

Habits, communication and documentation

Coding in Public workshop
2024



Table of Contents

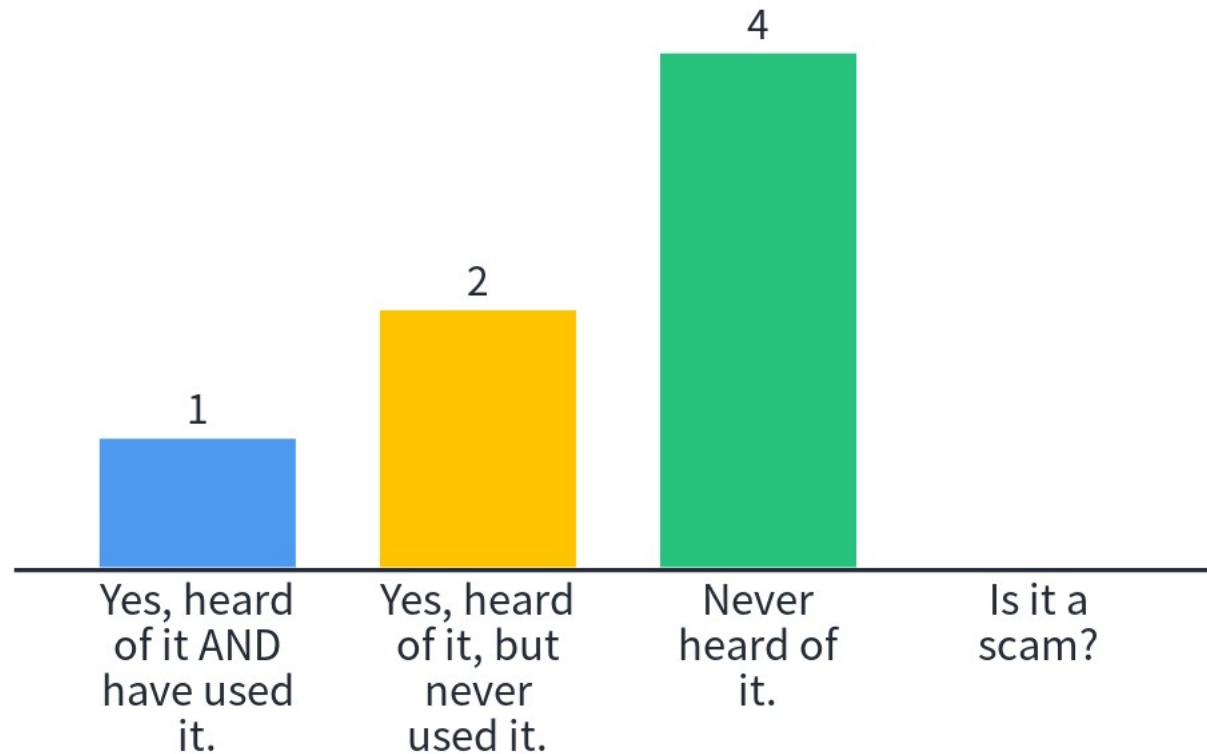
- Brains and Getting Things Done
- Collaboration and communication
- Documentation must match the purpose
- When to stop
- Questions?
- Breakout #2

Brains and Getting Things Done

Brains: good for having ideas, not keeping ideas



Have you heard of GTD?



1 - Capture

- Things you read/watch/listen to that you might want to refer to later
- Tweets/hashtags that seem useful
- People you meet that you might want to talk to/collaborate with/etc.
- Ideas you have
- Questions you want to answer
- Emails/chat messages/etc. where you discuss or ask things
- Et cetera

DO NOT:

- assume you'll remember
- evaluate its worth in the moment
- get embarrassed about capturing this thing

1- Capture tools

- Notebook by the bed
- Browser extension
- Camera + cloud
- Phone apps
- Dictaphone/smart board/tablet/QR notebook/etc.
- To-do lists
- Brainstorm/mind-mapping tools
- Meeting recordings/auto-notes
- Other

What tools have you used/heard of?

iPhone notes app, Word Docs (not the best method but old habits die hard)

Taking pics

notes app in phone

notion

NotionWhatssap

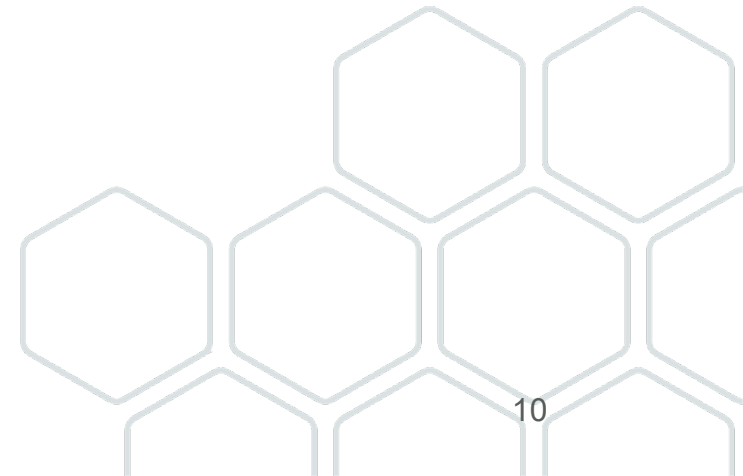
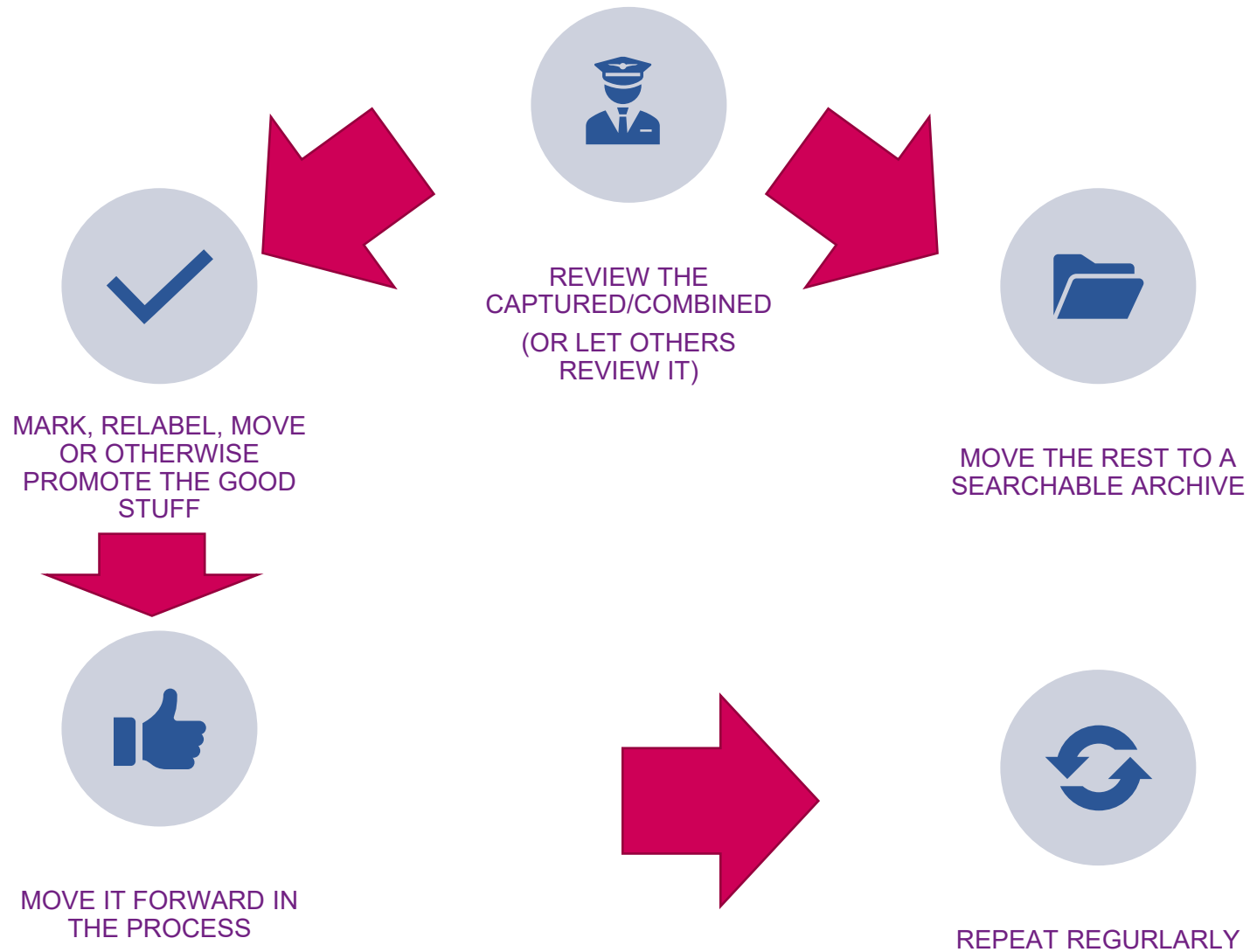
Notes, clicking pics, calendar

Notes and whatsapp

2 - Combine

- Schedule time to regularly combine your various sources
- Automate combining sources if possible
- Good processes for combining any remaining manual tasks
- Always look for better processes

3 - Sort



3 - Sort honestly

Schedule regular sorting time.

Read everything in your central place and ask yourself:

1. **Do you understand it?** If no, delete it.
2. If yes, ask yourself: **Will you use it in the next 6 months?**
 1. If no, archive it (searchable, not binned, low effort)
 2. If yes, sort it by
 - * Moving it to other places/folders/apps/etc.
 - * Tag/label/rename
 - * Link to other things (as appropriate)

Talk about your combine and/or sort processes

If using Microsoft apps,
then Loops

Check task list at the start
of every week and sort
through items, update
deadlines etc.

Weekend is for sorting.

Large to do list for
weekly and longer term

Always try weekly (Friday
for week coming) but often
the first thing that slips
with deadlines

Notion, check pending
tasks every week.

Go through handwritten notes
and make a to-do list in the
notebook, then make a new
to-do list when the
handwritten notes hide it.
Might not be the best process

Refine turns vague into actionable

- Questions to ask
- Clarifying hypotheses
- Steps to take
- Things that need to be learned/researched
- Resources to use
- People to talk to/get onboard
- Etc.

4 – Action

- If you can do it quickly (less than 2 minutes), just do it.
- If you can do it but not quickly, then schedule it to be done.
- If cannot do it without another thing, schedule both the things.

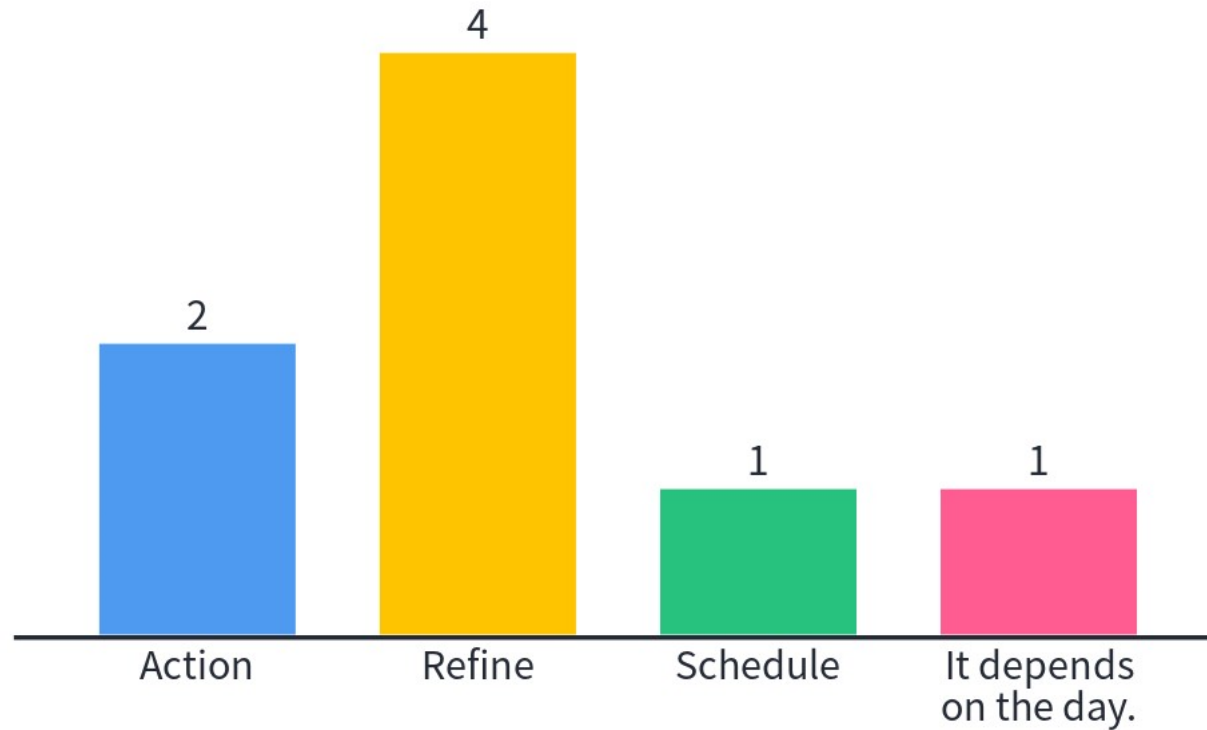
4 – Action and/or schedule

- If you can do it quickly (less than 2 minutes), just do it.
- If you can do it but not quickly, then schedule it to be done.
- If cannot do it without another thing, schedule both the things.

SCHEDULING IS AN ACTION

- What are the priority actions (necessary to do)
- What are the urgent actions (need to be done first)
- Who is doing what?
- When are they being done?
- Who or what is dependent on who/what?
- Who is a resource if things go wrong?

Which do you think is harder to do?



5 - Reflect

- What is working/not working at each step?
- What are you proud of?
- Do you need to do things differently? Action or schedule these
- Write the reflections down.
- Discuss them with people when you can.
- Reflect on the written reflections occasionally.

The steps are not always totally separate

- Collaborative work often is capture + combine + sort
- Writing the agenda and minutes are part of this
- Checking the minutes/agreeing the action items is too
- May need explicit follow ups to reflect on brainstorm outputs
- Put everything in a central, accessible place
- Make it easy for others (e.g. put a link to the doc in an email, mark the things you want them to check with their initials/in colour/etc.)
- Put sessions in comms plan/calendars/to-do lists/Trello boards/etc.

Doesn't that take a lot of time?

- Not as much as you think.
- It gets easier/faster with practice.
- If you protect this time in your calendar, others will respect it.
- This time is an investment in you.
- This time/effort saves a lot of panicked scramble later (where was that quote from?!?, etc.)
- You will know yourself better (is today a sort day or a refine day? A schedule day or a do day?) and act appropriately.

Incentives

- Attach new habits to existing habits
- Get help from others
- Pomodoro technique – 4 X 25 focus minutes + 5 break minutes
- Reward yourself for any steps done, tasks done, or pomodoros
- Alternate different kinds of tasks – creative tasks, detail tasks, menial tasks, physical tasks, etc.
- Consider your strengths, motivations and incentives when scheduling

What motivates or incentivises you?

Feeling productive and clearing the mental 'to do' list

Using a timer for an hour then having a break, and repeat. Hitting small research goals motivates me e.g. finish making that graph by the end of today.

Using a wide desktop screen motivates me to work more compared to my laptop.

Fun apps, built in templates. Clear checked off to-do lists w/ visualisations

Meeting free days

Document everything that allows me to see that I am making progress: pomodoro, number of words per week, graphs of task achieved, etc.

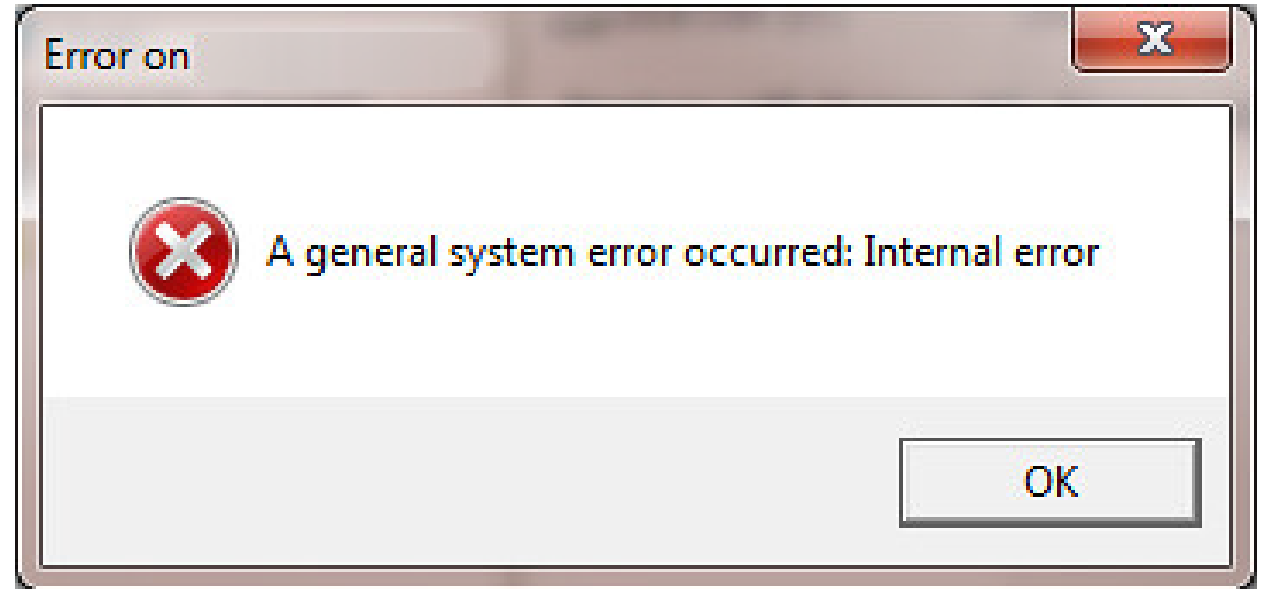
Stationary always helps, like the idea of chunk of time (25min + 5 min break).

To feel free
Rewards
Planners,
stickers

Collaboration and communication

Collaboration and communication blocks reproducible if:

- Things are not equally accessible
- Processes are not clear/documented
- Teams/people are siloed
- Decisions are not discussed, justified or documented
- Credit or contribution is not agreed or documented
- Timelines are not documented



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

Which of these has frustrated you in the past?

Not updated documentation, package versions incorrect. Files not accessible. Old version of software.

Unclear or non-existent documentation

Losing what was agreed in meetings for not documenting. Returning to the same discussions because of not documenting

Gave the wrong email address and had to wait weeks for another approval, it set the project back oopsie

Assume nothing

- Language
- Terminology
- Platforms
- Requirements
- Traditions
- Habits
- Naming conventions
- Storage locations
- Track changes?
- Etc.

Distribute, designate, delegate

- Distribute – divide work to all according to individual skills/responsibility/time
- Designate – Clarify the priority and urgency of each task
- Delegate – Trust the trustworthy to represent you



Use good tools

"FINAL".doc



FINAL.doc!



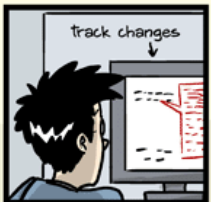
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments49.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc

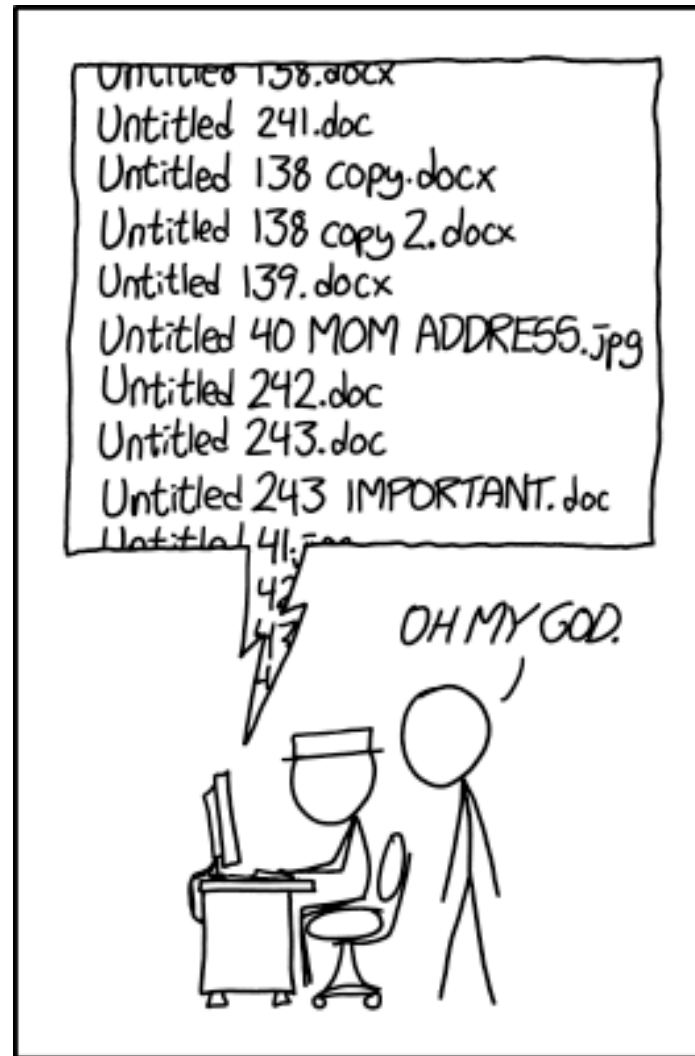


FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRAD SCHOOL?????.doc



JORGE CHAM © 2012

WWW.PHDCOMICS.COM



PRO TIP: NEVER LOOK IN SOMEONE ELSE'S DOCUMENTS FOLDER.

Reflect! Processes are never done

- Consider the legacy
- Sometimes things blow up unexpectedly (and much later)
- Take time to reflect on what was done/how/when/etc.
- Maintain contact with collaborators (if possible) to make it easier to reflect collaboratively
- Share what went well on the project as well as what went well in the process

Share some collab/comms wins and/or fails

Collaborating on a GitHub repo with colleagues and none of us deleted the whole repo by mistake (a win)!

Success: Using teams and share point

No examples, but communication is vital. Everyone communicates differently which makes it difficult but more important to get right.

Fails: code been overwritten before we used git. Win: using Jira & Confluence to manage tasks & documentation.

Gave the wrong email address and had to wait weeks for another approval, set the project back by weeks oops

Documentation must match
purpose

Documentation by you, for you

- Lots of your documentation may never be used/seen by others
- It is still useful to document because:
 - Can relieve your mind (out of my head, into my lists)
 - Create an evidence trail should there ever be some kind of formal complaint (always document things along this line)
 - Reassuring when you feel you've "done nothing"
 - Are useful to peruse when you go to brush up your CV
 - Lots of other reasons that may be quite personal

For you, by you

- Spend less time on how it looks, more time on how it works
- It is important that you feel it is easy (or pleasant?) to use this documentation system for your own benefit
- Make sure you know how to get things out of it safely if needed
- Make a legacy plan for it

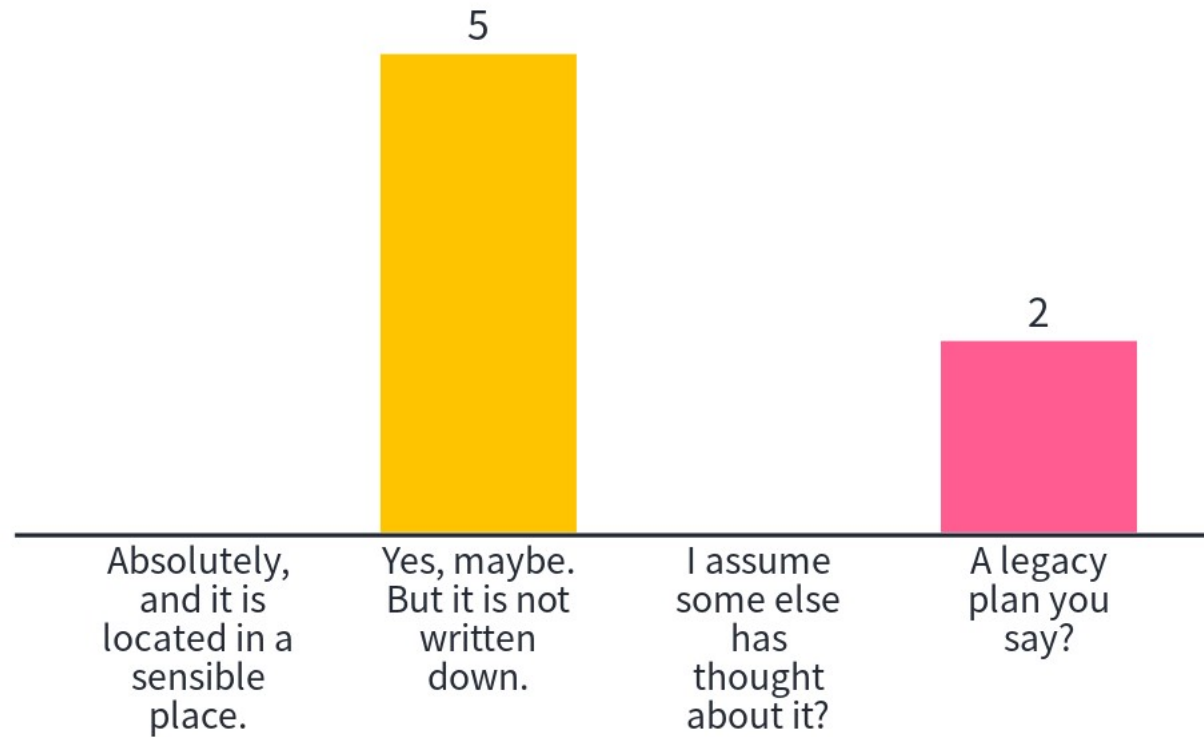
Documentation to stay in-team

- Lots of documentation is only seen/used briefly by your direct team
- That doesn't mean they are not useful to document because:
 - Necessary step to moving ideas forward to tangible products
 - Track credit/contribution for ideas to appropriate people/teams
 - Resource for sense check (especially after a holiday...)
 - Capture discussions that may turn into future work
 - Et cetera

Stay in-team

- Locations, conventions and responsibilities agreed
- Spend more time checking it is readable, works for everyone, that everyone is participating
- Make sure everyone knows what can and cannot be shared outside of the team if needed
- Needs a legacy plan to ensure everything works if key players leave or after project is done

Does your project have a legacy plan?



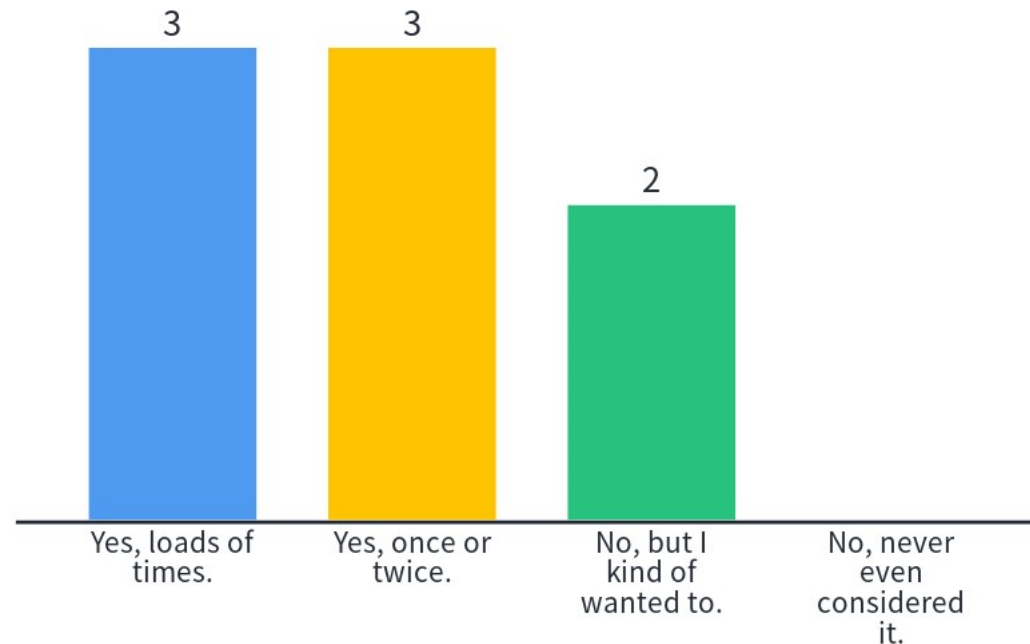
Documentation to share with the higher ups

- Fewer things will be seen/used with formal structures
- Ethics applications, funding bids, job extension requests, etc.
- You need to keep these as well as the supporting documentation that produced them:
 - Things in formal documentation are not always as they appear
 - Formal requirements for you can mean formal support/approval too!
 - Et cetera

Share with the higher ups

- Probably comes with specific forms, criteria, deadlines
- Always keep copies for yourself
- Ensure you are getting what you are owed as well as doing what you are required to do (formal agreements are always two-way streets)
- Maybe share more widely with/without anonymisation?

Have you ever requested a formal document from a colleague as an example?



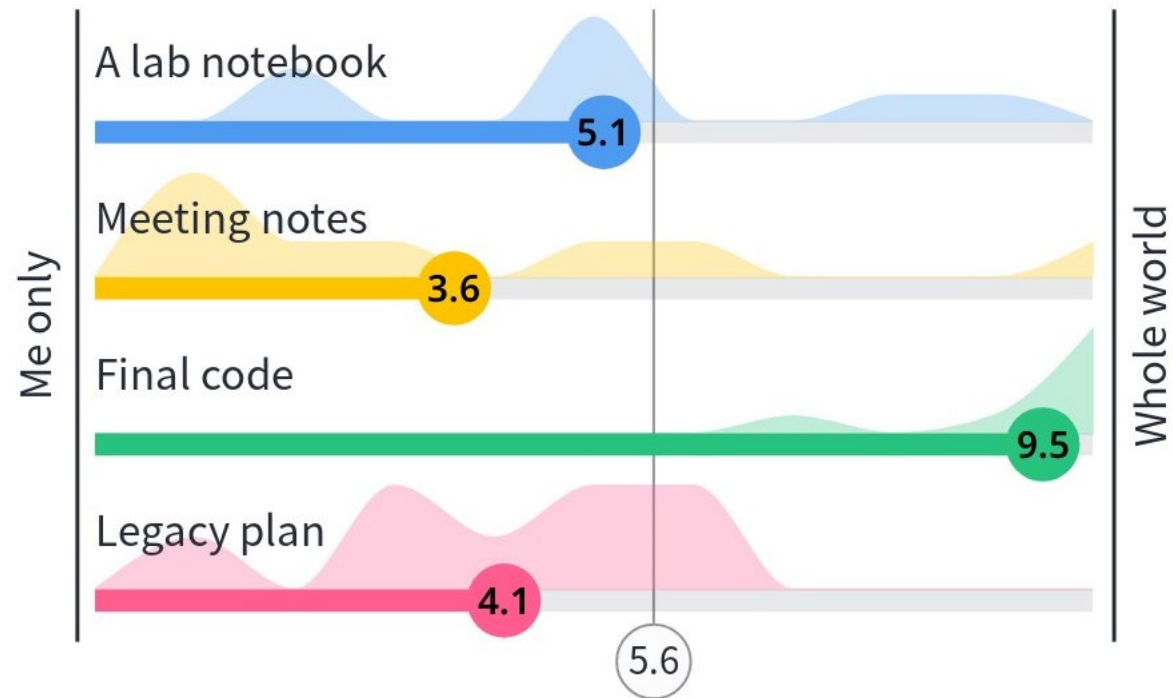
Documentation to share with other researchers

- Put it where everyone can see your article/chapter/output
- Final stage and working stages are different
- Legacy plan here too
- Not everyone will want to examine your code/data
- But you should still document as if they would:
 - As much end-to-end process as possible
 - As much data/code/analysis/other as possible
 - No one has ever told me my code was ugly
 - A couple of really techy people were genuinely amazed that I published my entire research project in an interactive notebook.
- Don't be embarrassed.

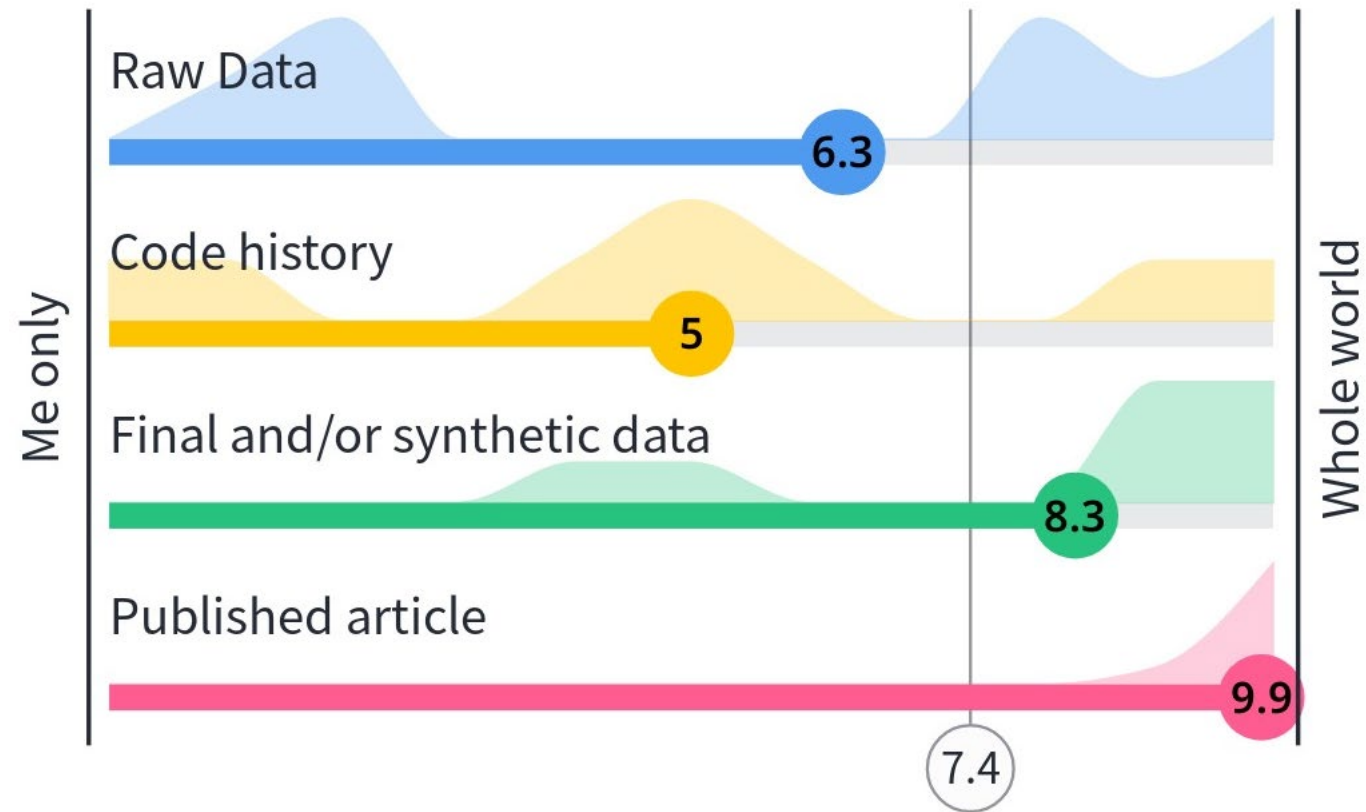
How to tell what kind of documentation it is?

- Is it necessary to reproduce the work *exactly* ?
- Is it helpful to reproduce the work in any way?
- Are there any restrictions against sharing it?
- Will it be useful to support you in some kind of claim/dispute?
- Is it a massive hassle to make it suitable for public consumption?
- Will it hurt you to share it?

What level of sharing is right for this? 1



What level of sharing is right for this? 2



When to stop

Keep everything that you can keep



Ideas, agenda, minutes, brainstorming, sources, code environments, etc.



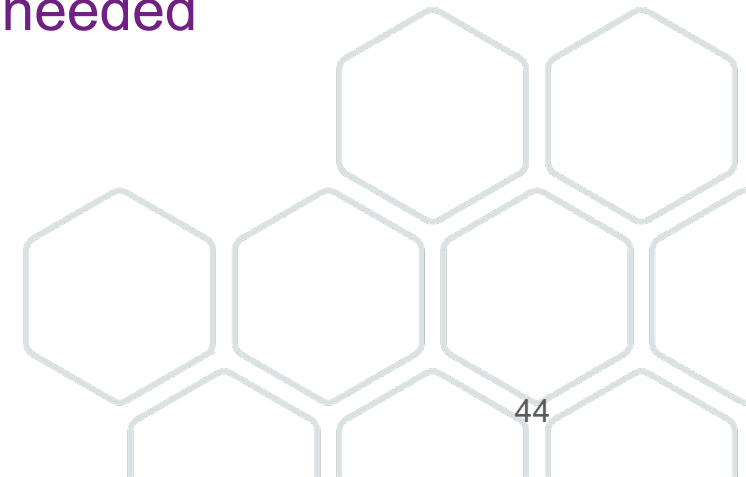
To document the process, work, decisions, actions, etc.



More effort while creating/using them = more useful later



Anonymize or store securely if needed



When to stop documenting

- Do not document/share anything that you do not have a right to document/share
- E.G. ask to record interviews, ask if people are happy to be listed as corresponding author, etc.

Also, stop documenting when...

You are spending more time on the documentation than on the actual work.

If you find the documentation to workflow balance is off, try:

- Automating the boring stuff
- Using better tools/processes
- Scaling back/rescoping the project
- Recruiting help for sanity checks
- Other suggestions?

Any more ideas about documentation?

We test our process docs with someone outside project or less involved to check fit for purpose.

Its hard to document as you go, especially when trial-and-erroring code, any tips?

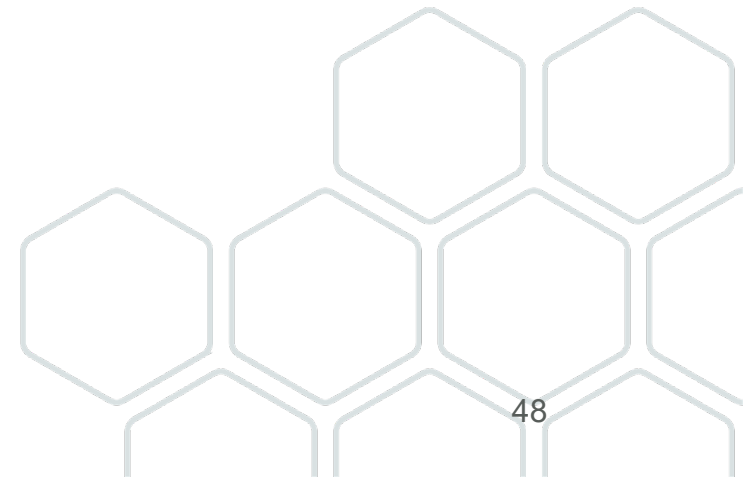
Appropriate folderization of documentation.

If applicable, indicate versioning

Clear code comments help when it comes to writing your methodology, especially when you look back and think "What does that function do?"

References

- Falk Huettig, Régine Kolinsky & Thomas Lachmann (2018) The culturally coopted brain: how literacy affects the human mind, *Language, Cognition and Neuroscience*, 33:3, 275-277, <https://doi.org/10.1080/23273798.2018.1425803>
- Demoulin, C., Kolinsky, R. Does learning to read shape verbal working memory?. *Psychon Bull Rev* 23, 703–722 (2016). <https://doi.org/10.3758/s13423-015-0956-7>
- Wikipedia community. (2022). "Getting Things Done." Retrieved 21/03/2022, from https://en.wikipedia.org/wiki/Getting_Things_Done.
- Wikipedia community. (2022). "Pomodoro Technique." Retrieved 02/05/202, from https://en.wikipedia.org/wiki/Pomodoro_Technique.



Q&A

Breakout #2