
The Good Docs Project

Template roadmap agenda and notes - 1.0
Adopted May 2023



Template roadmap

WHAT'S IN THIS DOCUMENT?	1
WHY DO WE NEED A ROADMAP?	1
TEMPLATE ROADMAP	2
Our personas: who are our templates for?	2
Templates for the core documentation pack	3
Templates for the OSS community docs pack	3
Future template packs	4
Expanding beyond templates to other types of content	8
Template deliverable changes	8
RESOURCES	11

WHAT'S IN THIS DOCUMENT?

This document outlines the plan for The Good Docs Project's future template roadmap and long-term product vision for the next 1-2 years of work. This plan was created in a series of five meetings with the template leads, including Cameron Shorter, Deanna Thompson, Joanne Fung, Gayathri Krishnaswamy, and Alyssa Rock.

IMPORTANT: This plan is merely a draft and it's open for comments and suggestions from any member of the project, especially members of the template working groups. Feel free to leave a comment on the Google Draft!

WHY DO WE NEED A ROADMAP?

We've got a good plan in place for the templates for our current release (Capilano), but we need to begin thinking about our future release (Dragon). Our project is taking on new templateers all the time and they need work to do. We need a roadmap and vision of where the templates

project is going so that we can build out our backlog of future work and to ensure our backlog includes the work we actually want templateers to do.

As this document currently stands, it represents our best assessment of what would be most helpful to our users. Our plan will inevitably need to be revised and perfected iteratively over time. As the project matures and hopefully begins to attract more users, we'll adjust this plan based on their feedback and what appears to be most helpful to our target audience.

TEMPLATE ROADMAP

Our personas: who are our templates for?

Who are the core personas we're targeting? Let's define our primary and secondary personas. Who might use our templates?

Decision: We agreed that Developer-Writer Darby is our core persona. We're creating templates that will be used by developers with varying degrees of writing ability and confidence in their writing abilities.

A secondary persona is Doc System Owner Olly. Later as a documentation project matures, the project might bring on a technical writer who might actually be the entry point for Darby to use our templates. Olly might download our templates for use by the developers who will create documentation on their team in a docs-as-code system. See: [The Good Docs Project Personas](#), specifically slides 4-12.

What is a template pack?

A template pack is a collection of templates organized together by:

- Common use cases or tasks
- Needs of particular user personas
- Popular or interesting documentation frameworks
- [Maturity models](#)
- Future criteria or needs based on user research and feedback

Templates for the core documentation pack

Which criteria should we use to decide which templates will go into the core documentation pack?

Decision: The core documentation pack will be our flagship template pack and it will include the core, fundamental content types that every documentation project needs. If you download one template pack for your project, it should be this one.

Note that this is slightly different from a “minimal viable documentation pack” for starter documentation projects, which is a future template pack with a slightly different focus.

The criteria we use for determining whether a template goes into this pack is whether we can answer “yes” to the question: will this content type be used in nearly every documentation project?

Discussion question: Are there any additional templates that need to go into the core documentation pack?

Decision: These templates will go into the core template pack for the Capilano release:

- How-to
- Release notes
- Tutorial
- README
- Reference
- Explanation (concept)

Templates for the OSS community docs pack

Are there any additional templates that need to go in the OSS community docs pack?

Decision: We had a robust discussion about whether to have two template packs: one is an OSS community starter pack for beginning projects and one for mature projects. In the end, we decided to only offer one community template pack and to indicate that not all templates are

required or necessary. We'll include a guide for the template pack that indicates when you need certain types of documentation and when you don't.

For reference, see [Cameron's requirements guide](#) (see also [his GitLab template](#)), which we can use when defining which templates are required and which are optional. Using Cameron's language, we can explain the conditions under which you would use certain templates. His requirements template suggests language to use for different optional conditions for when to use template features, such as "while <state is true>, then ..." and so forth.

The templates that will go into the community pack:

- README
- Contributing guide
- Changelog
- Issue template and bug report template ★
- Merge request/pull request template
- Code of Conduct
- Templates that might be optional (not required):
 - Our team
 - Funding (new, not started, no issue)
 - Security (new, not started, no issue)
 - Support (new, not started, issue exists)

★ - The issue template and bug report template could be one template or two. We'll probably advocate for one template, but we won't know until these templates reach community review phase. The decision is yet to be made.

Future template packs

What criteria should we use when deciding to approve a template pack?

Decision: In general, new packs should be based on some sort of user request or research. We can approve the creation of future template packs that are organized around:

- Common use cases or tasks
- Needs of particular user personas
- Popular or interesting documentation frameworks

- [Maturity models](#)
- Future criteria or needs based on user research and feedback

As always, keep in mind that any given template can belong to more than one template pack, which means that template packs can be as flexible and remixable as needed.

Also keep in mind that the core documentation template pack will be the main template pack we drive users toward and that these additional template packs will appeal more to niche sets of users. Searching and navigating through these template packs is a challenge that we'll need to solve in a future release as we work on building out a better platform for finding and consuming templates through our website.

We decided to err on the side of having too many template packs rather than too few for a few reasons:

- From a user perspective, it's better to put a pack out there and get feedback on it. Based on user feedback, we can then make improvements or adjustments to the pack as needed.
- From a contributor perspective, we need work for our templateers to do. Also, this provides more support for community members to begin specializing in a particular pack or have more co-writing opportunities with people who like to work together in a pack.

Beyond these two template packs, which future template packs should we offer and which templates could go into these future packs?

With that in mind, we will add these template packs and templates to the roadmap and backlog, based on the [user research we did at the WTD conferences last year](#). Each template indicates its current status, whether it is complete, in progress, or new. For new templates, it exists whether we are tracking that with an issue so that we can create one later.

API documentation - Specialized versions of some of the basic documentation types, but in a way that is specific to APIs:

- API reference (complete)
- API overview (rework, in progress)
- API quickstart (rework, in progress)
- API tutorial (new, in progress)
- Logging (rework needed, not started)
- API use cases (new, not started, no issue)

- How does my code work? (new, not started, no issue) - Tentatively approved for this pack, but we'll need to research it more first. Could also be: systems diagram, workflow diagram, UML diagrams.

Diataxis - Content types used in Daniele Procida's [Divio/Diataxis framework](#):

- Tutorial (complete)
- How-to (complete)
- Explanation (rework, in progress)
- Reference (rework, in progress)

Documentation maturity - Templates for the minimum viable documentation starter pack. See <https://github.com/google/opensdocs> :

- Take the Google-sponsored Open Source [maturity framework](#) created by Daniel Beck and build a series of template packs around that. This framework is based around a maturity audit model and indicates which documents you would need at which stage.
- Most of the issues around this pack will need to be built out later after a deeper exploration of its contents.
- Possible templates for your documentation project as it grows and matures:
 - Style guide (complete)
 - Terminology system (new, in progress)
 - Installation guide (new, in progress)
 - Case studies/success stories (new, in progress)

User research and audience - Templates to help you research and understand your users and audience:

- User personas (new, in progress)
 - Audience analysis from Product is Docs (new, not started, no issue)
- User stories (new, not started, no issue)
- End-to-end user journeys (new, not started, no issue)
- Friction log (new, in progress)

Internal documentation - Templates for internal company and doc team needs:

- Policies (new, not started, no issue)
 - Social media engagement guidelines (new, not started, no issue)
 - Code of Conduct (in progress)

- Procedures (new, not started, no issue) - needs to be thought out and researched more in terms of which procedures would be offered.
- Meeting notes (new, not started, no issue)
 - Decision records (new, not started, no issue)
 - Sprint summaries (new, not started, no issue)
- RACI charts (new, not started, no issue)
 - Responsible, Accountable, Consulted, Informed
 - There could be variants of this template
- Doc tool or environment setup (new, not started, no issue)
 - Probably being covered by the installation guide, could swap that guide into this template pack
- Employee onboarding/employee handbook

Customer support - Templates for providing better customer support:

- Knowledge base articles (new, not started, no issue)
- Support policies (new, not started, no issue)
 - Service Level Agreements (SLAs)
 - Version support lifecycle (new, not started, no issue)
 - Supported systems or product requirements (new, not started, no issue)
 - Support responses (new, not started, no issue)
- Runbooks (new, not started, no issue)
- Contact support (new, not started, no issue)

Communication strategy - Templates to help with marketing and communicating with your customers:

- Case studies/success stories (new, in progress)
- White papers (new, not started, no issue)
- Announcements (new, not started, no issue)
 - Security alerts (new, not started, no issue)
 - Release announcements (new, not started, no issue)
- Newsletters (new, not started, no issue)
- Emails (new, not started, no issue)
- Micro-copy (new, not started, no issue) - Tentatively approved

Expanding beyond templates to other types of content

How do we feel about eventually beginning to expand beyond templates to offer topic-based guides (such as articles and best practices) or even curriculum?

Decision: Yes, we'd like to expand beyond templates at some point. In theory, these non-template content projects would go through the same contributing process as templates, but would be published in a different place, such as the website. We could explore having a UI for these articles that are similar to the [Open Source Guides](#).

Possible projects:

- Information architecture guide (complete)
- Docs landing page (new, in progress)
- Creating effective visual or multimedia content (new, not started, no issue)
- Documentation analytics (new, not started, no issue)
- Documentation feedback (new, not started, no issue)
- Product roadmaps and documentation roadmaps (new, not started, no issue)
- Business case for documentation (in progress)
- Documentation audit (new, not started, in issue)

We will need to workshop with the community to decide what to call these content types.

Possible names include: articles, best practices, topical guides, process, concepts, and others.

Template deliverable changes

Should we make changes to the template deliverables?

For reference, the current [template deliverables](#) are:

- Template file
- Template guide (also referred to as “about” because of a recent style decision)
- Deep dive (optional)
- Example (optional)

In light of all the previous roadmap discussions and all that has changed since we first introduced these deliverables, we wanted to revisit which deliverables were still necessary, whether they needed to be renamed or redefined, and whether we should add new ones.

Do we still need the example file now that the Chronologue group creates examples of the templates as part of their usability testing?

Decision: Yes, we still need to include the examples as part of the deliverables. However, the responsibility of who creates the files has changed. We should change the example file from optional to required and indicate that the templateer is not responsible for providing the example file as part of their template contributing process. Rather, the example file will be created by the Chronologue group.

We still have an open action item to work out the details of how users will find the examples for each template as they are completed by the Chronologue group.

Do we want to add an additional template deliverable that explains the process for researching, writing, and maintaining each content type?

Decision: For context, we've been having a long debate in the template working groups about whether templates should include guidance around the process of researching, writing, and maintaining the type of content type for each template. Some templateers feel that including process guidance is out of scope, while others feel that it's a major oversight not to include guidance around the creation and maintenance of content since each content type involves different strategies.

In light of this debate, we decided to create a new required template deliverable that is tentatively called "Process." As a minimum viable document, it can be very lightweight and include simple content about researching, writing, and maintaining that content type—perhaps a paragraph each. If a more detailed explanation is needed for that content type, it can go deeper.

It seems like the deep dive document hasn't been adopted much among the templateers. Do we need to make adjustments to the deep dive document as a template deliverable?

We probably haven't been getting as much use out of the deep dive document for a few reasons:

- It's unclear what kind of content the deep dive document should contain and what its scope is.
- Some templateers might be writing overly complex template guides (about files) when some of that content actually belongs in a deep dive.

For some historical context, the template leads had originally called this deliverable file the “theory” file, but it got pushback from some of the members of the community, which is why the name changed to “deep dive.”

Decision: From this point forward, we recommend making these changes to the deep dive document:

- We are changing the name of this deliverable to “resources.”
- The purpose of the resources document is to document the resources (books, blog entries, guides) that the templateer used during the research phase of creating the template. The templateer can also include high quality examples of that content type that they looked at when drawing inspiration for the template.
- We recommend using a format for reporting the resources similar to [Cameron’s references section](#) on his draft requirements template.
- The resources file has a number of advantages for:
 - The project: it gives us a way to ethically cite our sources and give credit where credit is due.
 - The contributor: it gives them a starting point and ending point (a goal + a final deliverable) for the research phase.
 - Template readers who choose to look at the document: it explains the theory behind why certain decisions were made to the template, which can help them understand why the template author made certain design decisions. It can also help them know how they can customize the template to their organization’s needs.
 - Future template maintainers: this document help them understand why a template was developed a certain way. It will help them decide whether to change the template in future (or not).
- In some of our ongoing template projects for the Capilano release, some templateers had planned to use the deep dive file to explore some aspect of that content type that deserves a deeper explanation but is only relevant to users who truly want to go in-depth. For that use case, we recommend creating an additional file that is a topic-based article. We could eventually include that article in our topic-based guides (see [Expanding beyond templates to other types of content](#) for more information).

Summary of use cases for template deliverables

This table explains how we conceive the template deliverables being used by our target users.

Req?	Deliverable	Persona	Use case
Yes	Template file	Developer-Writer Darby	Just give me something easy I can fill out and nothing more!
Yes	Guide (about)	Developer-Writer Darby	When I get stuck, I'll refer to the guide to help me out. Please keep this simple and lightweight for me!
Yes	Process	Developer-Writer Darby, Doc System Owner Olly	Help me understand more about best practices involved in researching, writing, and maintaining this content type.
Yes	Resources	Doc System Owner Olly, Template Terry	Help me understand the theory that informs this template on a deeper level. If I want to customize this template for my project, I can use this guide to make informed changes. If I need to maintain this template and make changes to it in the future, I can understand the thinking that went into the template originally.
Yes	Example	Developer-Writer Darby	Help me see an example of this template in action so that I can see what "good enough" looks like.
No	Topic-based article	Doc System Owner Olly, Developer-Writer Darby	I want general guidance about the other elements of this content type that can't be captured easily in the template or the supporting template deliverables.

RESOURCES

For recordings of the meetings where we made these decisions, see:

- [Template roadmap discussion 1](#)
- [Template roadmap discussion 2](#) - NOTE that we forgot to record the first 30 minutes
- [Template roadmap discussion 3](#)
- [Template roadmap discussion 4](#)

- [Template roadmap discussion 5](#)