**Monica Powell**

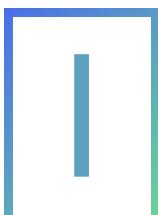
SENIOR SOFTWARE ENGINEER, NEWSELA



The **ReadME Project** amplifies the voices of the open source community: the maintainers, developers, and teams whose contributions move the world forward every day.



ALSO IN GUIDES

[Using code as documentation to save time and share context](#)

I learned early in my developer journey that teaching others is an effective way to quickly deepen my understanding of a new concept or technology. I've found that needing to articulate a particular concept to others causes me to revisit my assumptions and leads me to do additional research to fill any knowledge gaps. This realization helped prepare me to lead my first technical workshop at a conference before becoming a full-time software engineer. While preparing the curriculum for the technical workshop I naturally began to synthesize my learnings into a compelling curriculum that could be delivered within an afternoon.

Giving technical talks, creating videos, and writing articles and tutorials are all great ways to deepen your understanding of a technology and grow as a developer. In this article I'll guide you through how to get started publicly documenting your learnings. In additional articles in this series, I'll cover various forms of personal documentation—including note-taking and what I call “shine documents”—as well as context-specific documentation such as code comments, templates, and config files. By the end, you'll have a wide variety of tools and techniques to document and share your discoveries, decisions, and accomplishments.

Teaching can be a superpower

“To teach is to learn twice.” – Joseph Joubert

Teaching brings superpowers that span beyond software development. Numerous studies have shown a psychological phenomena, known as the Protégé Effect, in which learners who teach or prepare to teach others gain a deeper understanding of the material in the process. A [study published in Memory & Cognition in 2014](#) determined that just the expectation that one will need to teach can enhance one's learning and organization of knowledge.

One of the things I appreciate about using teaching as a learning strategy is that regardless of where someone is in their learning process, they can benefit from teaching others by publicly documenting some of their learnings and taking a documentation-driven approach to accelerating their learning and growth. A key component of documentation-driven growth is taking advantage of the protégé effect by learning in anticipation that you will be sharing some of your learnings and decisions with others.

Presenting Learnings

Sharing learnings doesn't have to be restricted to written documentation or tied to a specific format. In order to benefit from learning in public and teaching others in the process, it's more important to get into the habit of consistently documenting and sharing your learnings than using a particular format. Depending on how someone learns best, certain formats may naturally work better than others. Individuals may gain more visibility within the tech community and even uncover new opportunities if they consistently share their knowledge in public and develop a sense of authority within a given domain.

Some formats I've seen people use to share their learnings include Twitter threads, sketch notes, articles, YouTube videos, Twitch live streams, newsletters, participating in GitHub Discussions, answering questions on Stack Overflow, teaching a workshop, and speaking at a conference. I especially enjoy learning from engaging visual formats like [refactoring UI Twitter threads](#) with helpful design tips by TailwindCSS creator [@adamwathan](#) and [@steveschoger](#), or [sketchnotes](#) like those of Nitya Narasimhan, a Cloud Advocate at Microsoft who has some [great examples of sketchnotes on her Twitter](#). If you're interested in getting started with sketchnoting I recommend checking out [Let's Sketch Tech](#)—a community around sketchnoting in tech that hosts workshops and conferences.

What if I have nothing to share?

can get lost over time. Different authors often present the same technical topic from different perspectives so there is still something unique that you can offer by sharing your own point of view. Just as my writing resonates more with me, my writing may also resonate with certain individuals more than articles written by someone else that cover similar topics.

Write what you would have appreciated three months ago or something that you can later use as a reference. If others find value in what you shared, that's great. If not, that is okay too—you've still created an artifact that you can reference in the future. Instead of searching for things to learn with the intent of sharing learnings, I'd recommend sharing learnings from what you are already inclined to learn as you tackle new challenges. The learnings I tend to share publicly generally involve a lightbulb moment where I learn something that I wish I knew sooner. If you're regularly contributing to software then you will likely have your own "I wish I knew that sooner" moments as you explore new areas and tackle novel problems. Depending on your experience levels, the scope of these learnings may be larger or smaller. But regardless of size, sharing them with others is invaluable.

Local tech groups sometimes host events to help attendees generate ideas for solid conference talks. The [Global Diversity CFP Day](#) community and [Write/Speak/Code](#) are both invaluable organizations creating professional development programming to help increase the representation of people of marginalized identities who do public speaking within the tech industry. Often conferences will have a Call for Papers/Proposals (CFP) to gather speakers. Some websites where you can find out about upcoming CFPs are [PaperCall](#) and [CFP Land](#). Often information regarding the process to submit talks to a particular conference is announced on that conference's website and social media accounts. You usually do not need to have a fully-fleshed-out talk before you can submit CFPs as there will be some time between your CFP being accepted and you presenting the content. To submit a proposal you generally need a working title, abstract, and clear summary of the expected takeaways for the audience.

Share Your Learnings at a Sustainable Pace

Sharing what you learn in public does not have to mean committing to a huge series, writing 20-minute long reads, creating a 90-minute video, or giving a 30-minute presentation. You can—and probably should—start smaller. I encourage folks to start with something that they can commit to. One of the main takeaways from [Atomic Habits](#) by James Clear is that, over time, it's generally more impactful to make small, consistent progress towards an identity-based goal (like being someone who runs regularly) than inconsistently focusing on one huge goal (like running a marathon). Over time, these consistent habits bring you much closer to a more sustainable version of yourself that is aligned with your goals.

While having a personal blog is not a prerequisite for sharing your learnings in public I do think there is value in having an independent site where you can make up your own rules about how you share your learnings. Joel Hook's [article on treating your website like a digital garden](#) shifted how I view maintaining my blog. He describes a digital garden as a place where different pieces of writing can be shared at different stages of growth, emphasizing "process, care, and craft" over following the traditional blogging conventions like displaying recent posts in chronological order, etc. This digital garden mindset can be helpful for getting comfortable with sharing more content without getting too caught up in details and never being quite ready to hit publish.

Getting into the process and mindset of creating these tangible artifacts to document where you are in your learning journey is more impactful than only sharing a much smaller fraction of information in hopes of "polishing" the rest.

Over time your presentation of information will become more refined and it will likely become easier for you to synthesize your thoughts publicly. The growth that comes with regularly sharing with others can deepen your technical knowledge and enhance your communication skills when explaining new concepts to others.

In the next article in this series we will focus more on the process of documenting learnings in order to build an internal database of information that is tailored to your individual learning style and can be quickly referenced and re-used.

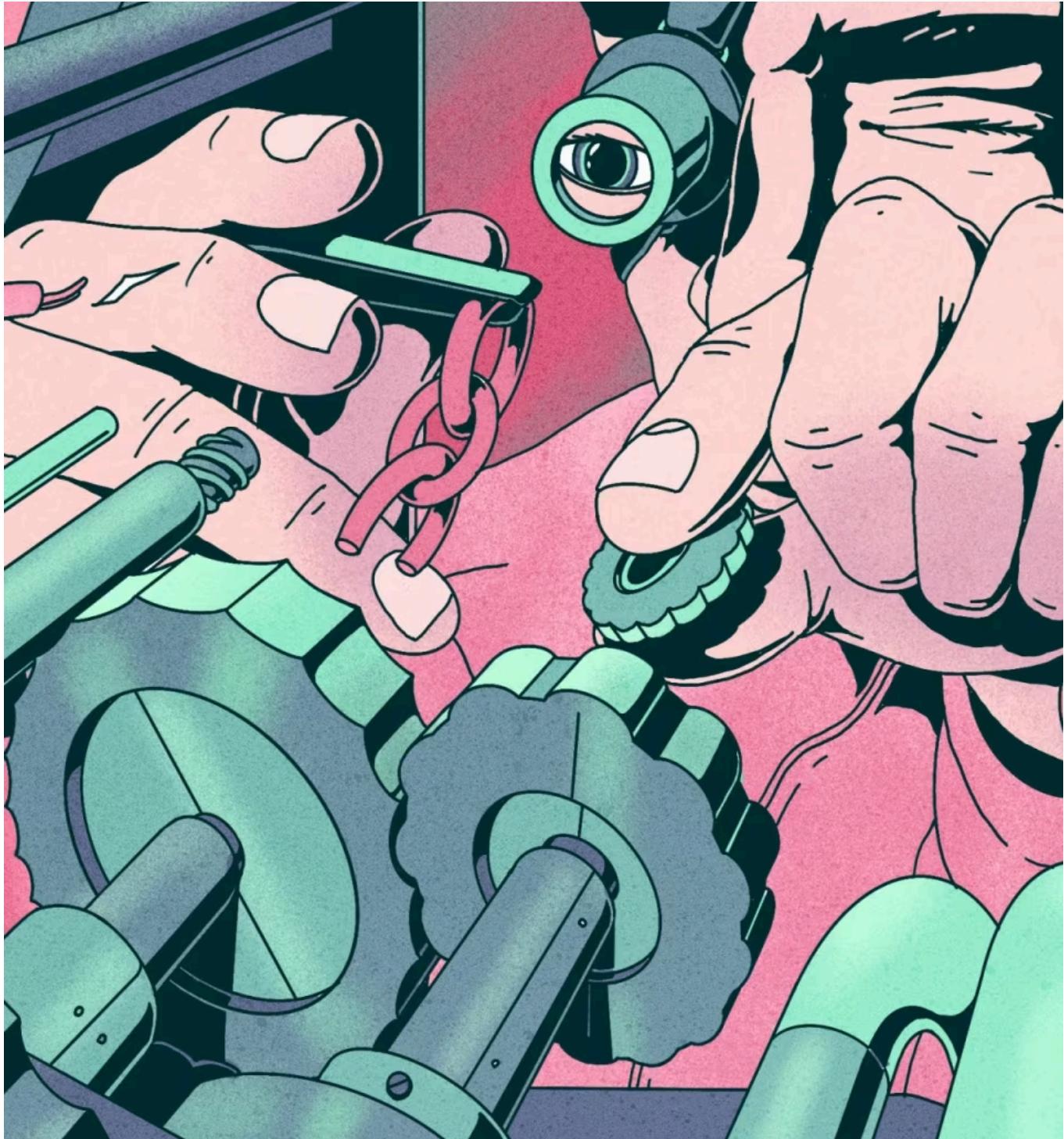
And if you want to learn more about this topic, check out my other articles from this series focusing on document-driven growth:

- Part 1: [Using code as documentation to save time and share context](#)
- Part 3: [Your future self will thank you: building your personal documentation](#)
- Part 4: [Brag now, remember later: Document your accomplishments](#)



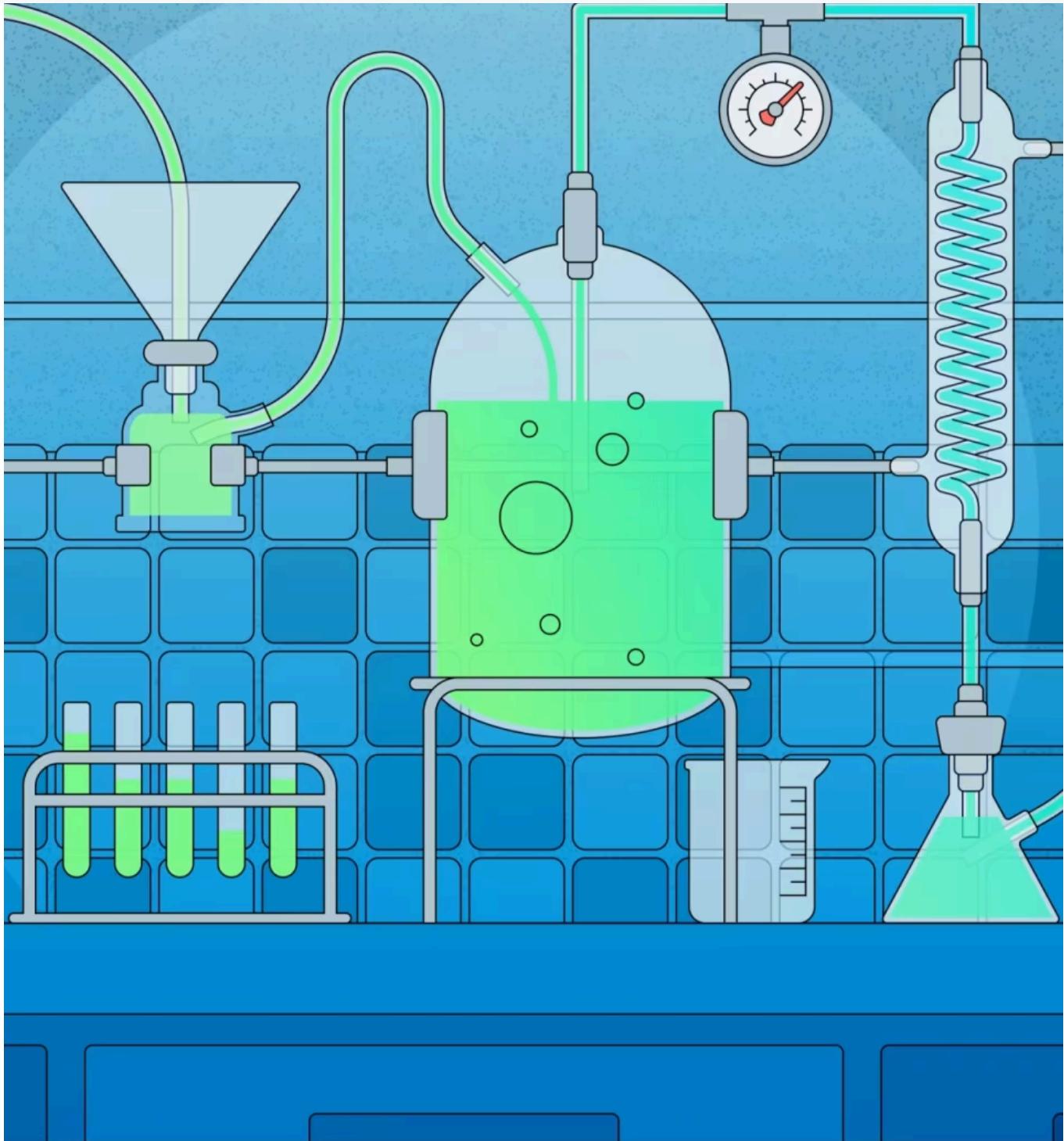
Hi, I'm Monica! I'm passionate about making open source more approachable, elevating people through technology, and cultivating communities. I enjoy working with [ReactJS](#), [Jamstack](#), JavaScript, APIs + Markup, and [GraphQL](#). I built [React Ladies](#) as a safe space for women and non-binary ReactJS developers to create and collaborate together. I like contributing directly to the open source software I'm using and also enjoy creating supplemental educational materials [on my website](#). I'm also a [GitHub Star](#) ⭐.

More stories



A beginner's guide to running and managing custom CodeQL queries

DENYS LASHCHEVSKYI // BETSSON



Provisioning self-hosted GitHub Actions runners on demand

NIEK PALM // PHILIPS



Treat accessibility issues as bugs, not feature requests

MIKE GIFFORD // CIVICACTIONS

About The ReadME Project

Coding is usually seen as a solitary activity, but it's actually the world's largest community effort led by open source maintainers, contributors, and teams. These unsung heroes put in long hours to build software, fix issues, field questions, and manage communities.

The ReadME Project is part of GitHub's ongoing effort to amplify the voices of the developer community. It's an evolving space to engage with the community and explore the stories, challenges, technology, and culture that surround the world of open source.

FOLLOW US: [X](#) [f](#) [o](#) [in](#) [r](#)

NOMINATE A DEVELOPER >

Nominate inspiring developers and projects you think we should feature in The ReadME Project.

SUPPORT THE COMMUNITY >

Recognize developers working behind the scenes and help open source projects get the resources they need.

SIGN UP FOR NEWSLETTER

Every month we'll share new articles from The ReadME Project, episodes of The ReadME Podcast, and other great developer content from around the community.

Email Address *

you@company.com

[SUBSCRIBE](#)