DOCUMENT360

Webinar On

Documenting large-scale and complex products

Speaker

Sofia Emelianova

Technical Writer

Google Chrome Developer Tools

Who am I?

Hey there!

I’m Sofia.

Technical Writer at Google

Owner of Chrome DevTools documentation

GitHub: @sofiayem

LinkedIn: www.linkedin.com/in/sofia-yemelianova/

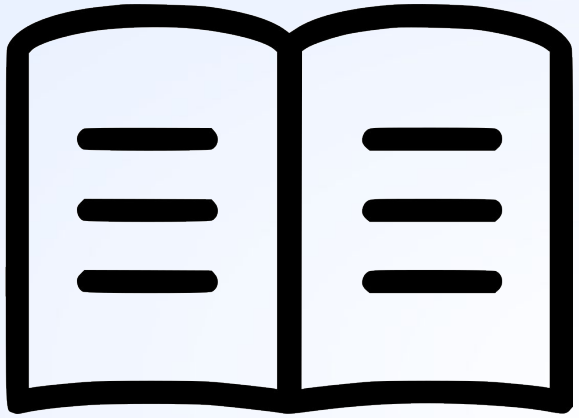
DOCUMENT360

Who is this webinar for?

Technical Writers, who:

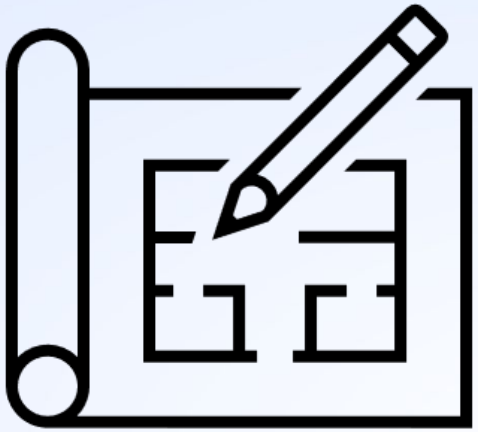
- Work on large and/or complex products solo
- Start a documentation project from scratch
- Need to bring structure to chaos
- Want to expand sphere of influence on your writers team
- Work less and do more

Overview



- Discover your scope and manage your priorities
- Build an information architecture and content strategy
- Integrate yourself into the development cycle
- Make your progress transparent for management
- Prepare for potential task delegation and onboarding of fellow Technical Writers

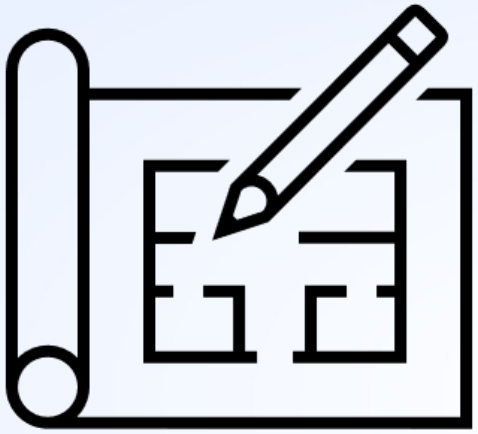
Discover your scope



- Use your product
- Talk to Subject Matter Experts and learn
- Set your technical writing goal

Next? Decompose down to tasks and log into a bug tracker.

Build an information architecture



Information architecture simplified:

- **Structure**
- Reader experience
- Information discoverability

Of a documentation set

Structure

Structure should **follow and guide** the user journey of your product

Structure

Example:

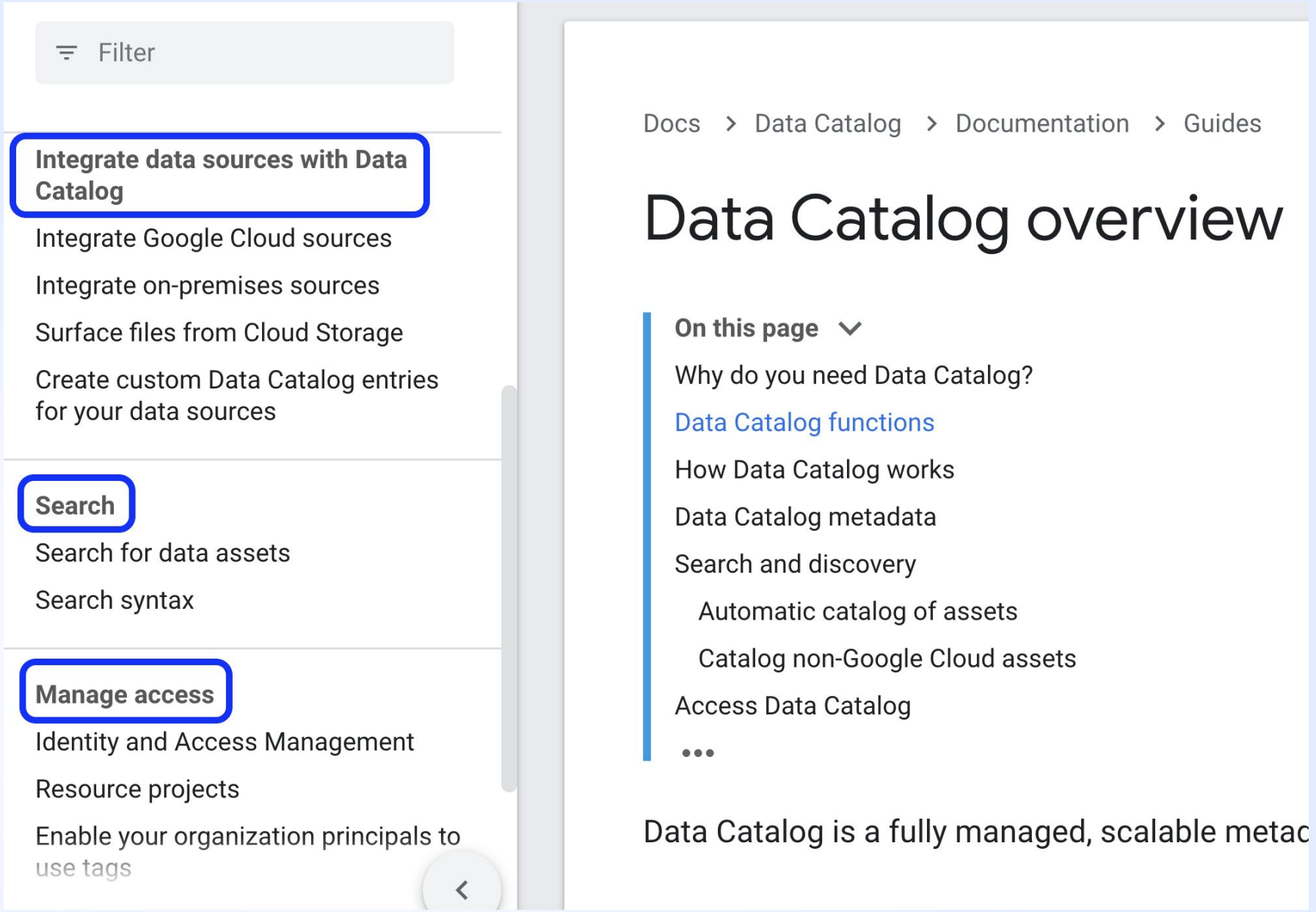
Google Cloud Data Catalog is a metadata management service. In Data Catalog,

companies store data about data on their internal resources and databases.

Structure

What do you do with a metadata management service?

1. Integrate your systems and data sources
2. Store, create, and maintain metadata
3. Search for metadata across your systems
4. Grant and control access to internal users of Data Catalog

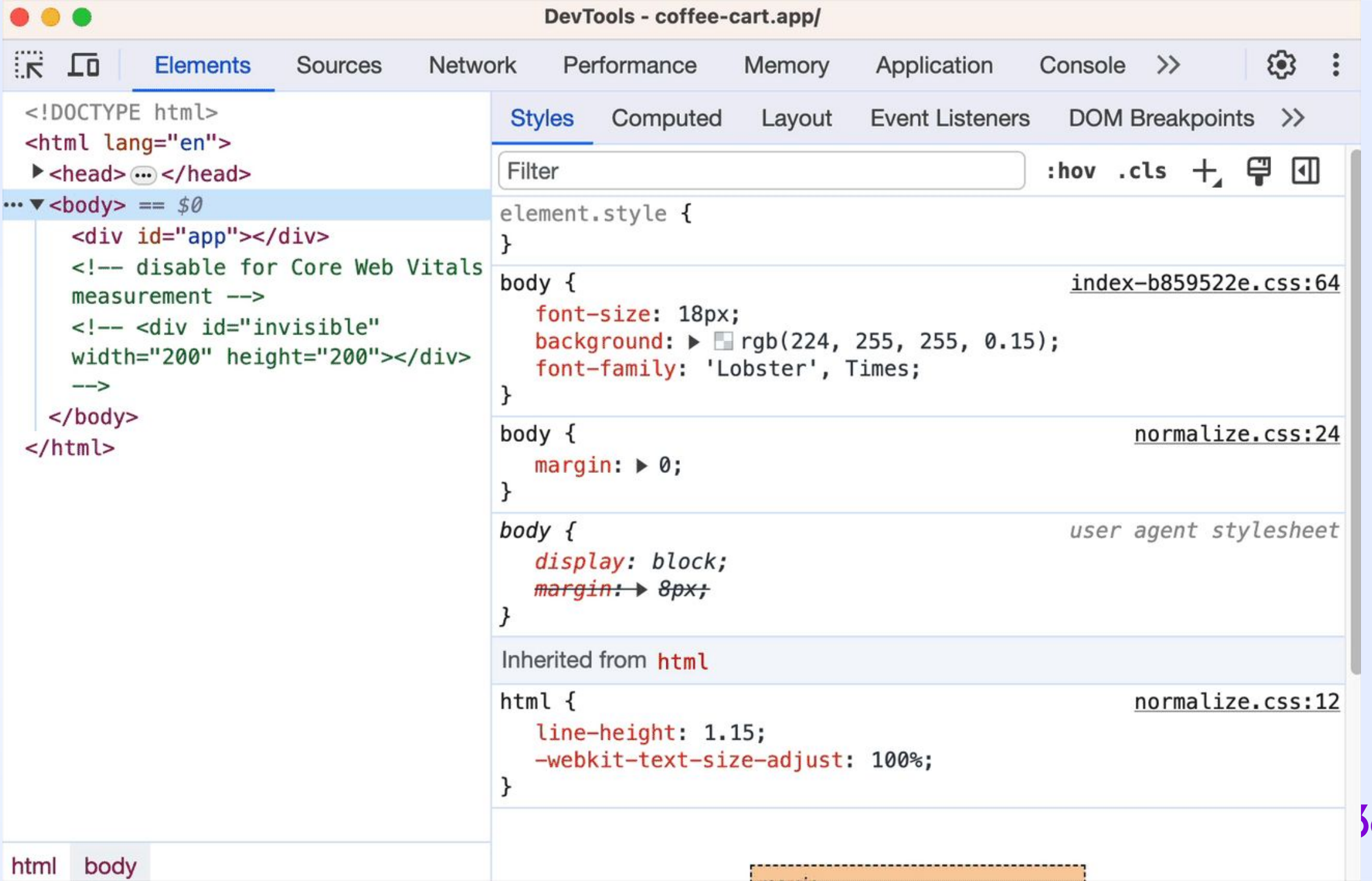


Structure

Counter example:

Google Chrome Developer Tools is a collection of tools that let you debug everything on the web.

There’s no high-level user journey.


DevTools - coffee-cart.app/
Elements
Sources
Network
Performance
Memory
Application
Console
<!DOCTYPE html>
<html lang="en">
<head> … </head>
<body> == $0

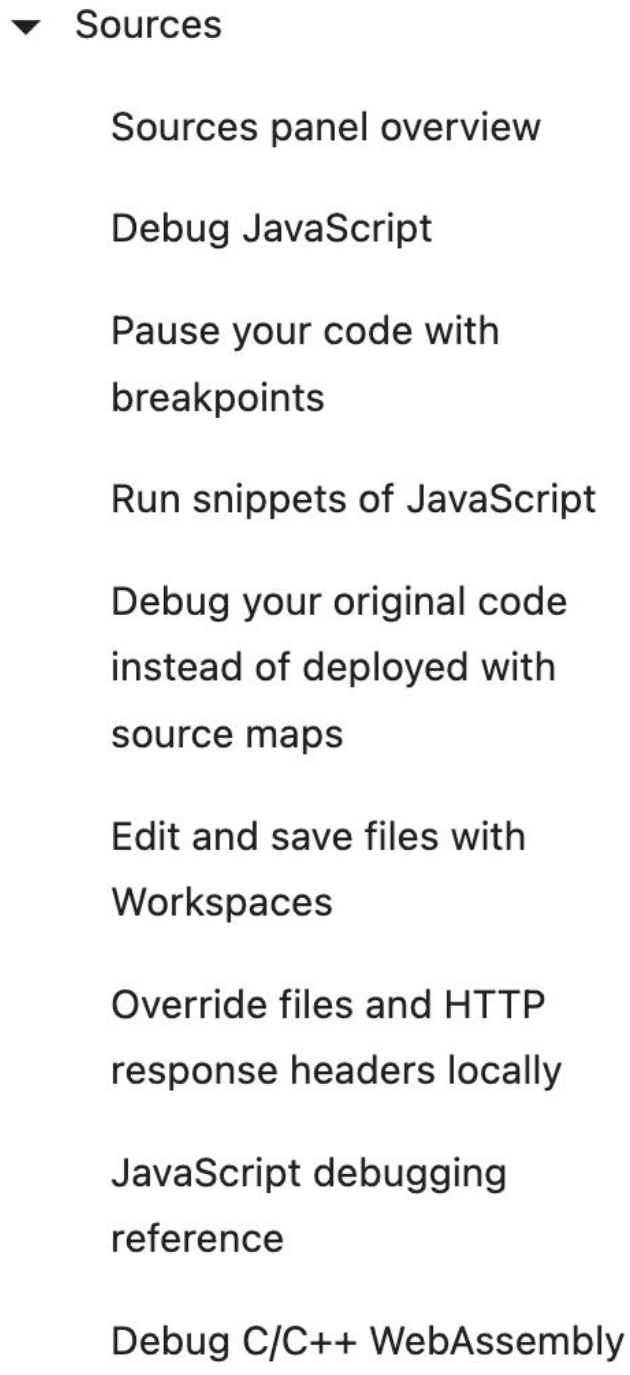
<!-- disable for Core Web Vitals measurement -->
<!-- <div id="invisible" width="200" height="200"></div> -->
</body>
</html>
html
body
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Filter
:hov
.cls
element.style {
}
body {
index-b859522e.css:64
font-size: 18px;
background: rgb(224, 255, 255, 0.15);
font-family: 'Lobster', Times;
}
body {
normalize.css:24
margin: 0;
}
body {
user agent stylesheet
display: block;
margin: 8px;
}
Inherited from html
html {
normalize.css:12
line-height: 1.15;
-webkit-text-size-adjust: 100%;
}

Structure

It’s a set of workflow snippets, that web devs can integrate into their workflow.

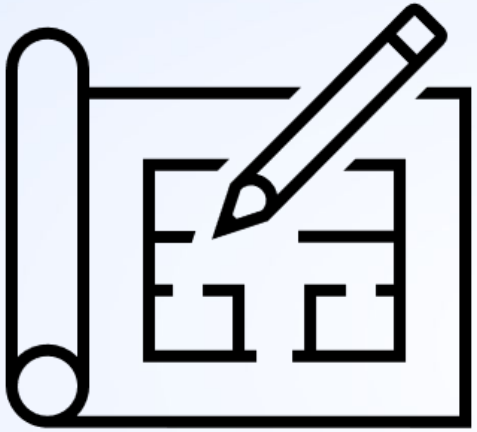
In this case, the doc structure:

- First level of ToC are sections about panels (tabs).
- There’s a separate set of tutorials that teach the basics.
- Important and big features get their dedicated tutorials.



- Overview of a panel
- Basic workflow 1
- Basic workflow 2
- Basic workflow 2
- Important feature 1
- Important feature 2
- Important feature 2
- Features reference (checkboxes and nuances)
- Extra niche thing

Build a content strategy



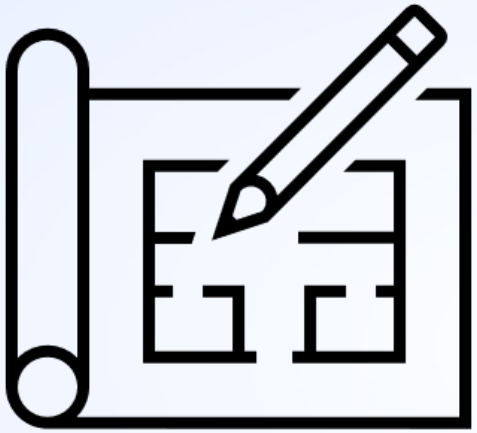
Content strategy simplified:

- Planning
- Development
- Management

of content

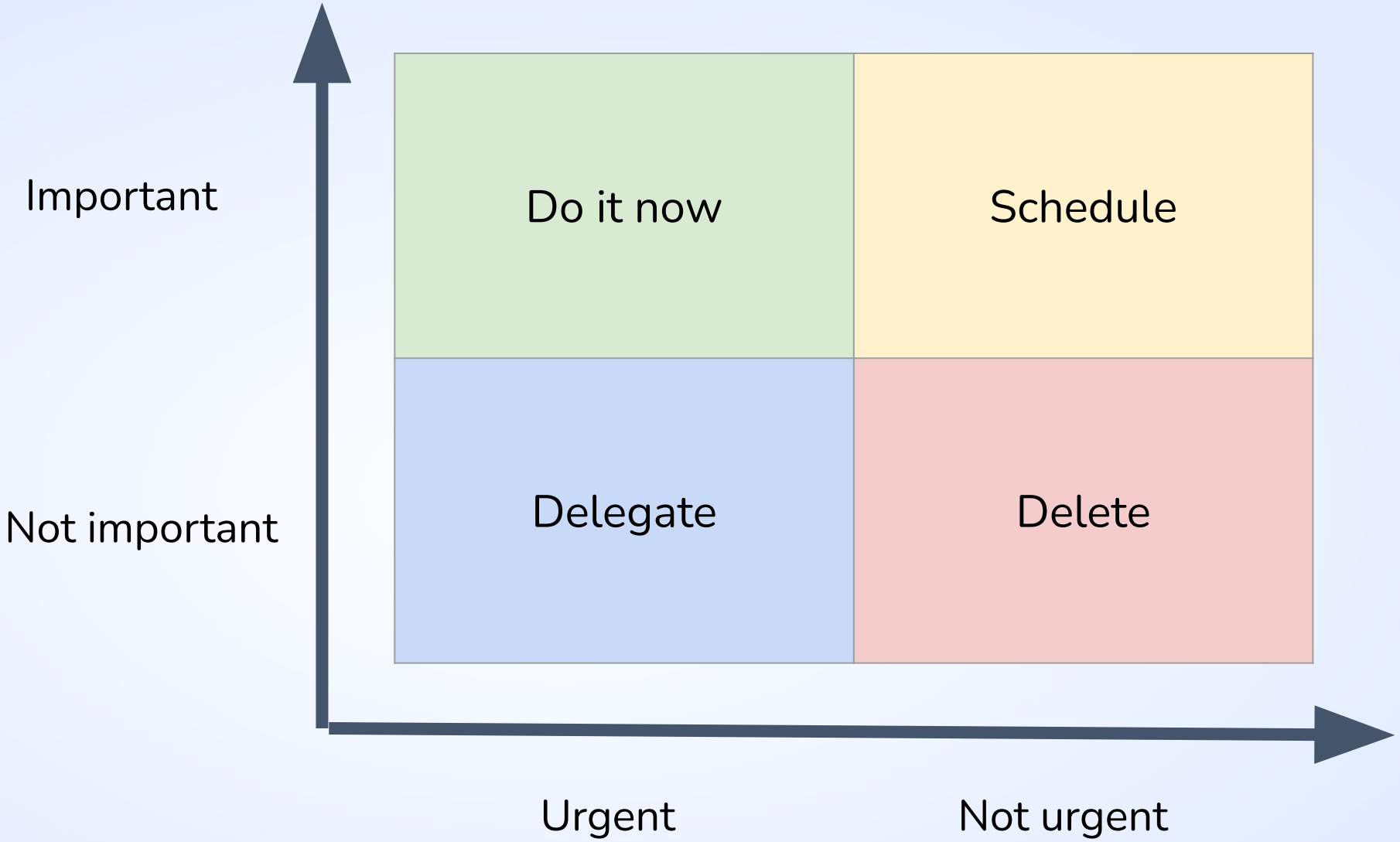
Now is the time to create a backlog

Create a backlog



- Log uncovered and/or outdated tasks into a bug tracker
- Guesstimate T-shirt sizes for your tasks (S, M, L)
    - Avoid XL, though, it means it’s not decomposed.
- Divide your work into “streams”
    - New content, content updates, org work, and misc (opt)
- Use the Eisenhower matrix to determine priorities for each stream

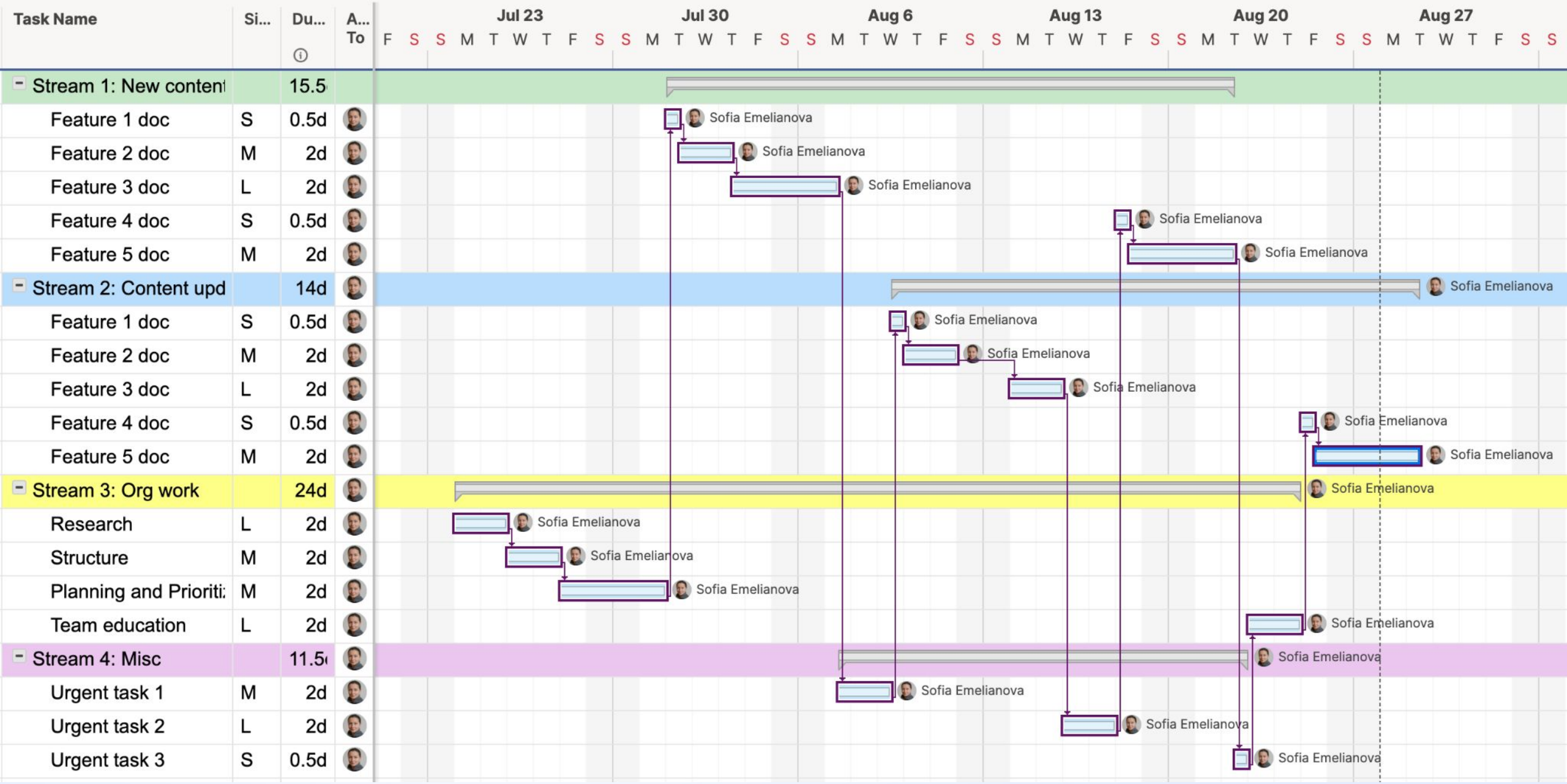
Prioritize



Log your progress

Use a Gantt chart. It helps you:

- Illustrate your work completed over a period of time
- Lets you log dependencies
- Lets you assign tasks to people

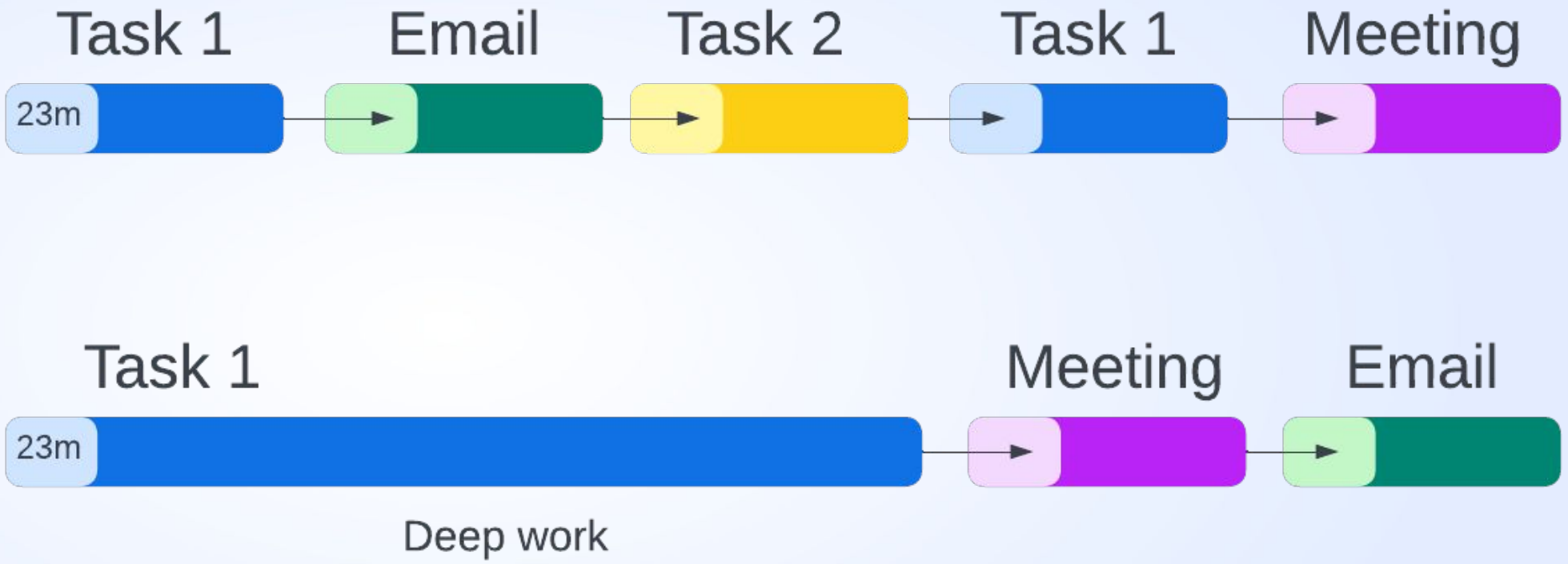


Multitasking?

What seems like multitasking is, in reality, rapid context switching.

Context switching has cognitive load

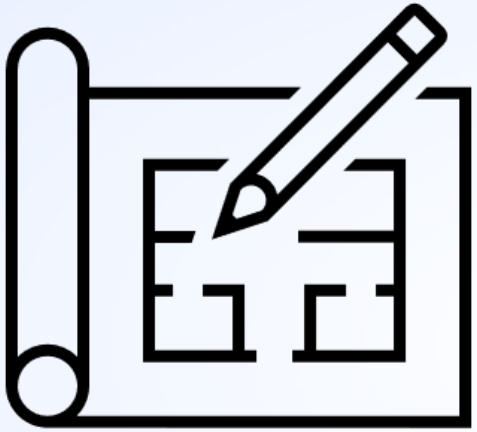
It can take more ~23 minutes to regain focus after a switch.



Source: [The Cost of Interrupted Work: More Speed and Stress]

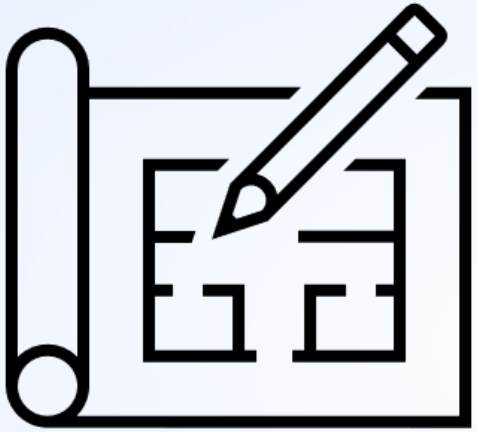
DOCUMENT360

Make your progress transparent



- Have regular 1-on-1s with your manager
- Make sure they understand the importance of documentation
- Show them the Gantt chart and explain the cognitive load of context switching
- Explain the necessity of organization work

Negotiate for more resources



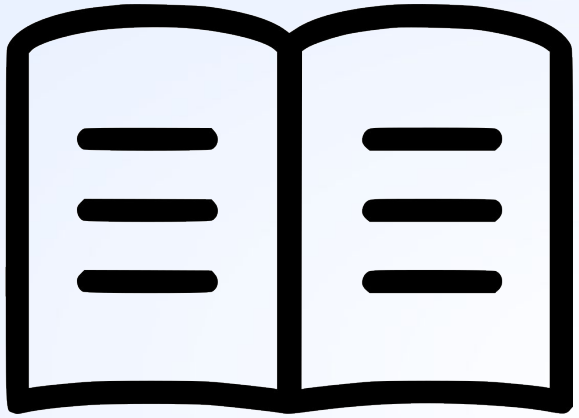
Justify and request for:

- Additional headcount for full-timers
- Temporary contractors
- Funding for outsourcing

Don’t do overtime

It’s unsustainable and leads to burnout

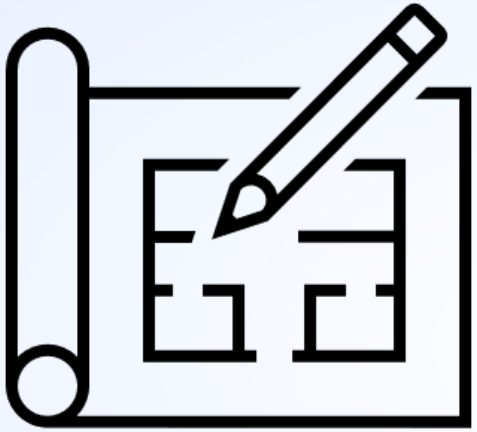
Recap



You now have:

- A framework for everything that comes at you
- You can plan ahead
- You communicate progress and pain points
- You’re (almost) prepared for delegation

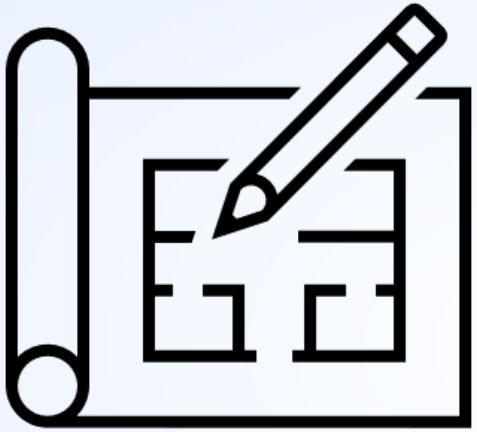
Onboarding



Prepare an onboarding doc with:

- Step-by-step workflows
- List of go-to people and their respective areas
- Overview of tribal knowledge
- Style guide to follow

Bonus: Integrate into the development cycle



- Educate your development team on:
  - Why docs are important
  - What a tech writer does?
  - How to work with a technical writer?

FAQ of a tech writer

1. What does the feature *do* and what it *doesn't do*?
2. Does the user understand all the *terms* that you used in UI?
3. Is the feature always *present / available*? If not, why?
4. What are *common mistakes* that may break it?
5. Any side effects?

Integrate into the development cycle

- Get engineering leads on board
- Learn to:
  - Read code and familiarize yourself with the code base
  - Understand “engineer-speak” and read commit logs
  - Contribute to the product directly, if you can

Delegation 101

1. Explain where you are, where you’re going, and how to get there

Delegation 101

2. Make your expectations clear

Delegation 101

3. Mentor your directs

Delegation 101

4. Delegate only the tasks you know how to do

Delegation 101

5. Agree on a timeline

Delegation 101

6. Reward and praise effort, not outcome

Delegation 101

7. Only carrots, no sticks (IMHO)

Delegation 101

8. Trust and let go

Delegation 101 recap

1. Explain where you are, where you're going, and how to get there
2. Make your expectations clear
3. Mentor your directs
4. Delegate only the tasks you know how to do
5. Agree on a timeline
6. Reward and praise effort, not outcome
7. Only carrots, no sticks (IMHO)
8. Trust and let go

Questions ?

Thank You!