

# The Open Source Way 2.0

The guidebook for community management best practices

Thank you for joining me here today.

I'm here to talk with you about the latest version of the Open Source Way, the guidebook for community management best practices.

My name is Karsten Wade, I'm a community architect in Red Hat's Open Source Program Office. Although I served as project lead for the 2.0 release, I am otherwise one of many writers, editors, community experts, and open source practitioners who worked on this guidebook. You can find a full list of contributors at the end of the guidebook, which I'll be showing you in a few minutes.

## An opinionated guidebook



This is a guidebook, a way of getting things done following a collection of best—or at least good enough—practices. It is a method you can learn and evolve, built on understanding not only the *what* and *how* of a practice, but also the *why*.

- **A repeatable method** defined and refined by a community of diverse practitioners from all experience levels ...
- To **attract users**, who are the many individuals and organizations consuming your software, ...
- And **enable participants** to care and get involved ...
- In order to **grow contributors** of all types and levels, arising from the users and participants by accident or purpose.

This is an opinionated guidebook.

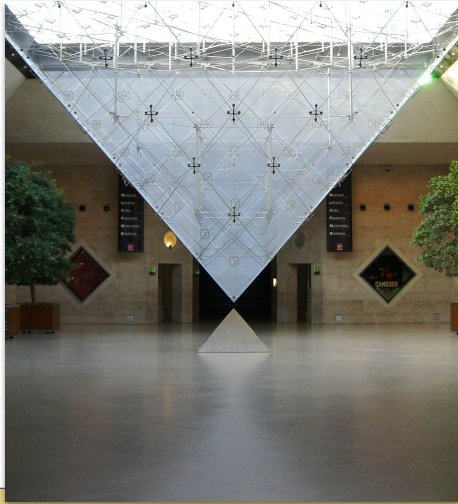
But frankly, much of the really good guidance and practices around creating and sustaining an open source project is fairly well known by experienced practitioners. Not a lot of secret sauce here.

The impetus for this project came out of a number of discussions over recent years in response to the ongoing growth of new communities and people who are new to open source in general.

The consensus is, there is a need for a guidebook written by a community of practitioners for all open source practitioners, bringing a diverse range of experiences but having a common understanding of what makes a practice "best."

There are many ways to share knowledge and learn. This guidebook by this community is for the reader and practitioner who enjoys, prefers, or learns best from and among a group of like-minded people. People who like to form as a community of fellow learners and practitioners.

This guidebook is centered on community management best practices. It allows you to start from the very beginning, to consider and engage with the open source way from the very beginning of your idea for an open source project. And it carries you through the creating to the sustaining of your project, from attracting users to when they pass your scant barriers to become contributors.



## Method to create & sustain

Steps in this method map to the sections in the guidebook (*Getting Started, Attracting Users, Guiding Participants, Growing Contributors, Measuring Success*)

1. Focus on understanding and attracting users.
2. Make sure when those potential users arrive on your webpage, there is something for them to do right away.
3. As people try out and become users, make it easy for them to participate in whatever they find.
4. Move the contribution moment to the left with an inclusive and welcoming community.
5. Understand what is important to sustain your community, and measure and track that.

Before jumping in the guidebook and showing you are around, let's take a moment and review the overall community growth method in the guidebook. The method is generally mapped to the sections of the guidebook:

Getting Started > Attracting Users > Guiding Participants > Growing Contributors > Measuring Success

When starting, you want to do the harder work of being completely open and transparent from the beginning, even when there is no one there watching you be open and transparent, putting up the infrastructure of participation with barriers appropriately lowered for users and contributors. All while focusing on building something for your users.

Make sure when those potential users arrive on your webpage, at the top of the webpage is a short description of the project's software. It should include a purpose that users can understand and imagine. It should give them something to do right there, a call to action—download and try out, login and try out, a quickstart to use, and so forth.

Making software and content your users need is one of the main purposes of an open source project. It out of those users that most contributors arise, from the enthusiastic advocates all the way to the most ardent contributors.

As people try out and become users, and explore the world around your software, make it easy for them to participate in whatever they find. Make information easy to find and forum access straightforward. Make it easy for enthusiasts to share what they know about your project.

The idea that contributors come from users was true when open source was characterized by the hobbyist, and it is still true today when users of your software now include companies making products out of it and corporate foundations wanting a common platform for competitors to share.

So whether your users are a broad base of individuals or small organizations, or your users are some of the largest corporations on the planet, the conceptual math still applies. Of an entire corporate participant in your project, only a handful of people will contribute directly to the project in some fashion. Many others will contribute indirectly in their work supporting the overall user-organization's contributions.

For example, that is how I earned my seat on the inaugural CentOS Governing Board, because of the work I did at Red Hat behind the scenes to bring about the new relationship between Red Hat and the CentOS Project.

So that's it: the bigger your user base, the bigger the pool of potential contributors.

Open source projects are a type of community of practice, which is a group that gathers to share, learn, and improve practices in as particular domain they're interested in. In the science that studies communities of practice, there is the idea of "Legitimate Peripheral Participation", which describes how newcomers become experienced and eventually old timers in a CoP or collaborative project. From the Wikipedia page:

*"According to Legitimate Peripheral Participation, newcomers become members of a community initially by participating in simple and low-risk tasks*

*that are nonetheless productive and necessary and further the goals of the community. Through peripheral activities, novices become acquainted with the tasks, vocabulary, and organizing principles of the community's practitioners. Gradually, as newcomers become old timers and gain a recognized level of mastery, their participation takes forms that are more and more central to the functioning of the community."*

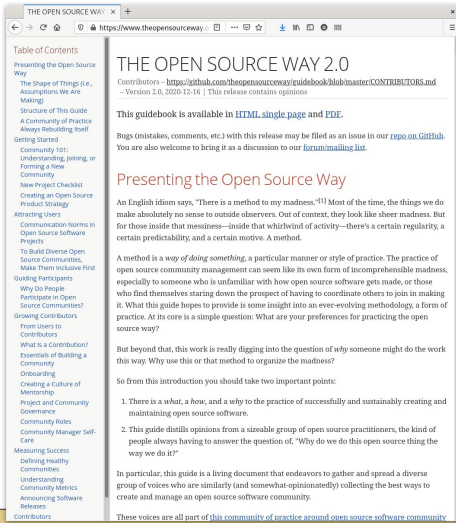
One important take away from this is that contributors don't come paragliding in from nowhere, they are made from within the project. And while it is true that people with needed skills and expertise do come directly to projects to contribute, the process by which they arrived will have followed the Legitimate Peripheral Participation path in some way. In the corporate contributor example, there will have been some period of precursory activity from the sponsoring organization and so forth that precedes the uplift to contributor. Once arrived, that person will still have to learn all the ways and means of the community before they can become an experienced member, regardless of incoming cachet.

An important goal for a project is to be sustainable. This is about more than the economics; although that is an important part of the conversation, all the money in the world isn't going to make a toxic community healthy again. A sustainable open source community is one that is able to take in new contributors and promote them to leaders, without the stepping-back of "old timers" meaning the breakdown of systems. A sustainable community can have key maintainers able to go on long vacations or choose to step back from the project without scrambling and struggling.

So in choosing what to measure for in your community, maintain a holistic viewpoint, and don't get stuck on simple first-order data like "average time to close a bug report" as the only thing worth watching and caring about.



## A look through the guidebook



There are at least several ways to read and use the guidebook:

1. Read from beginning to end.
2. Start anywhere, follow your interests and skip around.
3. Bring a problem or question for a specific answer from one or a few parts of the guidebook.

The chapters are written to work as a stand-alone set of principles and practices on a subject.

As we get ready to take a look at the guidebook, I want to share some thoughts about the flow of the book.

This guidebook can be entered into and used in a number of ways:

1. You can read from the beginning to the end, and build your knowledge as you go.
2. You can start with what interests you the most, follow some cross-references, and skip around to read.
3. You can come with a specific problem or general question, then find and read only the content relevant to your situation.

That's most of the ways, I reckon.

The chapters are written to work as a stand-alone set of principles and practices on a subject. For the most part, no chapter is a prerequisite to understanding any other chapter.

We made this stylistic choice as a way to provide the most value to the reader,

without spending six more months polishing the chapters so they moved together perfectly smoothly

Most of the chapters are co-authored by two or more people, with the merge work sometimes done by a third person. While we have a style guide to follow, you can tell there are different writers at work on a chapter.

But overall, the content has been curated and polished to be useful for different types of readers' needs. We'll continue to polish the content in future versions, so the reading experience is smoother over time. That's open source content—release and iterate on improvements next release.

Let's take a look at the book.

{switch to browser}

There's a lot that's different about this book from the 1.0 version—the structure, the reading experience, and so on. But for a sizeable chunk of the core ideas, they are much the same as they were ten years ago. Most open source principles are largely the same or somewhat evolved.

There are two key differences in the content itself from the past:

1. Some topics that only got a few paragraphs mention in the past now have entire chapters. Take Governance as an example. In 2009 it was much less of a popular topic beyond certain groups of people. By 2019 when we started outlining the 2.0 version, it became worthy of a chapter itself, and in fact is the longest chapter in the guidebook.

2. Some topics were not being discussed at all or openly in many community management circles. One example of this is the chapter on community manager self care, which is currently the second longest chapter.

- **Presenting the Open Source Way** the introduction that sets the vision for the guidebook, then explains how that vision is delivered in:
  - **The Shape of Things (I.e., Assumptions We Are Making)**
  - **Structure of This Guide**



- **A Community of Practice Always Rebuilding Itself**
- **Getting Started** this section is for the early work, or even work done before a community is formed, to establish and set the right direction from the start.
  - **Community 101: Understanding, Joining, or Forming a New Community** provides a broad overview of how to analyze your community and others.
  - **New Project Checklist** is a checklist of elements that are typical and essential to starting a new project. It's quite like the one we hand out to Red Hat engineers, and it's good for making sure you've considered all the common stumbling blocks.
  - **Creating an Open Source Product Strategy** is useful for when you want to plan and prepare for creating a new commercial product out of your open source software project.
- **Attracting Users** can't tell you all the ways you can attract your special users to your unique project, but it can tell you how to keep them running away when they arrive.
  - **Communication Norms in Open Source Software Projects** is the heart and soul of how to prevent or deal with difficult and toxic situations, as well as the whole range of communication practices that keep things running smoothly. This ranges from project roadmaps to issue tracker best practices.
  - **To Build Diverse Open Source Communities, Make Them Inclusive First** explains why and how to focus on inclusivity as the first step of increasing the diversity of your community. This chapter offers an overview of efforts by various open source communities to make projects more diverse and inclusive. It also offers initial steps open source community managers and project maintainers can take to begin building communities and projects that benefit from a diverse contributor base.
- **Guiding Participants** helps you understand general participants and enthusiasts.
  - **Why Do People Participate in Open Source Communities?** gets into the kinds of motivations that compell people to contribute to open source.
- **Growing Contributors** is the section with the most complexity, which is similar to the relationship of contributors in a project.

- **From Users to Contributors** helps you create a welcoming environment for newcomers who may or are interested in becoming contributors.
- **What Is a Contribution?** Provides a concise explanation of a contribution, with examples.
- **Essentials of Building a Community** is a concise single chapter covering a range of essential components in building a community.
- **Onboarding** focuses on the basics, practices, and resources for bringing new contributors into the community. It describes the concept of contributor pathways as ways to help newcomers grow to become experienced contributors.
- **Creating a Culture of Mentorship** covers the concepts and practices for creating a peer-to-peer mentorship program. Most successful open source projects have formal or informal mentoring underway all the time, and find it essential to contributor success.
- **Project and Community Governance** is a thorough exploration of governance in open source communities.
- **Community Roles** is a deeper look at the many different types of roles beyond code and other typically named contributions.
- **Community Manager Self-Care** represents a new area of best practices for the 2.0 version, one that I'm not sure many of us would have thought up ten years ago. The idea for this chapter came about during a content design workshop at FOSDEM 2020 in Brussels, and the community manager who suggested the content ended up writing the chapter for the release. One note about this chapter: as with all of our chapters, we had a subject matter expert, or SME, review the content. Where other chapters were in the domain expertise of the other writers, we could often cross-check each other's work. In the case of this chapter, we asked Dr. Karen Hixson to review the chapter with the author.
- **Measuring Success** encompasses the lifecycle of a project and the metrics you use to keep track of health and accomplishment.
  - **Defining Healthy Communities** helps you understand what a healthy open source community looks like, coming from as high-level right down to hands-on inquiries. These inquiries take

- apart and consider many aspects of a project, as viewed from the outside looking in, and from inside of the project looking at itself.
- **Understanding Community Metrics** is a discussion of the many types of topics to measure, with question suggestions for each topic, followed by a few sections on choosing metrics for your project.
- **Announcing Software Releases** is a step-by-step milestone-based process with templates for preparing-for and announcing a software release.
- **Contributors**
  - **Chapters writers**
  - **Project teams**

## A community of practice: OSW



More than a guidebook, a full community.

- Be a community of practice around community management.
- Create and curate content of all types, from members and the wider community, into resources such as guidebooks and courses.
- [WIP] Have regular meetups to hear from special presenters and share practices among community members and the ecosystem at large.

As we come toward the end, let's circle back for a moment to talk about the Open Source Way community itself.

It is a slightly-odd thing to say we have a community of practice around communities of practice, but that in a very real sense is what this is.

We are defining and curating principles, practices, and stories (reasons why) from across many open source projects, which are themselves communities of practice. And doing this as a community of practice ourselves.

But anyway ... our first efforts are around curating (creating, finding and merging in) useful resources for our users, such as guidebooks and courses. We'll continue expanding outward, make updates to 2.x series with a print version, and listen to what the users and contributors want next.

A common tool of communities of practice is a meeting or meetup, to include presentations by members or special guests, with discussions among members about related topics. I'm looking to start that schedule in the a bit later in 2021.

## Questions

Here are some!

1. If attracting users is so important, why are there so few chapters on that subject?
2. Who is in this community of practice, sharing all these opinions?
3. Can I pull just the chapters or material I want into a smaller handbook?
4. What if I disagree with some of your practices and principles?

And now we're coming to the questions and answers period.

In case you don't have any questions, here are some I thought of for you all out of material I didn't have time to include in the main presentation.

1. If attracting users is so important, why don't you have more chapters on that subject?
  - a. For open source software, it turns out that taking care of contributors shares a lot in common across projects, and it's a different set of practices than many people have experienced in non-open source development environments. There is the most common knowledge to teach there. On the other hand, what users need is wide and varied, and the ways to attract them to a specific piece of software is not something that can be generalized. There are plenty of materials out there on building a "product" people want, and those are more potentially more applicable than the
2. Who is in this community of practice, sharing all these opinions?
  - a. We can take a look at the list of a writers and editors.
3. Can I pull just the chapters or material I want into a smaller handbook?

- a. Yes, that's an explicit goal of the project, to work toward making that useful and easy to do.
- 2. What if I disagree with some of your practices and principles?
  - a. We may just have a different approach. But if you'd like to engage in a discussion with the project writers, you can come to our forum or file an issue in our git repository.

## Image attributions



Slide 2: *Sign Self Guided Trail* (CC BY 2.0)

<https://www.flickr.com/photos/shinazy/7304697496/>

Slide 3: *Pyramide Inversée* (CC BY 2.0)

<https://www.flickr.com/photos/carloszgz/28444272150/>

Slide 5: *Maslow's hierarchy of (community) needs* (CC BY 2.0)

<https://www.flickr.com/photos/opensourceway/4371000464/>

Slide 6: *Intersection of core values in open source and domain driven design* (CC BY 2.0)

<https://flickr.com/photos/opensourceway/5445602998/>

Slide 7: *Creative Commons - Attribution* (CC BY 2.0)

<https://flickr.com/photos/mathplourde/4418832286/>

Slide 8: *A GRAND MORNING FOR A WALK* (CC BY 2.0)

<https://flickr.com/photos/summonedbyfells/8674608581/>



## Thanks for the walk!

<https://theopensourceway.org>

<https://github.com/theopensourceway/guidebook>

<https://lists.theopensourceway.org/> (discussion fora)

Karsten Wade, Sr. Community Architect, [@quaid](https://twitter.com/quaid)

[kwade@redhat.com](mailto:kwade@redhat.com)

[iquaid.org](http://iquaid.org)

So thank you for spending your time with me, and for sharing your attention. If you have any questions, feel free to reach out to me. You can find me on various social media as 'quaid', or you can email me at [kwade@redhat.com](mailto:kwade@redhat.com). And I'm now available for your questions.