

# Technical Documentation

Communicating Effectively & The Design Doc

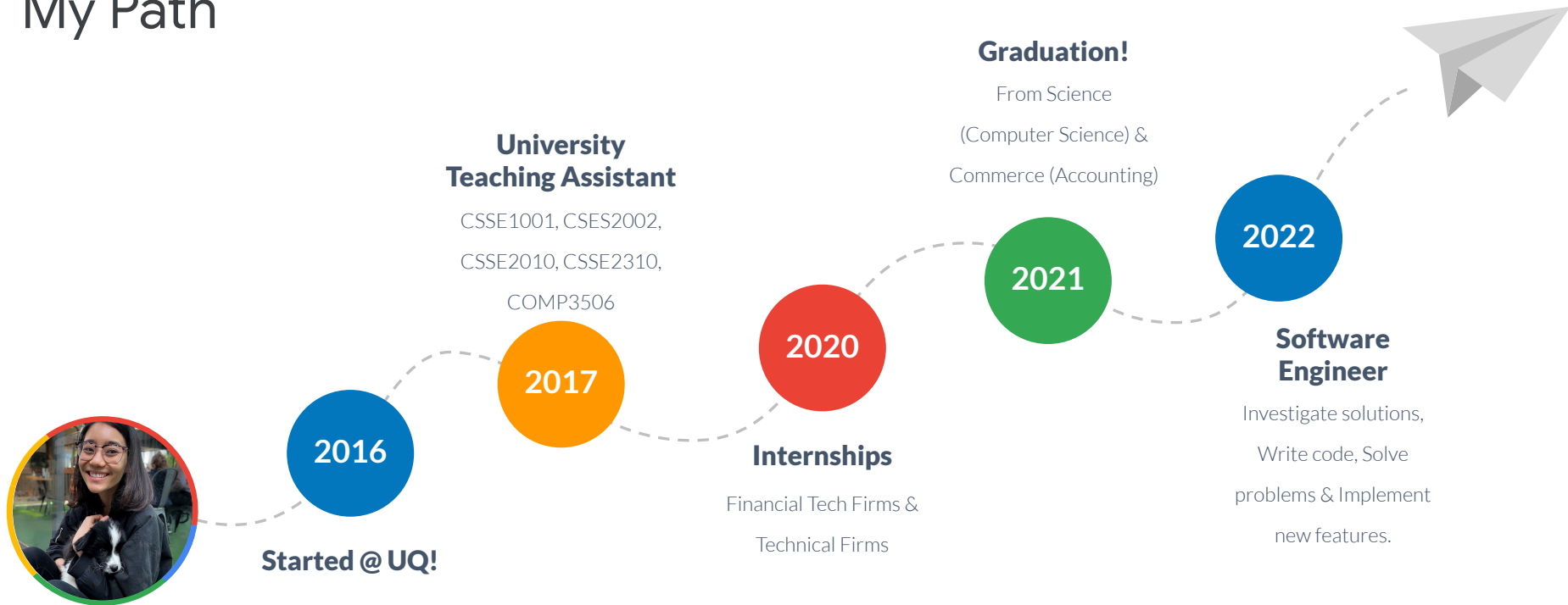


Anna Truffet



# Hello!

# My Path



# Technical Documentation

# Types of Technical Documentation

# Types of Technical Documentation

**Instructional, Authoritative  
& Backwards Looking**

# Types of Technical Documentation

## **Instructional, Authoritative & Backwards Looking**

- Architectural Records Design
- Wikis
- Read Me files
- Coding Labs

# Types of Technical Documentation

## **Instructional, Authoritative & Backwards Looking**

- Architectural Records Design
- Wikis
- Read Me files
- Coding Labs

## **Decision-making, Discussion-based & Forwards Looking**



# Types of Technical Documentation

## **Instructional, Authoritative & Backwards Looking**

- Architectural Records Design
- Wikis
- Read Me files
- Coding Labs

## **Decision-making, Discussion-based & Forwards Looking**

- Engineering Design Document
- Product Requirements Document
- Scoping Document
- Implementation Plan

# Types of Technical Documentation

## Instructional, Authoritative & Backwards Looking

- Architectural Records Design
- Wikis
- Read Me files
- Coding Labs

## Decision-making, Discussion-based & Forwards Looking

- **Engineering Design Document**
- Product Requirements Document
- Scoping Document
- Implementation Plan

# Engineering Design Docs

# What is an engineering design doc?

A document written by engineers,  
designing and discussing, with many,  
an upcoming feature's design and implications.

A **document** written by **engineers**,  
designing and discussing, with many,  
an upcoming feature's design and implications.

A document written by engineers,  
designing and **discussing, with many,**  
an upcoming feature's design and implications.

A document written by engineers,  
designing and discussing, with many,  
an upcoming **feature's design and implications.**



A document written by engineers,  
designing and discussing, with many,  
an **upcoming** feature's design and implications.

# An example

# A group example

- Task Overflow needs a new feature.
- It is your job to implement it, but...
  - There are several questions about the best way to implement it.
  - You need agreement with domain experts in Europe.
  - You are new to the code base.

## Add a new item

Title

Description

Choose a date

MM/DD/YYYY

☐ Give Brae a great staff evaluation

Apr 23, 2023

What a great guy

☒ Join Richard Thomas fan club

Feb 26, 2023

Join the CSSE6400 slack

# A real example

## Operation Agent's Tool: Determining the Display Order for Card Shown

ary + Confidential

[Document self link](#)

Username	Role	Status	Last Change
<a href="#">Person A</a>	Approver	PENDING	2023-03-20
<a href="#">Person B</a>	Approver	APPROVED	2023-03-24

**Status:** Draft > Current > Needs update > Obsolete

**Authors:** Anna Truffet

**Key Contributors & Collaborators:** Person A, Person C, Person D

### Objective

For Project Operation Agents Tool (Ops Tool), the [sorted display order of the cards need to be changed to accommodate for the increased number card types](#). The new order will include sorting based on who created the card. This one-ish page doc aims to determine the best location for the required sorting logic.

### Background (Optional)

Current Sorting



# Let's break that down

# What is an engineering design a doc?

- Explains the how the proposed feature will solve (at least most of) the hard problems
- Communicates any decisions
- Solicits feedback from other engineers & teams
- Gives sufficient detail that:
  - Others (without much other context) can understand and provide feedback
  - a knowledgeable engineer can build it... without making major design decisions

# What an engineering design doc is not

- A persuasive essay
- An implementation plan
- A product spec or a list of features
  - This is a product requirements doc! A product manager would have written this prior.
- a TODO list

# When do I write an engineering design doc?

- Not for small code changes

For features that may:

- Be **large** & affecting **significant** amounts of functionality
- Affect **other teams**
- Have **many unknowns**
- Have **several potential solutions**
- Need you to **provide information/explanation** for **future** readers



# The audience

# Who will read your engineering design doc?

The name is deceptive!

# Who will read your engineering design doc?

- Team members
- Future employees
- People who are affected by your change.
- Leadership (approvers)
- Implementers
- Security & privacy reviewers
- Product managers

Your audiences = multiple audiences!

# A recipe for beginners

## Pasta & Pine Nuts



### Ingredients (serves 2-4)

- 8 oz (225g) of pasta
- 2 tsp (12g) salt
- ½ cup (110g) of salted butter
- ¼ cup (35g) pine nuts
- ⅓ cup (30g) freshly grated Parmesan cheese
- Coarsely ground black pepper

Browned butter, pinenuts, and pasta combine for simple comfort food at its finest. The butter and pine nuts start and finish together with the pasta – in about 10 minutes. (If you've never browned butter before, read through this [tutorial](#) first.)

### Instructions

1. Add 8 cups (2 liters) of water to a pot. Place over high heat. Add salt.
2. When the water is boiling, add the pasta.
3. Add butter and pine nuts to a skillet over medium heat.
4. After the butter has melted (about 5 minutes), stir frequently so that the nuts cook evenly.
5. When the butter has browned (usually another 5 minutes), transfer the butter and nuts to a heatproof bowl (no plastic).
6. Drain the pasta in a colander and then add to the butter and pine nuts. Toss until the pasta is coated with butter.
7. Divide onto plates and top with Parmesan and pepper.

# A recipe for experienced cooks

Put a stick of butter and a handful of pine nuts in a skillet. Cook over medium heat until both are brown. Toss with cooked pasta, grated Parmesan and black pepper.

– #51 of [Summer Express: 101 Simple Meals Ready in 10 Minutes or Less](#)

Mark Bittman for the *New York Times*

What do your readers need to know? What are their goals?

# Writing a good engineering design doc



# A good engineering design doc

- Concise
- Clear
- Detailed enough, but not rambling
- Organized
- Not a wall of text! Diagrams, lists, links, memes

# You laugh but...

Option 1: Simple Opt-In



Option 2: Both Opt-in and Opt-out



**Docs and perf**

Obligatory memes: <http://meme/5288693555003392>, <http://meme/5594262439198720>

1. What does your reader need?

## 2. Keep it concise

### 3. Outline & Dot point first

## 4. Improve readability

## Example:

In order to well understand the information at hand about load balancers, it is important to understand the components of a load balancer. There are effectively three components of the digital load balancer: the first component is the listener which allows traffic to enter the Load Balancer and each listener has a port (e.g. port 80) and a protocol (e.g. HTTP) associated with it. The second component are Target Groups are groups of nodes which the load balancer can route to. Each target group has a protocol and a port associated with it, allowing us (the programmer) to switch ports on the way through the load balancer. This is useful if the targets are using a different port to the ports we want to expose. The 3rd component is the load balancer routes the traffic to the target groups based on rules that we set up.

## 4. Improve Readability

- Use lists if you list things
- Remove redundant sentences, words and phrases
- Consider your voice (active vs passive voice)
- Keep your sentences concise
- Use formatting



5. It is a tool in communication, but it is not the only tool

# Getting reviews

 Google Developer Student Clubs

# Google Developer Student Clubs (GDSC)

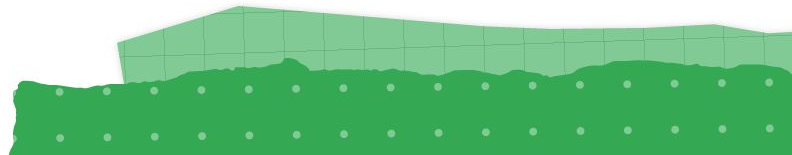
Learn more at [goo.gle/GDSC](https://goo.gle/GDSC)



# What are Google Developer Student Clubs?

Helping students bridge the gap between theory and practice

- Google Developer Student Clubs are **university based community groups** for students interested in Google developer technologies.
- Students from all undergraduate or graduate programs with an interest in growing as a developer are welcome.
- By joining a GDSC, students grow their knowledge in a peer-to-peer learning environment and build solutions for local businesses and their community.

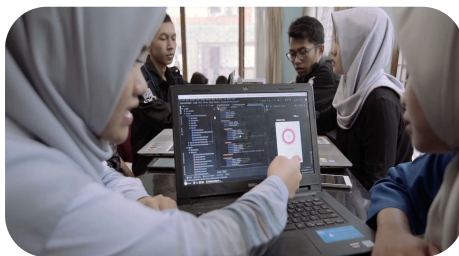


# What should you expect?



## Learn

You will learn Google technologies like (Android, Google Cloud, etc) together



## Build

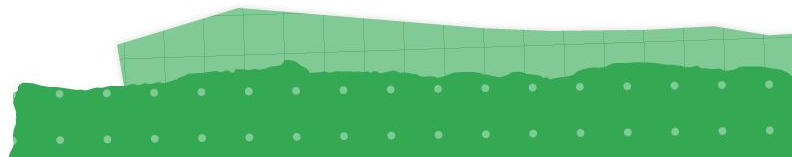
Create & build solutions for local problems through project-based code labs and content



## Connect

Share solutions through fun meetups, study jams and events like demo days

# Learn more!



# Writing Resources

- These [Overview of technical writing courses](#)
- Software Engineering at Google's [Documentation Chapter](#)
- “On writing well” by William Zinsser (a book about non-fiction writing styles)