



# Documentation is King

2013

3 min read • 614 words

---

As a member of the development community, I have the unique opportunity to take part of a powerful social movement, creating the tools that are fundamentally changing the world around us. Open source has found itself to be the crux of my professional, personal, and social life. There's something special about it.

However, it's also not the only activity that I give my attention to. I'm also [a photographer](#). My camera never leaves my side.

It's crucial to have an analog balance in our digital world. No digital interface will ever replicate the feeling of a manually focused rangefinder camera,

Rangefinder cameras like the Leica M series use a separate optical viewfinder with a focusing mechanism that requires manual alignment of two images—a tactile, meditative process that contrasts sharply with digital autofocus systems.

or the attention to detail and craftsmanship of a mechanical watch. It's all about context and perspective.

## Constraints

During my creative ventures in photography, music, and design, I've noticed a recurring pattern — I need constraints.

More and more, I've found that the simpler the tool is, the more effective it is as a channel of creativity.

A synthesizer with fewer knobs, options, and features allows me to focus on what's truly of value: making music.

### A prime lens

Prime lenses have a fixed focal length (unlike zoom lenses), which Henri Cartier-Bresson famously championed. This constraint forces photographers to "zoom with their feet" and often results in more thoughtful composition.

forces me to concentrate on capturing the moment as-is, rather than modifying my perspective (e.g. zooming). This forces me to take much better photos. Code is a creative craft, as is photography. The same principles can be directly applied.

## Internal vs. External

If you survey a large number of successful, thriving open source projects out there, you'll notice a recurring pattern. Great open source projects often share some common goals:

- Solve a clear problem.
- Have great documentation.
- Manage expectations realistically.

They communicate well.

I've unfortunately noticed another trend, found in many internal codebases I've worked with:

- Technical debt.
- Little to no documentation
- Tight coupling of components.

Look closely at these problems, and the root cause will be clear: communication is lacking. No open source project would ever be successful if it was run this way, so why should your company? The same principles can be directly applied to any project, open source or not.

# Documentation is King

Documentation leads to better code.

My code improves far more from writing documentation than writing tests.

Documentation forces you to approach your APIs symmetrically.

This reflects the principle of "documentation-driven development," where writing documentation first helps clarify design decisions and API usability before implementation, similar to test-driven development.

When I'm writing and maintaining robust documentation for a project, it's reflected in the code and API. I'm not building software for myself as an engineer, I'm designing an API and experience for myself and others as developers.

An open source project wouldn't be sustainable if it didn't have great documentation. Neither is your internal code.

Every design decision should be documented. Imagine not having to have tap your coworkers on the shoulder when you're working on an unfamiliar part of the codebase, or on-boarding a new employee. Imagine being able to make the change, run the tests, and push to production without questioning yourself, because the process was documented — or better yet, automated.

## Conclusion

Treat your [teams like open teams](#) and [your codebases like open codebases](#).

This philosophy, championed by GitHub's co-founder Tom Preston-Werner, suggests that the transparency and quality standards of open source development should be applied to internal company projects.

All of these communication problems will naturally erode because they don't have a choice.

Constrain yourself.