

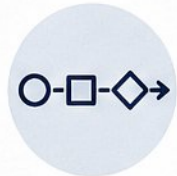
Technical Writing Fundamentals



- Reader-centered communication for completing tasks and making decisions



- Focuses on clarity, accuracy, usability—unlike marketing or academic writing



- Course arc: audience, clarity, structure, procedures, visuals, APIs, style, review



- Serves writers, product teams, developers, and educators



What Makes Technical Writing Different



- Technical communication spans text, visuals, video, and interactive help



- Procedural writing gives steps; scientific writing persuades with data



- Workflow: plan, draft, review, test, and maintain



- Information products are built, tested, and maintained like the product itself

BEFORE

PROCEDURE GUIDE



AFTER

WHITE PAPER



PLAN



DRAFT



REVIEW



TEST



MAINTAIN

Audience Analysis: The First Step



- Define audience roles using the expert-technician-executive-non-specialist framework



- Use reader goals and current knowledge to decide what to include or omit



- Apply proximity and 'curse of knowledge' checks to avoid assuming internal context



- Use a profile worksheet: role, knowledge, goals, context, and likely questions



“



ROLE



KNOWLEDGE LEVEL



GOALS



CONTEXT



LIKELY QUESTIONS

Audience Fit: Tone, Terminology, and Depth



- Match vocabulary and detail to reader expertise; never talk down



- Layer content for mixed audiences with clear scope statements



- Avoid idioms and cultural references for global readers

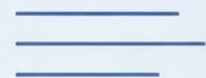


- Use summaries and prerequisites to help readers self-select



SUMMARY FOR EXECUTIVES

High-level overview of goals, key ideas, and implications.



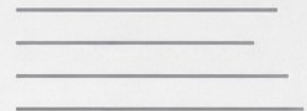
DETAIL FOR TECHNICIANS

In-depth explanations, processes, and technical considerations.



SCOPE STATEMENT FOR NON-SPECIALISTS

Plain-language context, purpose, and boundaries.



Principles of Clear and Concise Writing



- Use active voice and present tense as default



- Prefer short sentences with precise, concrete verbs



- Avoid jargon, hidden verbs, and unnecessary words



- Clarify complex ideas without oversimplifying content



- Revise wordy drafts with before-and-after examples

BEFORE

It is recommended that the user undertake the configuration updates subsequent to the completion of the deployment process in order to enable optimal system performance.



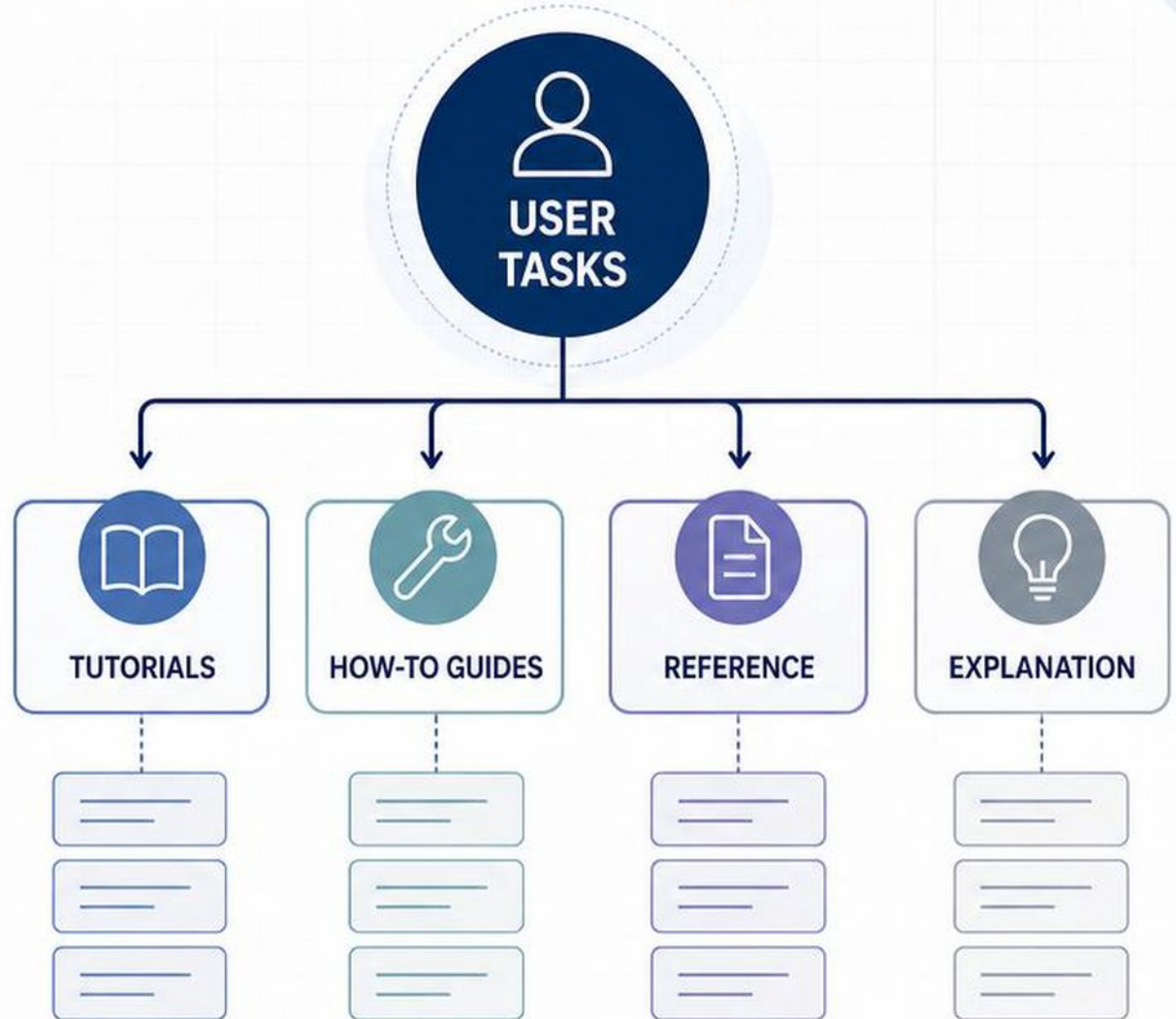
AFTER



After you finish deploying, update the configuration to optimize system performance.

Information Architecture: Structuring for Findability

- Match content to user goals: tutorials, how-tos, reference, explanation
- Organize by tasks and workflows, not internal product structure
- Use consistent headings, page groups, and navigation for scanning
- Apply templates and content types to keep docs predictable



Writing Procedures and Step-by-Step Instructions

1



- Start each step with an imperative verb (e.g., Click, Select)



2



- State location before action:
"In Settings, select Accounts"



3



- Keep one action per step;
split if it contains "and"



4



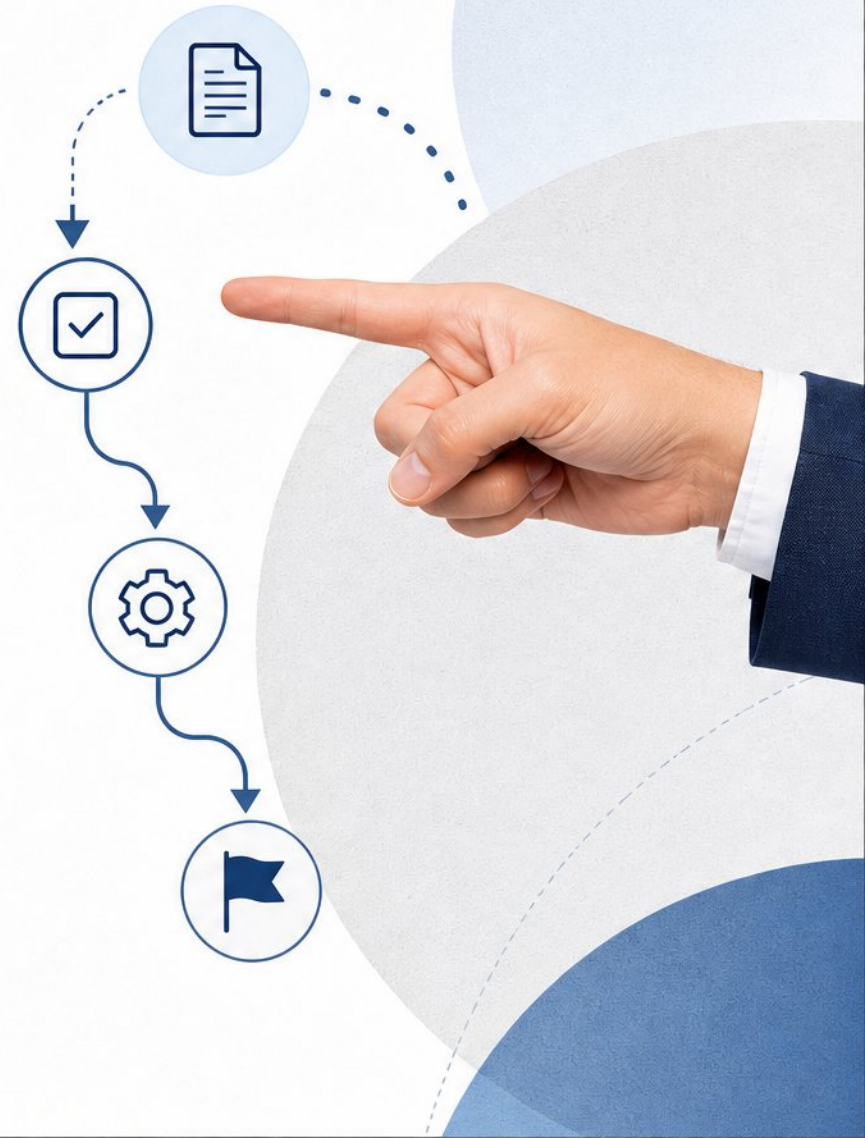
- Place warnings, conditions, and
expected results before the action



5



- Test with novices; revise for
stalls, backtracking, and wrong turns



Visuals in Technical Content



- Use visuals to clarify complex workflows or concepts



- Introduce each image with a complete sentence



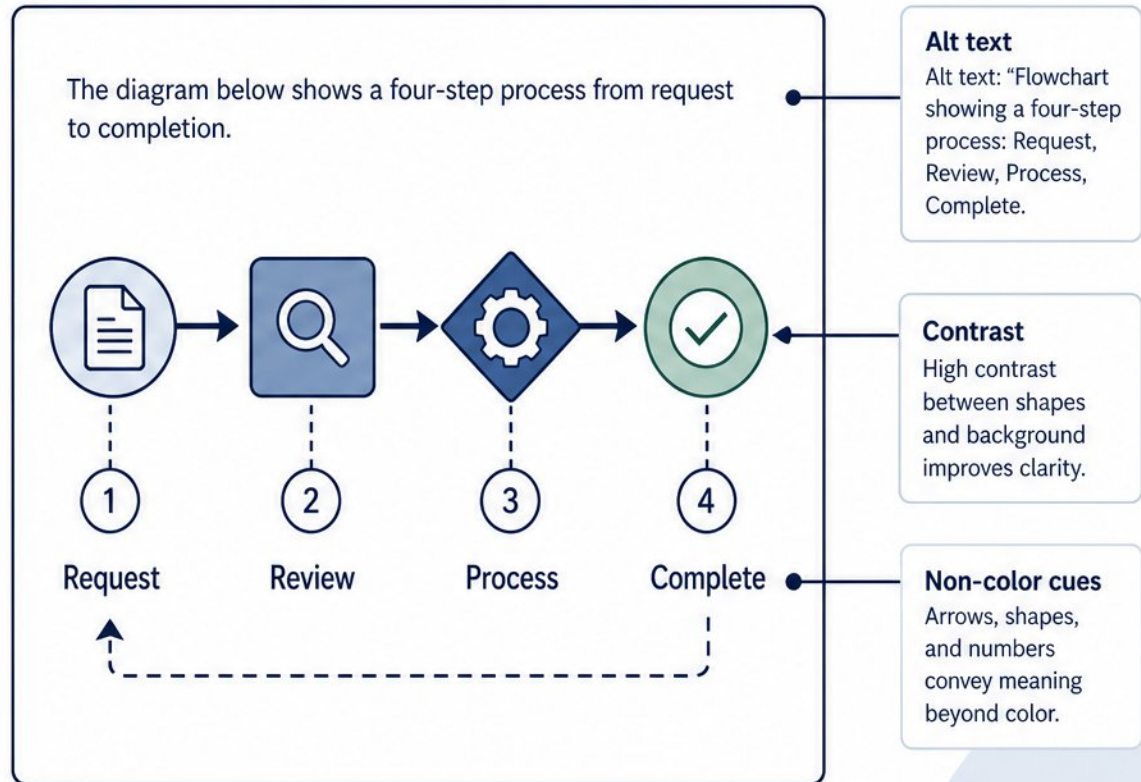
- Write concise, contextual alt text for every image



- Never place new information only inside an image



- Ensure sufficient contrast and non-color cues



Documenting APIs and Developer-Facing Products



- Structure docs around five essentials: quickstart, auth, reference, errors, task guides



- Write parameter tables, request/response examples, and actionable error descriptions



- Provide tested, runnable code examples in multiple languages



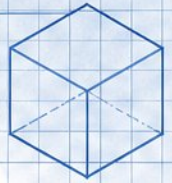
- Use docs-as-code and OpenAPI workflows to reduce drift



- Default to latest version; flag breaking changes with migration guides

GET /v1/items

Retrieve a list of items.



Parameters

Name	In	Type	Required	Description
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

Request example

```
{
  "query": "search term",
  "limit": 20,
  "sort": "created_at"
}
```

Response example

```
{
  "items": [
    {
      "id": "item_abc",
      "name": "Example Item"
    }
  ]
}
```

Errors





Style Guides and Terminology Management

- Start from Google, Microsoft, or Apple style guides
- Adapt few team-specific rules for voice and terms
- Maintain a glossary for consistent terminology
- Use rules to cut review friction across contributors



STYLE RULE: VOICE & FORMATTING

- ✓ Use clear, concise, and user-centered language.
- ✓ Write in active voice.
- ✓ Use consistent capitalization and punctuation.
- ✓ Prefer everyday words over jargon.

Reviewing, Editing, and Gathering Feedback

- Separate peer, SME, editorial, and stakeholder review stages
- Use checklists: clarity, accuracy, consistency, completeness, usability
- Sort feedback into must-fix, should-fix, and nice-to-have
- Close the loop: confirm changes and improve future drafts



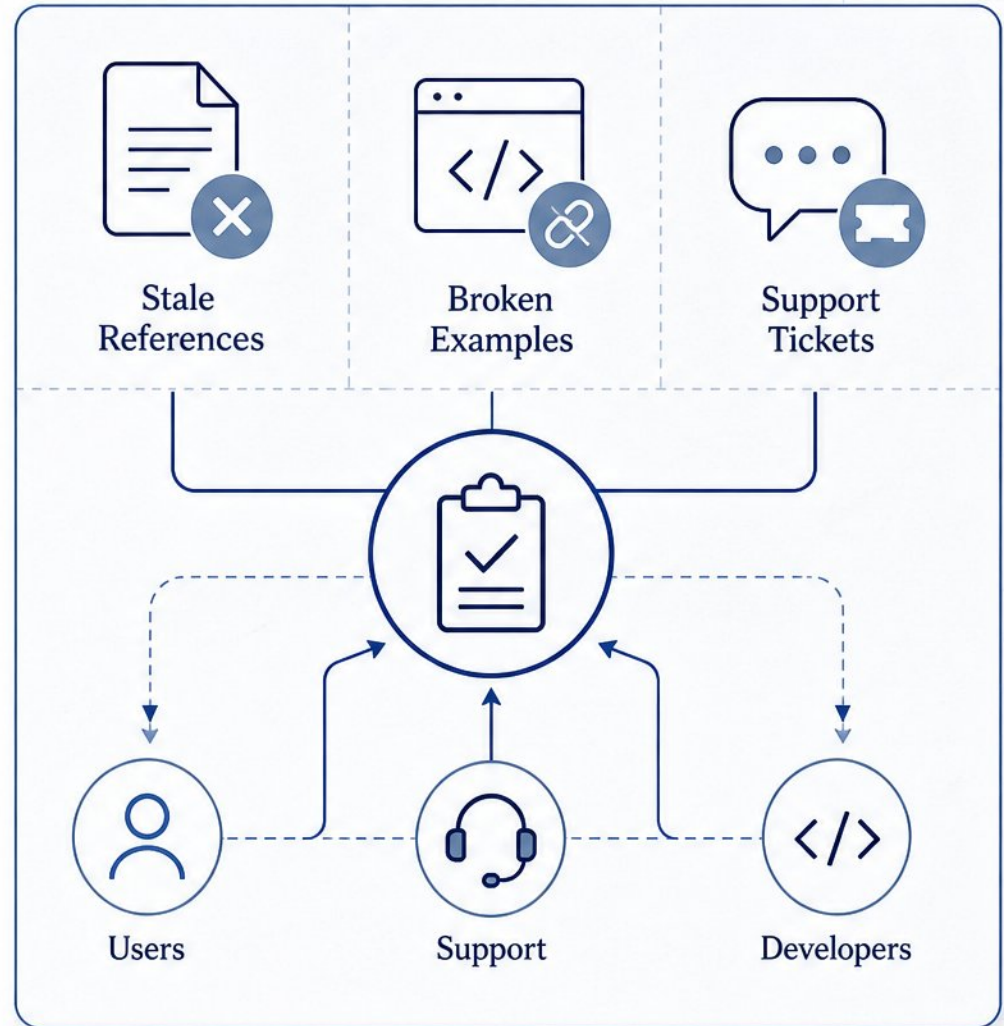
Documentation Workflows and Tooling



- - Five-stage workflow: plan, draft, review, publish, maintain
- - Pick tools that support collaboration, version control, and simple publishing
- - Use docs-as-code to let developers contribute alongside code changes
- - AI-assisted drafting saves time; keep human verification for accuracy

Maintaining Quality Over Time

- – Tie reviews to release cycles and automated checks
- – Track stale references, broken examples, and support tickets
- – Build lightweight practices that scale with doc growth
- – Create feedback loops from users, support, and developers



Applying Technical Writing in Your Role



- Plan a small documentation project from outline to publication



- Identify the highest-leverage improvement for your writing context



- Turn templates, checklists, and review stages into repeatable habits



- Consult style guides, communities, and practice resources for growth



- Start small: ship one doc end to end



NEXT DOC CHECKLIST

- ☐ _____
- ☐ _____
- ☐ _____
- ☐ _____
- ☐ _____

TEMPLATE



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/6900312b/public