Do Docs Better
(or, how to deliver value to your business through better documentation)

Riona MacNamara, Google
rimonam@
Documentation is core engineering work.
Documentation is as fundamental to software engineering as testing.
Fig1. - Problems encountered in open source

Source: opensourcesurvey.org

Incomplete or confusing documentation
Unresponsiveness
Dismissive responses
Conflict

VI. Challenges At Work

Stack Overflow survey, 2017

Unrealistic Expectations 34.9%
Poor Documentation 34.7%
Unspecific Requirements 33.5%
Inefficient Development Processes 30.3%
Fragile Code Base 29.6%
Changing Requirements 28.1%
Outdated Technologies 24.8%
Documentation is everybody's problem, but nobody's job.
Documentation is expensive.
Before we can do docs better, we need a **vocabulary** to describe documentation quality.
Structural quality: What does this doc look like?

- Are spelling and grammar correct?
- Does it comply with style and usage guidelines?
- Does it use proper voice and tone?
- Is it well-organized?
- Is it easy to navigate?
Functional quality: Is this doc effective?

- Does it **do what it's supposed to do?**
High structural quality + Low functional quality = Poor overall quality
Okay structural quality + High functional quality = Good overall quality
To deliver value, functional quality must be our primary goal.
1. Establish functional requirements:
Core SRE functions

- Monitoring and metrics
- Emergency response
- Capacity planning
- Service turn-up and turn-down
- Change management
- Performance
Docs that support core SRE functions:

- Service overviews
- Playbooks
- Postmortems
- Policies
- SLAs
Functional reqs: Playbooks

- Does it provide 100% coverage of alerts?
- Can the team rely on it to perform oncall duties?
- Is the playbook reliable (highly available)?
- Is it easy to create and update entries?
- Is each alert description accurate and complete?
- Does each entry give enough information to understand and resolve the alert?
- Does the entry give guidance on escalation?
2. Execute on those requirements! (Write the docs)
Doing docs better requires changes to tooling and culture.
Documentation will never be part of engineering culture until it is integrated into our codebase and workflow.
Meet engineers where they are

Keep documentation in source, next to its associated code.

- Simple Markdown easily readable in source
- Docs rendered at a predictable URL
- Engineers maintain docs as part of their regular workflow, using their standard toolset
Automate what you can

- **EngPlay**: Playbook server
- Alert manager integration
- Support for variables: doc pages adjusted with information from monitoring, such as the ID of the job that generated the alert, and ID of affected cluster
Better > Best

Set realistic standards for quality

Perfection is not the goal

Set an appropriate quality bar

What matters: Key information exists and is clearly conveyed
Use your power

Require docs as part of code review or launch

Rule of thumb: If an SRE, SWE, or user needs to change behavior after a change, the change should include docs.

Docs updates scale with the size of the change.

If it's doc'd, users can access it without bothering you.
Delete old docs!

Obsolete docs are positively harmful

Like code, every line of docs should serve a purpose.

A small set of fresh and accurate docs is better than a large assembly of "documentation" in various states of disrepair.
Recognize and reward docs

Documentation is engineering work!
Measuring and demonstrating business value:
Focus on functional quality

Gather data that demonstrates the quality, effectiveness, and business value of your documentation

- Measurable success
- User behavior
- Sentiment data
Goal: Decrease time for a new engineer to go oncall.

- I pitched a proposal to create a playbook with complete documentation for all alerts.
- I worked with the engineers to revamp the playbook so that each entry clearly conveyed what the alert meant and provided ways to address and mitigate any negative effect.
- As a result, the playbook received 5X more visits.
- Engineers reported that they could rely on the playbook during oncall.
- A follow-up study indicated that there was an x% decrease in the time it takes for a new engineer to go for oncall.
Share the data!

- With other SREs and SWEs
- With managers
- At promo and perf
- With partner teams
Call to action: Do docs better
Call to action:

- Focus on **functional** requirements and quality
- Reduce the **cultural and tooling barriers** to creating and maintaining documentation
- Gather data that demonstrates the **business value** of your docs
- **Use** that data to advocate for documentation
Doc work is core engineering work that is worthy of support, funding, recognition, and reward.
Questions?