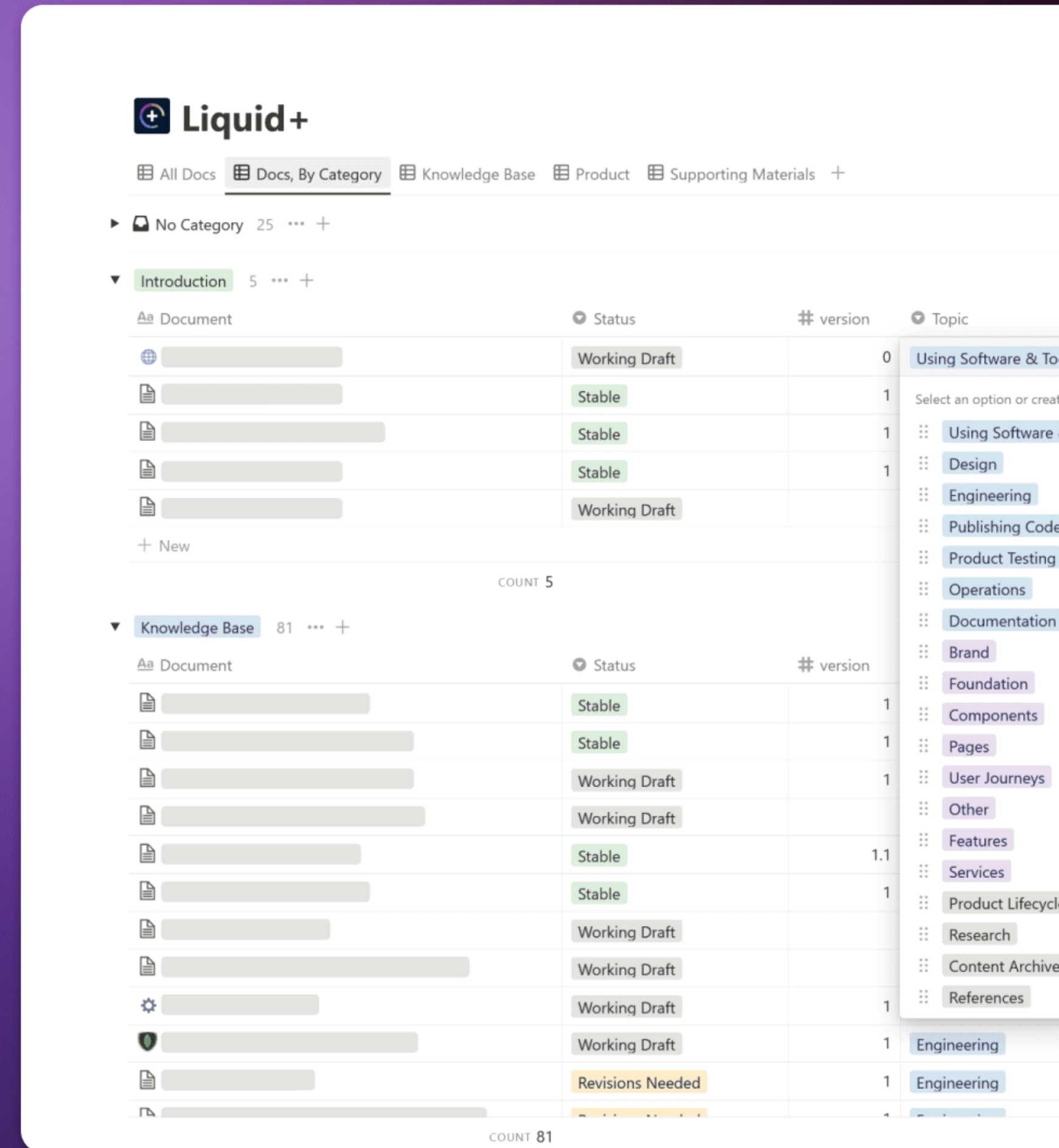


Liquid+ Documentation

Systematizing Product Knowledge for Better Collaboration

My Role: Product Designer, Researcher



The Core User Problem

Team members could not easily find, or trust documentation.

- Gaps in knowledge
- Information repeated
- Knowledge held by specific team members only
- Documents contradicted each-other

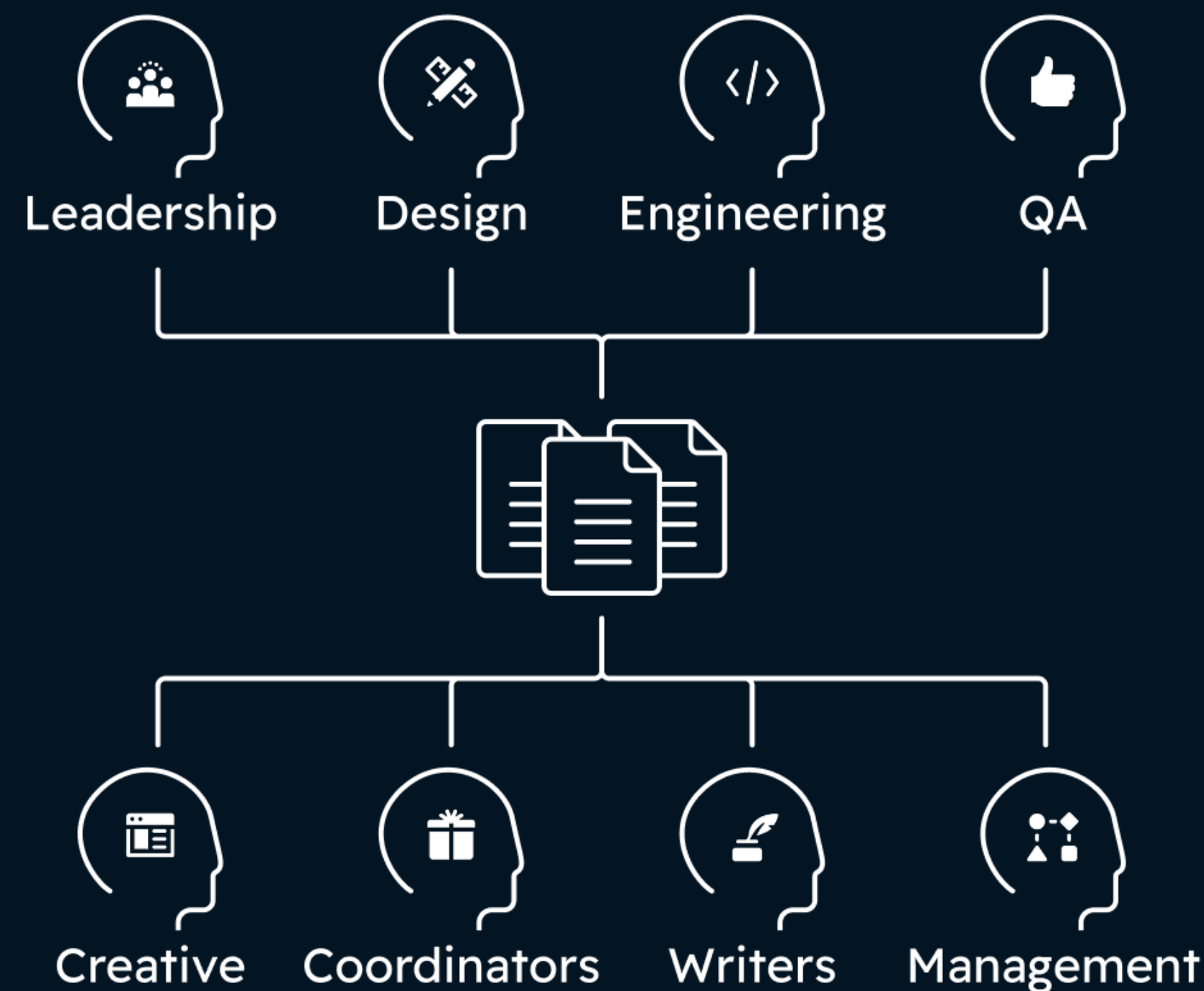
...And a Problem for the Business

Effectiveness was impeding our ability to meet objectives.

- Product was more costly to produce
- The growth of the product was slow
- Quality and timelines harder to meet, and more costly

Challenge

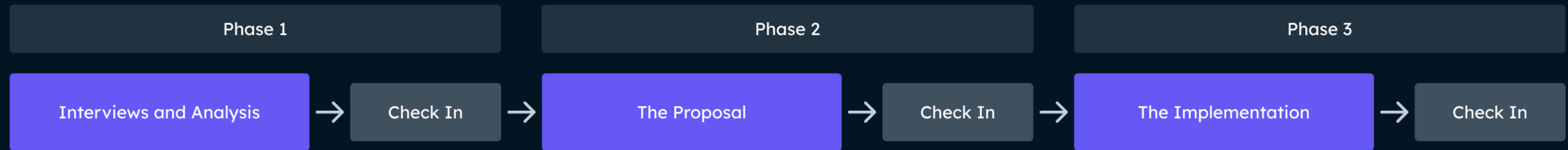
How do you craft a documentation solution that works for a cross-functional team?



Hypothesis and Validation

A successful system would help people **find information, know what is **true**, and understand how to **do** things.**

Would this be true for this team? And how would the solutions manifest?



1 – Interviews and

16 interviews were conducted across the team.

Designers

Developers

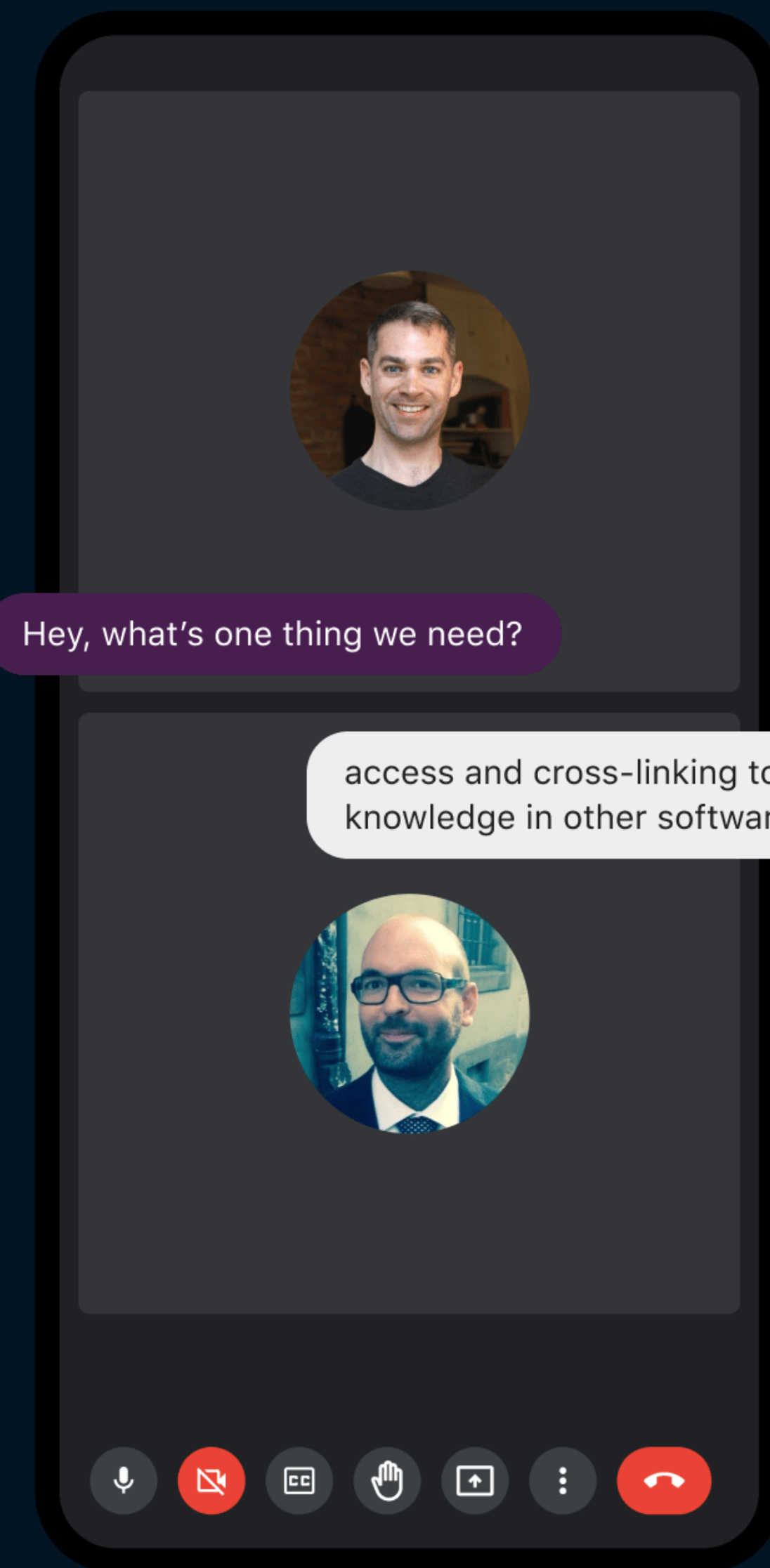
Quality Assurance

Project Leads

Coordinators

Copywriters

Operations



Qualitative interviews showed what the team wanted most.

- Documents for getting work done
- Documents not related to the service
- Knowing how to document things
- Knowing requirements
- A structure not based on disciplines
- Ability to add supporting materials to documents
- References being used in relevant ways
- Truth is knowable
- Tracking status/progress
- Multiple ways to find documents

Doc	origin	Attributes (what it's about?)	Topic	facilitator?
Sprint Board Task Statuses	general	scrum how-to ref	facilitator	<input checked="" type="checkbox"/>
definition of development	general	scrum qa how-to ref	facilitator	<input checked="" type="checkbox"/>
Design - Features	general	scrum design how-to ref	facilitator	<input checked="" type="checkbox"/>
definition of complete	general	scrum how-to ref	facilitator	<input checked="" type="checkbox"/>

Item	Noted By	Mentions	(Potential) Scope
getting started, or how to X guides (general)	leads qa design dev ops	8	doc section
requirements (business, functional, technical)	leads qa design dev ops	7	segment
general information on the service	qa copy dev ops	5	doc section
docs for processes and standards (catch-all for env setup, coding guides, special cases)	dev ops	5	doc section
search function	leads dev ops	4	ix
doc linkage: should reference other documents where relevant	leads design dev	4	segment
how to use documentation (where to put, how to write, templates)	leads design dev ops	4	doc section
feature documents	leads design dev ops	4	doc section
diagrams (embedding preferred, linking needed)	leads dev	3	segment
revision history of a doc (who, what)	leads dev	3	segment
research information	dev	3	segment doc
table of contents	design dev qa	3	segment
API documentation	qa design dev	3	section
component documents	design dev	3	doc
filtering by category/subcategory	leads dev	2	ix
robust editor	design dev	2	ix
the lifecycle of a thing (wd, 50%, complete, etc...)	leads ops	2	segment
doc health summary	leads dev	2	segment
document summary (doc, component, feature, ref GUI's feature notes)	dev	2	segment

Analyzing patterns from existing documents suggested a number of possible solutions.

Are factual and how-to documents different? What about information that crosses domains?

Types of Documents

Default view +

Filter Sort Q ... N

Doc	origin	Attributes (what it's about?)	Topic
Sprint Board Task Statuses	general	scrum how-to ref	facilitato
definition of development	general	scrum qa how-to ref	facilitato
Design - Features	general	scrum design how-to ref	facilitato
definition of complete	general	scrum how-to ref	facilitato
qa severity levels	general	scrum qa how-to	facilitato
platform connection diagrams	general	service ref	thing
git rebasing guide	general	git how-to ref	domain
points estimation	general	scrum ref how-to	facilitato
Draft: L+ Arch Proposal	general	service ref	thing
Draft: Service Guidelines	general	service approach how-to ref	
epic/task templates	general	scrum how-to ref	facilitato
deployment strategy	general	git how-to ref	domain
feature branch strategy	general	git how-to ref	domain
coding standards	general	how-to coding ref	domain
content pipeline assets	general	coding approach how-to ref	facilitato
clickup notification settings	general	scrum how-to ref	facilitato
quest creation guide	general	pipeline how-to ref	facilitato
reviewing merge requests	general	git how-to ref	domain

2 – The Proposal

Documentation goals

- clear and easy to navigate
- easily extensible/flexible
- capture the needs of each domain involved
- efficient, and prevent effort duplication
- predictable at the macro and micro level
- reflect how the team thinks, and what they need

Documents (specifically)

- assessable/parseable at a glance
- written by domain experts
- clear with what they document

Working through several approaches revealed some consistent themes about documents.

- **Introductions** to the product
- **Knowledge** about how to do work
- **Facts** about the **product**
- Information that **supports** the product

Pass 0

Liquid+
Features
=> What is the role of a quest
Design
=> How should a quest look
Front-end Browser app
=> What is the logic behind
Front-end Native app
=> What is the logic behind
Environments
=> What services are involv
Back-end
=> What does the Quest serv
Quality assurance
=> How do I test quests?
External services
=> How are fan activities g
Admin
=> How do I add a quest in
Support
=> What are most frequently
ts?
Branding & Marketing
=> How should we setup a pl
Organisational processes
=> Where do I add a new tas

Pass 1

learn what something is, or how do do s

- ▶ introduction
- ▶ general information
- ▶ scrum (sprints)
- ▶ content
- ▶ engineering
- ▶ codebase
- ▶ quality assurance
- ▶ document templates (is separation

Type '/' for commands

💡 The general setup implicitly says
external services), how do we rec

To me, DevOps are things like update
the platform and environments, etc. -

Knowledge Base by Topic

- ▶ introductory information
- ▶ technologies
- ▶ documentation
- ▶ engineering
- ▶ publishing code
- ▶ quality assurance
- ▶ operations

- ▶ content (is this relevant? content is

Pass 2

Liquid+

how to make things mutually exclusive; that's impossible

Knowledge Base

learn what something is, or how do do something (portable to any project?)

- ▶ introductory information
- ▶ technologies
- ▶ documentation
- ▶ design (!)
- ▶ engineering
- ▶ publishing code
- ▶ quality assurance
- ▶ operations

The Product

what is or will be (facts; also 'how' when talking about requirements)

a roadmap is also somewhat relevant in this category.

- ▶ brand
- ▶ features
- ▶ services
- ▶ design
- ▶ flow/user journeys
- ▶ pages

Supporting Materials

- ▶ research
- ▶ proposals
- ▶ content archive
- ▶ glossary

Creating logical sections within each category was mostly easy, except for product docs.

Knowing where to look needed to be easy. So I wanted to make sure the total depth was shallow, preventing an overly-formal structure.

The Product (v4, 10)

- ▶ brand
- ▶ foundation (design in polaris)
- ▶ content
- ▶ patterns
- ▶ components
- ▶ pages
- ▶ user journeys (flows)
- ▶ types of content
- ▶ features
- ▶ services

Product (v8, 9)

- ▶ brand
- ▶ writing
- ▶ design
- ▶ components
- ▶ pages
- ▶ journeys
- ▶ types of content
- ▶ features
- ▶ services

The Product (v5, 8)

- ▶ brand
- ▶ design
- ▶ components
- ▶ pages
- ▶ journeys
- ▶ types of content
- ▶ features
- ▶ services

Product (v9, 10)

this version is more about the design system that is made, and then the application that implements it (first 5, last 5)

- ▶ brand
- ▶ writing
- ▶ design foundation
- ▶ patterns
- ▶ components
- ▶ pages
- ▶ journeys
- ▶ types of content
- ▶ features
- ▶ services

The Product (v6, 6)

- ▶ brand
- ▶ design
- ▶ user application
- ▶ types of content
- ▶ features
- ▶ services

Product (v10, 5)

- ▶ brand
- ▶ writing
- ▶ foundation
- ▶ user application
- ▶ features & services

Product (v7, 7)

- ▶ foundation
- ▶ components
- ▶ pages
- ▶ journeys
- ▶ types of content
- ▶ features
- ▶ services

The Proposal

Introduction

Quick Overview

What is the Project
Who's Working on it
etc...

Knowledge Base

How we Work

Using Software & Tools
Design
Engineering
Publishing Code
Product Testing
Operations
Documentation

The Product

Facts

Brand
Foundations
Components
Pages
User Journeys
Types of Content
Features
Services

Support

Supporting Knowledge

Product Lifecycle
Product Testing
Research
Content Archives
References
Glossary

A Problem of Acquisition and Truth

**Even if you can find a document,
how do you quickly find the
information you need, and how
do you know what it says is true?**

**Several existing patterns
(in the wild) helped
solve these document
specific needs.**

📄 Getting Started with Documentation

📄 Migrating a Document

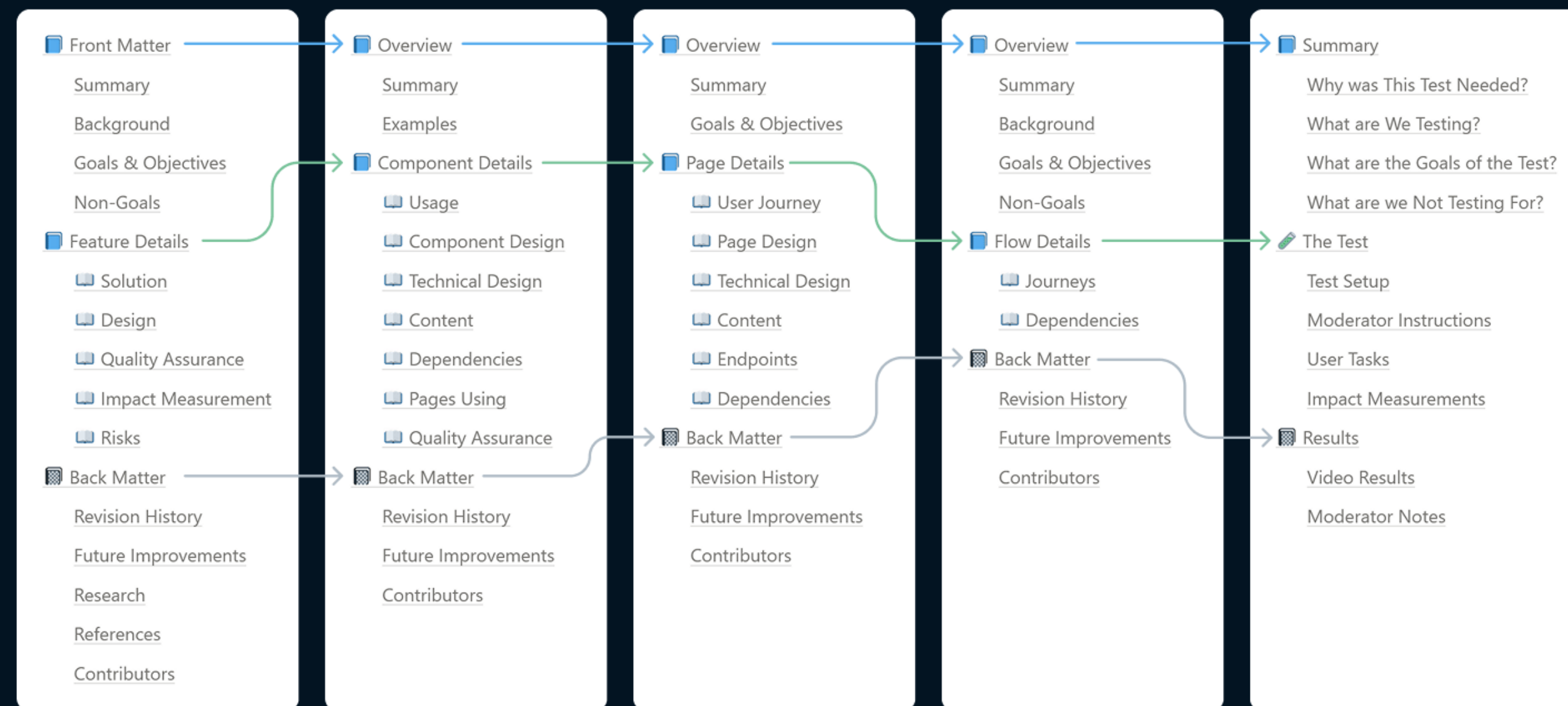
📄 Document Lifecycles

📄 How to Write a Document

📄 Editing Documents

📄 <Template/>

Documents borrowed from books, feature docs, and the overall theme of the system itself: **summarize**, **detail**, **support**.



Document lifecycles borrowed from several sources.



Presentation and Feedback

LIQUID+


DOCUMENTS

PROPOSAL (NO. 2)

SOLUTIONS

- Structure
 - Approach
 - Effective Documents
 - Trustable Documents

Introduction <ul style="list-style-type: none">• Product Overview• Technologies & Software Used• The Team	<i>Summarize</i>	<ul style="list-style-type: none">• Front Matter<ul style="list-style-type: none">◦ Summary◦ Background◦ Goals & Objectives◦ Non-Goals
Knowledge Base <ul style="list-style-type: none">• Using Software & Tools• Design• Engineering• Publishing Code• Product Testing• Operations• Documentation		
Product <ul style="list-style-type: none">• Brand• Foundation• Components• Pages• User Journeys• Types of Content• Features• Services	<i>Detail</i>	<ul style="list-style-type: none">• Feature Details<ul style="list-style-type: none">◦ Solution◦ Design◦ Quality Assurance◦ Impact Measurement◦ Risks
Supporting Materials <ul style="list-style-type: none">• Product Lifecycle• Product Testing• Research• Content Archive• References• Glossary	<i>Support</i>	<ul style="list-style-type: none">• Back Matter<ul style="list-style-type: none">◦ Revision History◦ Future Improvements◦ Research◦ Archive◦ References◦ Contributors



3 – Implementation

Interviews contained a few highly requested features for an environment.

The current environment was slow and hard to navigate, so other options needed to be researched. They needed to

- be fast;
- be easily searchable;
- have a robust editor; and
- include embedding

A competitive analysis of other environments helped understand which apps were best suited to the team's needs.

App Options (data)

You may spend at most 1hr exploring these applications.

Name	Rating	Abs Value	URL	Price	IA + Nav	IA Lvl	Editor Options	Edit Coll...	Embedding	Lifecycle Handling
Notion	A	4.6	https://www.nc	8/user/mo, or	A	inf	A	Full	B	A
Slite	A	4.6	https://slite.co	6.67/user/mo	A	inf	A	Full	S	B
Almanac	B	4.5	https://almana	99/mo (unlimit	B	inf	A	Full	S	S
Nuclino	B	4.5	https://www.nu	5/user/mo	A	inf	B	Full	S	B
TL's Resource Hub	B	4.4	https://hub.tea	free!	B	inf	S	None	A	B
Outline	B	4.2	https://www.ge	79/mo, or 4/us	A	inf	B	Full	A	B
Document 360	B	3.9	https://docum	299/mo (15 ac	B	inf	C	None	F	S
Tetra	B	3.6	https://tetra.c	8.33/user/mo	B	inf	C	None	B	A
ClickUP	B	3.4	https://clickup	free (9/user/mo	B	5	A	Full	B	A
Bookstack	B	3.2	https://www.bc	Free	C	4	B	None	D	B
Guru	C	3.6	https://www.ge	\$10/user/mo	C	3	C	None	C	B
Frontify	C	3.2	https://try.fron	275/mo	B	2	A		S	F
ClickHelp	C	3.1	https://clickhel	99/user/mo	B	inf	B		F	B
Zero height	D	3	https://zerohei	16/user/mo or	C	4	C		B	F
Read the docs	F	0	https://docs.re	free						
Paligo	F	0	https://paligo.r	179/mo, or 265						
Kipwise										

+ New

COUNT 17 EMPTY 1

1st Round

Editor Options

Name	Type	Notion	Slite	BookStack	Resource Hub
code, block	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
code, inline	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
divider	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
h1	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
h2	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
h3	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
image	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
list, bullets	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
list, numbered	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
quote	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
strikethrough	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
underline	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
video	basic	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
callout	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
comments	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
date	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
details (> disclose)	advanced	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
embed, iframe	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
embedding, non-iframe	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
list, checkboxes	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
mention, page (link)	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
mention, person	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
table, simple	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
text bg colors (highlights)	advanced	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
text colors	advanced	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
breadcrumb	specialty	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
contents	specialty	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
directories	specialty	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
document card	specialty	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2nd Round (finalists)

After a few rounds, the finalists were either



Notion

or



slite

Rating: A / 4.7

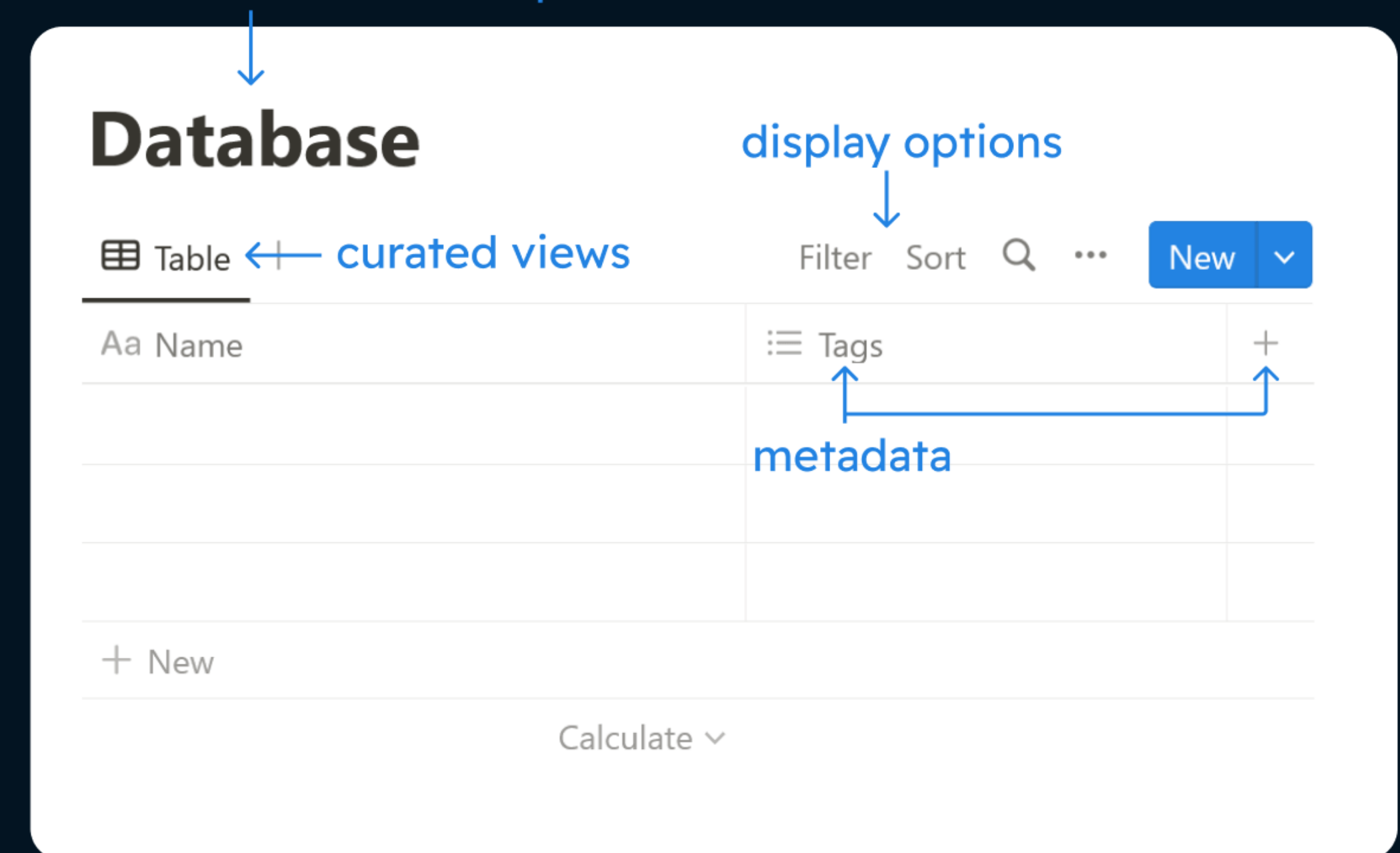
Rating: A / 4.6

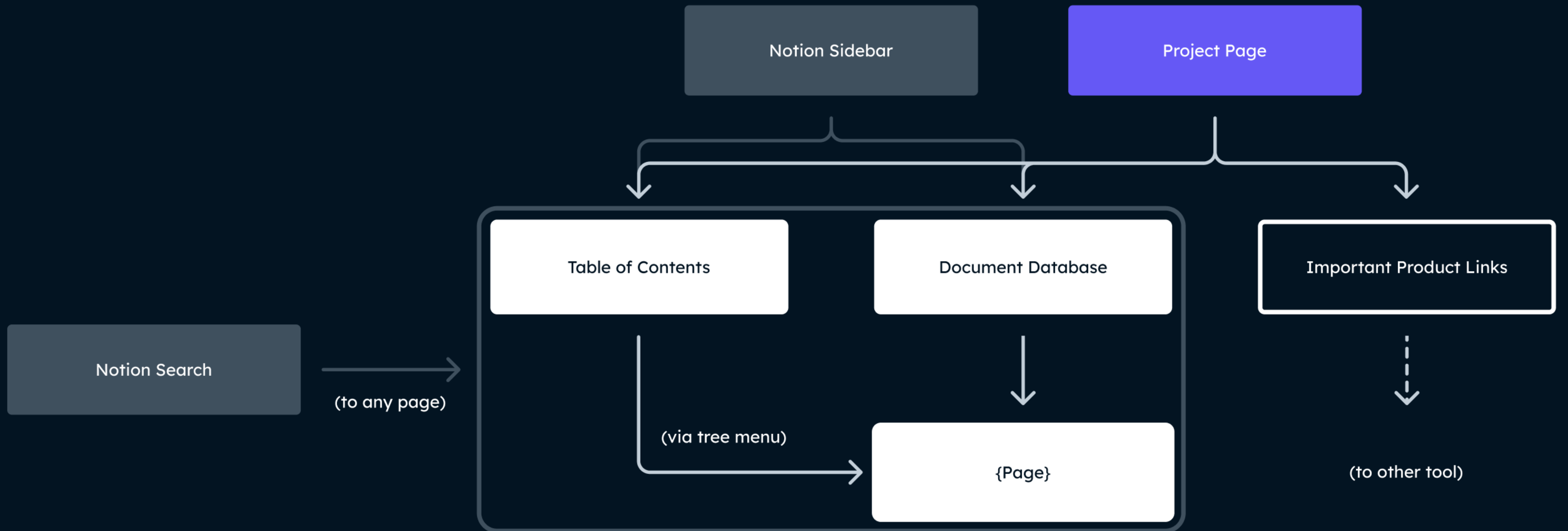
A Question and Challenge

Databases were more appealing, despite the added complexity.

Choosing the database approach created an additional challenge however. A manually generated tree would be necessary to help newer users.

referencable as a user-specific view





Revision history for QA was lacking, so we needed to create our own solution.

Team members needed to have a clear snapshot, and finding old data needed to be easy.

Revision History

Default view

Log

Version	Date	Summary	Contributors
1.8	2022/06/08	Troubleshooting: Added Windows + Java Virtual Machine guides	
1.7	2022/05/12	Updated "Inspecting Network Calls" debugging section	
1.6	2022/05/11	Added troubleshooting point on IBAgent-iOS	
1.5	2022/05/11	Apple Silicon M1 setup - Added missing step for Android	
1.4	2022/04/12	Added setup for Apple Silicon M1	
1.3	2022/04/06	Added the info about configuring trusted certificates for debugging in Android.	
1.2	2021/12/22	Added general debugging.	
1.1	2021/12/07	Merged multiple setup documents into one.	

Wrap Up

Project Outcomes

- **The team is happier, contributing much more, and trust is higher.**
- **Velocity of the team has slightly increased.**
- **Other teams adopted this approach for their own projects.**

Remaining Challenges

Some hesitation with contribution remains

Some adjustments would be helpful.

Improvements

- Incorporate documentation into sprint work
 - Review process and questions with staff
 - Encouragement
-
- Using databases for serialized content
 - A unique space for requirements
 - More utility with the sidebar

Final Product

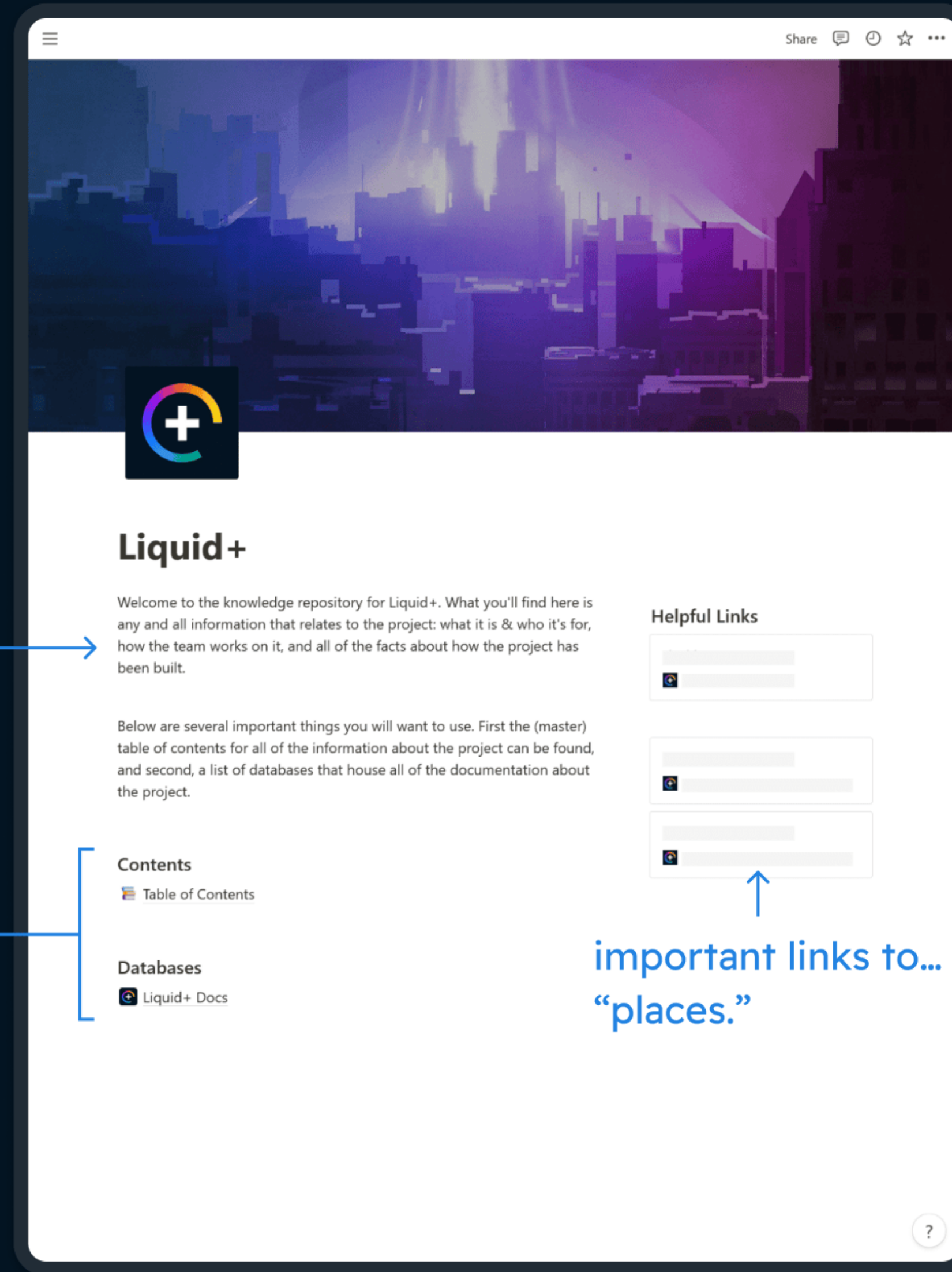
overview of the space

multiple paths to documents

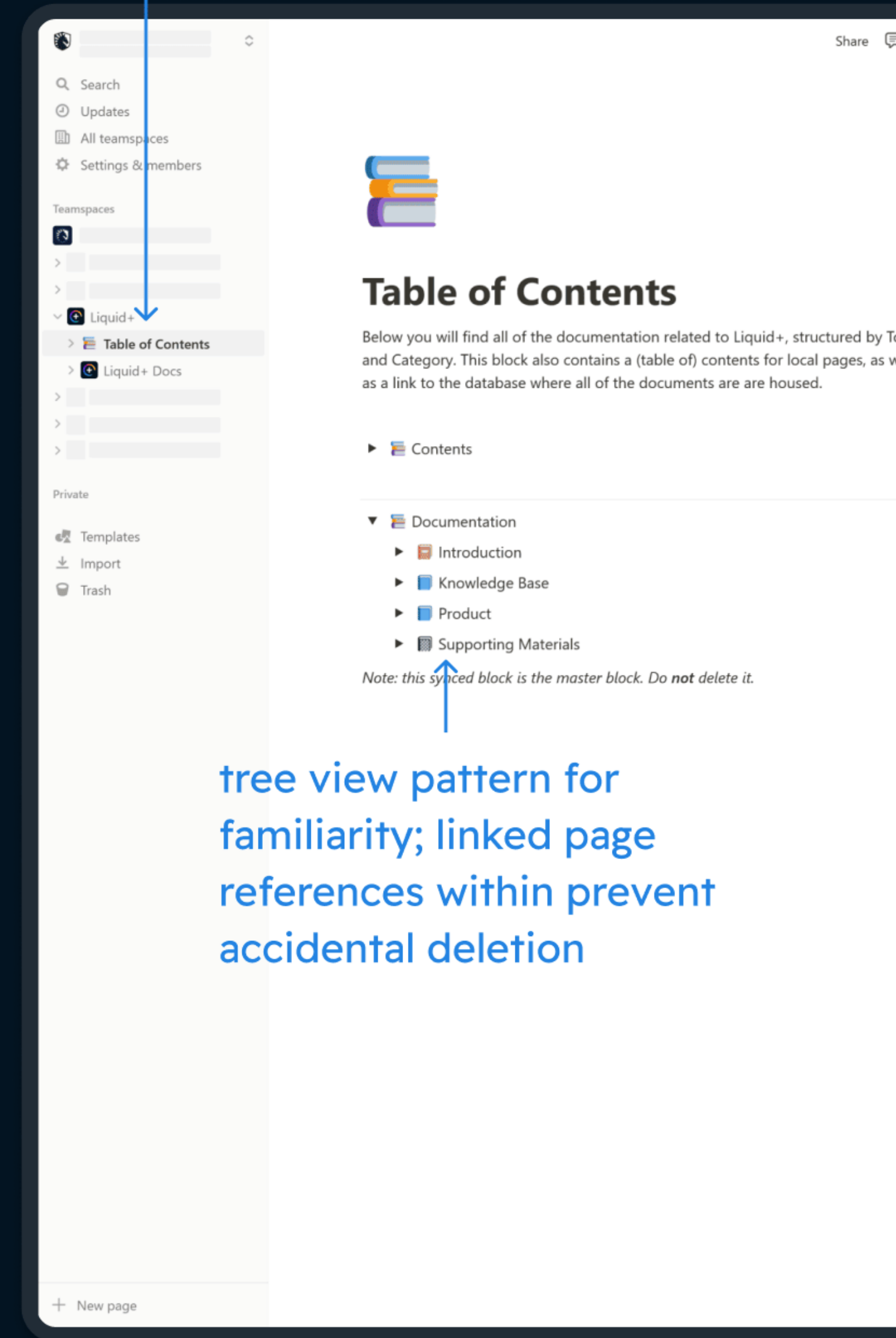
important links to...
"places."

quick paths to docs from sidebar

tree view pattern for familiarity; linked page references within prevent accidental deletion



Home Page



Contents and Areas of Documentnation

focused collections of docs maturity snapshot

templates for common page types

major product areas

The screenshot shows the 'Liquid+ Docs' interface. At the top, there are navigation tabs for 'All Docs', 'Docs, By Category', and 'Knowledge Base'. Below this, there are two main sections: 'Introduction' (5 documents) and 'Knowledge Base' (81 documents). The 'Introduction' section is expanded, showing a table of documents with columns for 'Document', 'Status', '# version', 'Topic', and 'Properties'. A dropdown menu is open for the 'Using Software & Tools' topic, listing various sub-topics such as 'Design', 'Engineering', 'Publishing Code', 'Product Testing', 'Operations', 'Documentation', 'Brand', 'Foundation', 'Components', 'Pages', 'User Journeys', 'Other', 'Features', 'Services', 'Product Lifecycle', 'Research', 'Content Archives', and 'References'. The 'Knowledge Base' section is also visible, showing a list of documents with various statuses like 'Stable', 'Working Draft', 'Revisions Needed', and 'Deprecated'.

database categorization (same as tree view)

thematic sections (for all document-types)

tags for themes and scope

The screenshot shows a 'Component Page' template. On the left, there is a 'Contents' table of contents with sections like 'Overview', 'Summary', 'Examples', 'Design', 'Code Snippet', 'Component Details', 'Usage', 'Component Design', 'States & Behavior', 'Technical Design', 'Blueprint', 'Properties', 'Functional Design', 'Content', 'Dependencies', 'Pages Using', 'Quality Assurance', 'Back Matter', 'Revision History', 'Future Improvements', and 'Contributors'. The main content area on the right shows the 'Overview' section, which includes a 'Summary' (what is this component), 'Examples', 'Design' (graphic example(s) of the component), and 'Code Snippet' (//implementation code). Below this, the 'Component Details' section is visible, including 'Usage' (how to use this component) and 'Component Design'.

Database View, with Key Properties (Documents)

A Single Page (template)



Designer, Researcher

Team

Nik Jeleniauskas

Me, The Liquid+ Team as Support