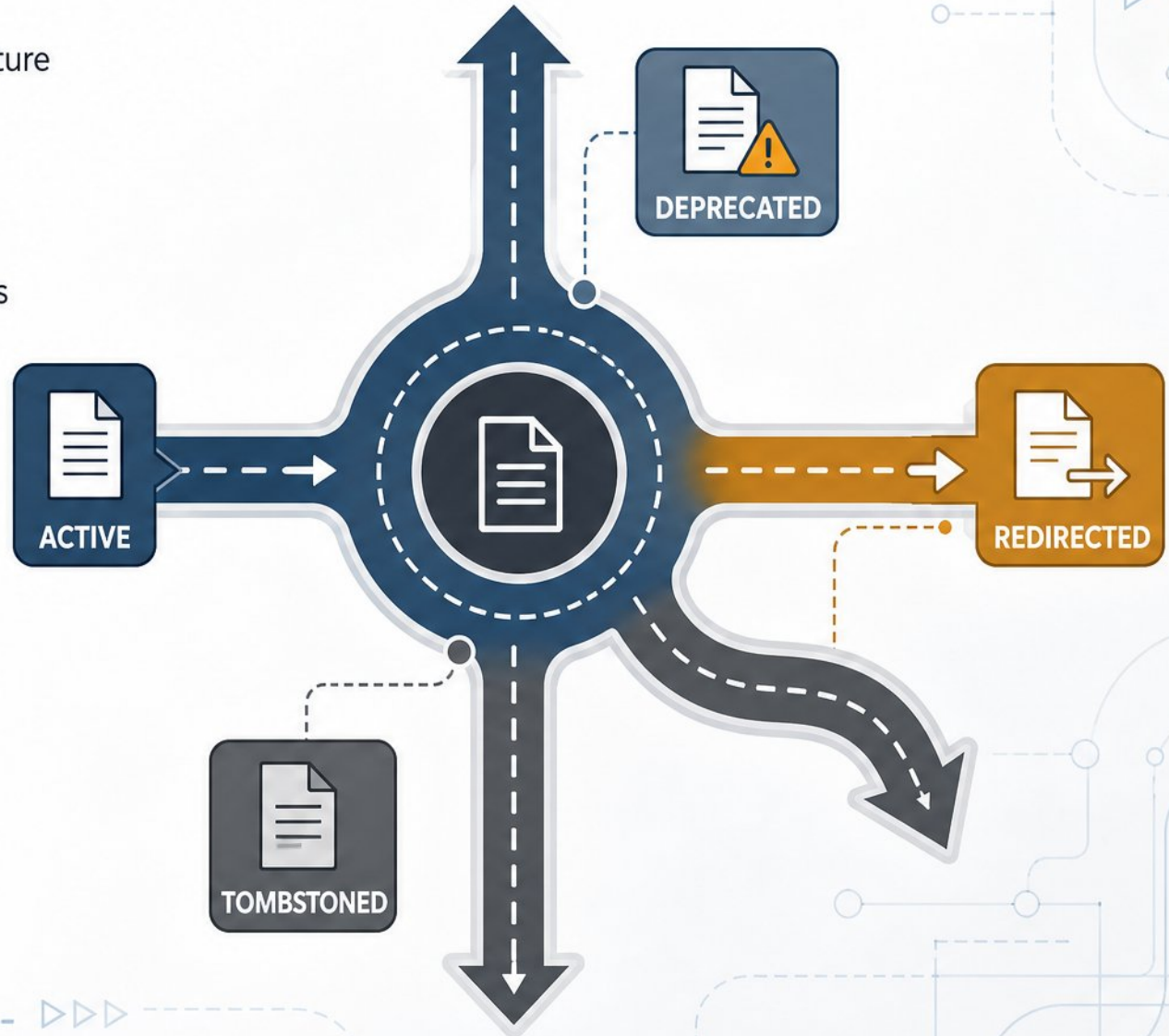
- Create deprecation notices and migration guides long before retirement.



- Use tombstone pages and intentional redirects instead of deleting content.



- Treat retirement as a safety mechanism that prevents conflicting documentation.



Aligning Documentation with Agile and Product Team Workflows



- Embed documentation tasks into sprint planning and the Definition of Done



- Add a documentation impact field to tickets: None / Update / New article



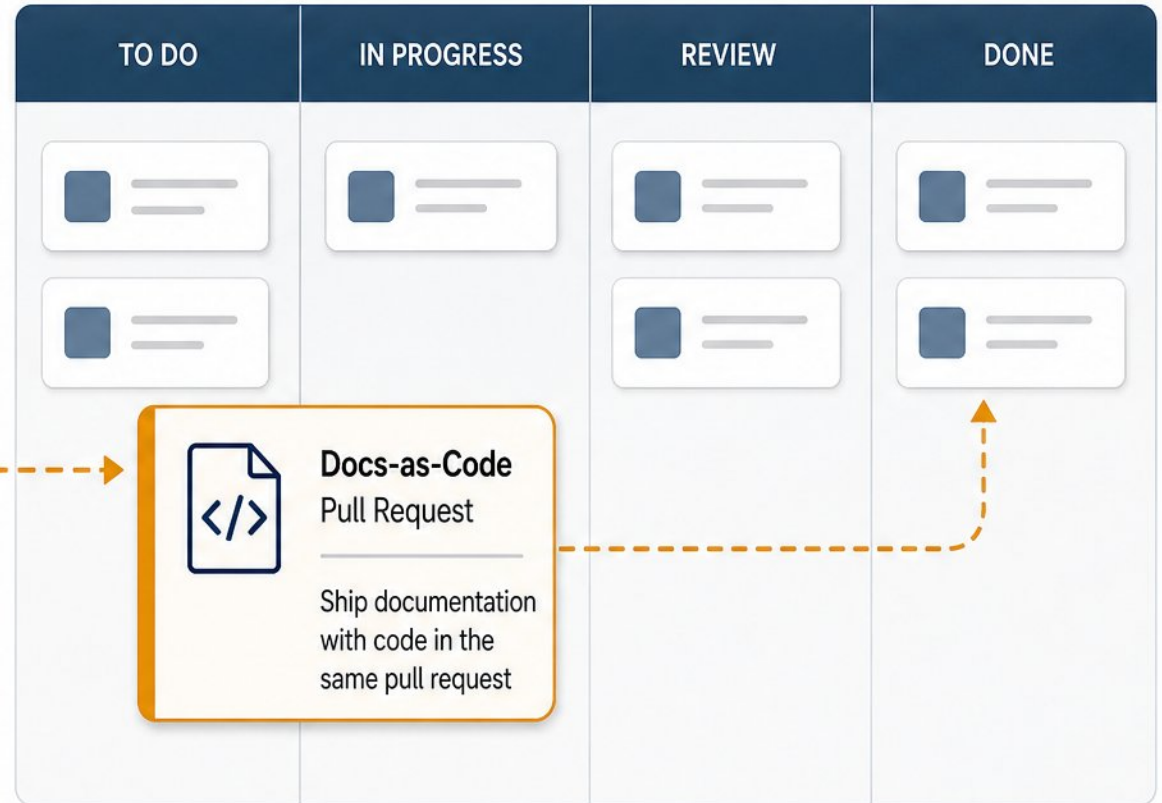
- Adopt docs-as-code: ship docs with code in the same pull request



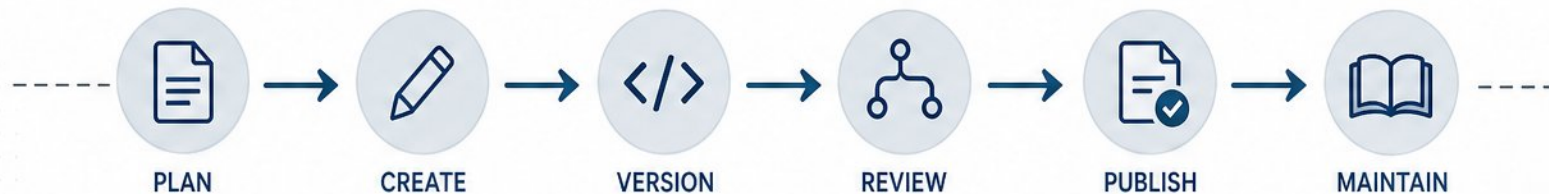
- Match documentation depth to increments, betas, and stable releases



- Prevent docs from becoming a separate track that misses product changes

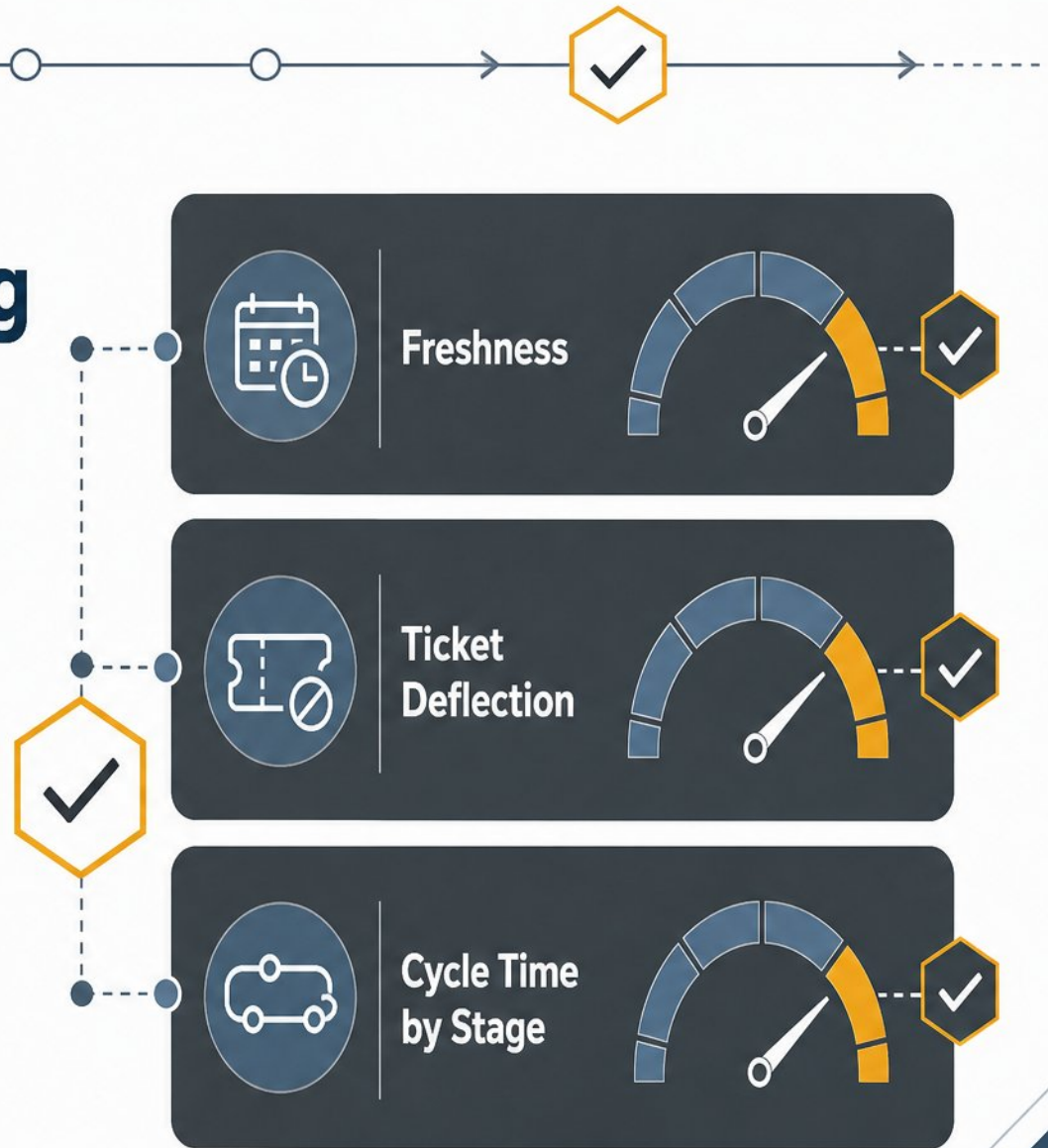


DOCUMENT LIFECYCLE ASSEMBLY LINE



Metrics That Prove the Process Is Working

- Track a small, balanced set across quality, usability, and efficiency
- Prioritize outcome metrics: ticket deflection, time-to-first-success, freshness
- Use cycle time by stage to find bottlenecks in drafting, review, or publication
- Monitor escaped defects and content drift to spot weak review coverage



Turning the Framework into Team Practice



- Build a simple team playbook with stage gates and templates



- Pick a pilot project with visible friction to test the process



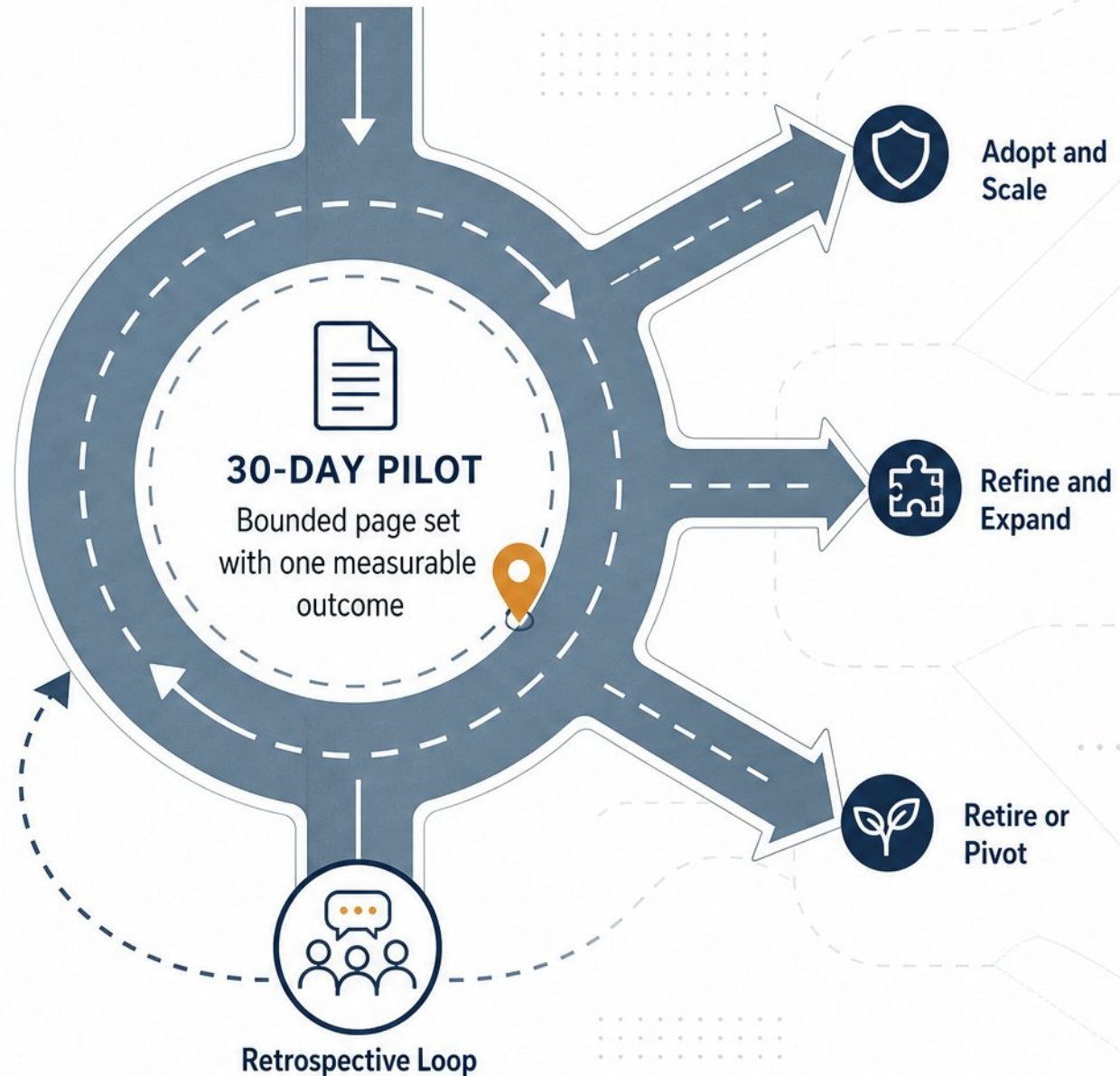
- Run a 30-day pilot with a bounded page set and one measurable outcome



- Use retrospectives and quarterly reviews for continuous improvement



- Keep the process lightweight under release pressure, structured enough to prevent drift



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/83006fbe/public