Managing an open source project

Tracy Teal

2023-08-29

Table of contents

Workshop overview

September 17, 2023

09:00 - 17:00

TBA

Click here to register

Overview

An open source project starts with ideas and code, but it continues with people. We know that most open source projects rely on just one or two people for most of the work. And code is just a small part of these roles, which also include project management, conflict resolution, decision making in uncertain situations, building an inclusive community, and lots and lots of communication. Whether you're just starting a project, interested in getting involved in open source, or already have a community of thousands, there are some tips, tricks and templates that you can use to make maintaining an open source project more manageable.

In this interactive 1-day workshop you'll learn some key practices for effect open source project management. You'll walk away with new approaches for making project decisions, better strategies to manage user and contributor interactions, and better ways to set boundaries. Different projects make different decisions based on their needs and in this workshop you'll see what types of decisions are possible, and how you can make the ones appropriate for you. During the workshop, you'll create useful documents for your repository and begin to outline some processes you can finalize later or with your team.

This workshop is for you if you:

- are involved in maintaining an open source project and struggling to feel like it's sustainable, or are looking for practice or guidance,
- are starting out or interested in being involved in maintaining an open source project, and want to learn how to set up the project for the most effective engagement from contributors and users, or
- are interested in learning more about the people side of open source project maintenance and connecting with other maintainers.

Workshop format

The workshop will include four modules:

- Setting project goals
- Managing expectations
- Decision making
- Surviving and thriving as an open source maintainer

Throughout each module we'll have a general presentation and discussion of the topic, and then hands-on time for you to consider your own project, and to start to create some of the documentation or resources you need.

Prework

No pre-work is required for this workshop, but part of the workshop will be asking you to consider some questions for your own project. So, you should be familiar with at least one open source project, and have reviewed the types of non-code issues or challenges you're seeing with the project currently.

You should plan to bring your laptop to the workshop, so you can review or look up things about your open source project. We will be writing some documentation, and a familiarity with GitHub and markdown syntax is assumed.

We will also be using pen and paper! These will be provided at the workshop, but if you have a favorite brainstorming pen or pencil, please bring it along.

Instructors

Instructor: **Dr. Tracy Teal** (she/her) is the Open Source Program Director at Posit. Previously, she was a co-founder of Data Carpentry and the Executive Director of The Carpentries. She developed open source bioinformatics software as an assistant professor at Michigan State

University and holds a PhD in computation and neural systems from California Institute of Technology. Tracy is involved in the open source software and reproducible research communities, including serving on advisory committees for NumFOCUS, pyOpenSci, the R Consortium and carbonplan, and has been working with open source communities, developing curriculum, and teaching people how to work with data and code as a developer, instructor and project leader throughout her career.

TA: Isabel Zimmerman (she/her) is an Open Source Software Engineer at Posit, where she builds Python-based MLOps frameworks. She recieved her Masters in Computer Science at Florida Polytechnic University where she specialized in data science. Isabel is also part of the editorial team at the open source scientific software community, pyOpenSci. When not thinking about machine learning systems, she enjoys gardening and competing in agility competitions with her dog, Toast.

1 Setting project goals {number-depth:=“ ” 1=“ “}

Lesson objectives

In this lesson we'll consider a few questions that help you think about and articulate the goals for your open source project.

Your project likely started out with some ideas in mind of what you or others were trying to achieve. That might have been to solve a problem you were having, an interesting idea to explore, or a shared challenge people were experiencing. Whatever those reasons were, you might not have been totally aware of them at the time, and those goals may have shifted over time.

The first step to managing an open source project is to visit, or revisit, these ideas around what you think your project is aiming to achieve, the resources you have available to you, and how you and the team you work with work best.

The people who use and contribute to your project all might have different ideas for any of these questions. Hearing from the community is crucial, but ultimately **you get and your team get to decide the goals for your project**. Once you know this, you can more clearly articulate the decisions you make, why and how you make them, expectations for the work, and how to contribute.

So, like with most things, we start first with more questions than answers!

- Slides
- Worksheet

We're going to go through the questions in the 'Project Goals' worksheet and answer them for your project. Note that none of these questions are meant to ask you to create solutions! This is about gathering information about your goals.

1.1 Project Goals questions

1.1.1 The numbers

Let's start with an overview of your project.

- Project name:
- Online resources: github repo, web site, etc
- How long has your project been around?
- About how many people use your project?
- About how many contributors do you have? (however you define contributors)
- About how many core project members do you have? (however you define that)
- What stage would you describe your project? experimental, stable, sunsetting, something else?
- What are you most proud of about this project (ok, this isn't a number)?

> **Exercise**
>
> Answer these questions for your project. Pair up with someone else in the workshop and go talk through your responses with each other.

1.1.2 Start with why.

You started or got involved in this project to solve some problem for yourself or others. Why are you involved in this project, who is that person you're trying to help? It's fine if this person is current, present or future you!

1.1.3 Your envisioned users

Describe one or two types of people you envision your project serving

- What field(s) do they work or are living/working in that are relevant for your project?
- When are they using your project? As a part of their work, their hobby, or managing their lives?
- What tasks are they struggling with that your project addresses?
- What are they trying to achieve in their work/life that's relevant for your project?
- Are there any restrictions on how they can use your project? e.g. control on installation, access to data, etc.
- How much time do they have for this work?
- Is this a core part of what they're trying to do, or something that's necessary, but not where they want to spend their time?

- What's their level of programming/data science experience?
- What other limitations or features are you trying to address for your users with your project?
- What other questions did you think of about your users that you'd like to answer?

> **Exercise**
>
> Answer these questions for your project. Pair up with someone else in the workshop. What's one thing you realized about who you're trying to serve with your project, going through these questions?
> Timing: 20 minutes answering questions, 10 minutes discussing

1.1.4 And onto how many

Now that you've thought about an individual user or two, now let's consider how many people you'd like your project to be able to serve, what community.

Considering your target users from above, how many people are in this category, where are they, and what percent of a field or community do they represent?

Again, this is about your goals, not necessarily where you are now.

- How many people does your target user(s) represent (in orders of magnitude)?
- Are they easily identified and reached?
- What percent of a field do they represent?
- What does success look like for you? What percent of these people using your project is success?

> **Exercise**
>
> Answer these questions for your project. Pair up with someone else in the workshop. What's one thing you realized about the community you're trying to reach in answering these questions?
> Timing: 15 minutes answering questions, 5 minutes discussing

1.1.5 You and your team

OK, so you've thought about the people and problems you want to serve with your project. But it's just you and your team! Maybe your team is just you. So, what's important to you and your team, about the time you have and how you work best.

- How many people would you consider to be core members of this project and who are they? (How you define this is up to you)
- How much time do they have to work on this project, and is a part of work or outside-of-work time, or a mix, where that time would be sustainable? Be realistic!
- Are any of the team members paid to work on this project?
- What parts of the project do you like working on best?
- What parts of the project worry you the most?
- On your team (if you have one), who likes to do what, and/or is good at different areas, on the project? (Not who does what, but who likes and/or is good at the different project aspects)
- What do you see as some gaps, i.e. 'this work isn't anyone on the team's current area of strength'.
- What personally is success for you and your relationship with this project? Complete this sentence. In one year, I will be happy if I am spending X amount of time on this project, am excited about Y and not worried about Z.

> **Exercise**
>
> Answer these questions for your project. Pair up with someone else in the workshop. Share with them your response to the last question (what does success look like for you). Timing: 15 minutes answering questions, 5 minutes discussing

1.1.6 Contributors

Now that you've thought about who you want to serve with your project, and what's important for you and your team, the other piece of the project is contributors. What are your goals here? How do you see that people can best contribute?

- What's an example of a contribution to your project that you really appreciated?
- What is an example of a contribution that was difficult to handle?
- Do you like to work through github, Slack, in person meetings or other channels?

> **Exercise**
>
> Answer these questions for your project. We'll use the responses in future sections.

2 Managing expectations

> Lesson objectives
>
> In this lesson we'll discuss the different ways you can communicate expectations for your project and work with some templates to create that documentation for your project.

Setting up your project is about setting expectations - about what you expect from people and what people can expect from the project.

People have a lot of different ideas of what open source means, if they're new or have been around awhile, what other communities they've worked with. They'll bring all of these things with them to your project. To ensure you and they have the best experinece with your project, it's good to be clear about expecationa, what they can expect from you and you can expect from them. This means you need to write it down and put it somewhere expected and ifnablae. It does not mean everyone is going to read it who comes to your project, but it's something that is there for onbaording and something you can point to in replies. While we're talking about open source, and that means certain things, there are many different modes of 'open contribution'.

Now, when do you need to write these things down? Sooner than you think you do, but not too soon. :) There are some things you need to write down write away, and others that can come later or change over time.

2.1 Project setup

Some standard files you'll have are a license and a Code of Conduct. If you don't have special needs for your project, the MIT License is likely a good choice, and the standard GitHub Code of Conduct. Remember to identify a reporting process for your Code of Conduct. That's a key component, just having the file is not enough.

> Exercise
>
> Check your project for license and Code of Conduct files in your repository. Are they there?

In your Code of Conduct do you have:

- contact information
- information on who receives an email/report
- for the people who receive a report, do they have an idea of how to handle a report?

If after looking through this information, you'd like to learn more about Codes of Conduct, this is a good resource.

While there are people who will mainly use your project, and not engage with the code, there's a spectrum of users to contributors, not clear categories. In open source, most contributors are also users of the software. That's why they want to contribute!

For this module we're going to focus on people who engage with the project through wherever you have the code for the project through issues and PRs.

When people share issues or PRs they usually have some expectation for a response. In the absense of information from the project on what that will be, people have a lot of their own ideas they bring to the interaction. Note, again, even if you write all of this down, people might not see it. But writing it down gives you something to point to, to respond. So, you've articulated your goals for the project and your team. How should people share with you, and what can they expect as a response?

Sometimes this information is in a CONTRIBUTING.md file, but you might have it in a README or somewhere else.

Exercise

Fill out this template for a CONTRIBUTING.md file for your project.
CONTRIBUTING.md template

You may have noticed in the CONTIBUTING file, we said that the guidelines were for people who are new contributors. That's because your project likely has different categories of contributors.

The tidyverse has a pull request guide that outlines three different types of contributors. For people in other modes of collaboration on your project, you should still be clear about the expectations for submission and review, but you likely want to address those differently than for someone who is submitting one or two issues/PRs to your project.

> **Exercise**
>
> Look at the .. section in the tidyverse PR guide. What types of collaboration do you have currently in your project? How might you adjust expectations for your project in the '...' mode, versus how the tidyverse guide does.

2.2 Language

You've done a great job figuring out how you'd best like contributions, and your goals and boundaries for the project. Still, people won't necessarily see this information when they come to your project, so it will be a process of consistently onboarding new people and replying to issues and PRs. Ideally you want to be able to say no when needed, and still keep people engaged and feel included.

These are some ideas and templates of language you can use in different situations. Note though, that everyone has their different style that's true to them, so adapt as needed to match who you and your project are.

2.2.1 Examples

These are some examples of response styles and approaches where you can see the style matches the maintainer, and also makes people feel positively about the project, even if the project isn't looking for a lot of active contributions.

- bear project
- tidyverse

What examples do you have that you like?

2.2.2 Sample language

These are some examples for different situations. Now, ChatGPT is likely to also give you some helpful starting places!

> **Exercise**
>
> Try adapting one of these language templates for one of your open issues or PRs.

3 Decision making

> Lesson objectives
>
> In this lesson we'll discuss decision making for your project. How can you determine how you make decisions? Resolve conflicts? How do you communicate with others how you make decisions?

Like any project, for your open source project, you need to make decisions about what to do, what to implement, what not, how to go about it, how to document something, or what language to use in an error message. Unlike some other projects you work on though, all the decisions you make are made in public. Therefore it's useful to have a framework for how decisions are made, so people don't just see a decision, but can see some of the process that you used to get there.

Decision making has a 'who' and a 'how' component.

Let's start with 'who'.

3.1 The 'who' of decision making: governance

Governance is decision making. You set up governance structures, so you can make decisions for the project. Here again you need to think about project goals.

I'll start by being a little controversial here. You don't always need a governance structure. If it's just you working on the project and you're making all the decisions, and it's at an early lifecycle stage, that's probably fine. That's maybe less fine as the project beomces more central or other factors though. And here's where I back off from that initial controversial claim. We can rephrase it as a single person making decisions is its own form of governance. We just don't often think about it that way.

So, let's talk about a few differnt models.

3.1.1 Single maintainer

So, it's just you. You're the primary contributor and docs writer and all the things. So, clearly you make all the decisions and everyone knows that, and it's fine, right. Alas, no, 1) some people might not know that and 2) you still need to decide and communicate how you make decisions. Think of future you here, will you understand how past you made decisions.

There's not necessarily an obvious file or section to write about a governance model. However, you won't have much to write here, so you could include some text in your CONTRIBUTING.md, even though someone could figure it out from the github information, like

> PERSON is the primary maintainer for this project. Decisions for the project will be made with (list or link to project goals) in mind and the need for continuing maintenance on the project. The project also needs to stay compatible with X, Y and Z.
>
> If you'd like to propose a new idea for the project, please share it an issue for discussion.

Again, this will be something you can point back to if, e.g. someone puts in a PR that's out of scope or will be hard to maintain.

3.1.2 Multi-person project

You're growing or started this project as a collaboration. The main thing you need to do is make some decisions about how you'll work together and make decisions. There's a lot of models for this.

Red Hat has a nice overview of six different models.

Understanding open source governance models

3.1.3 Multi-stakeholder project

How is a multi-stakeholder project different than a multi-person project? It's different because you are at the stage where you have multiple types of stakeholders - users and contributors who may have competing needs or interests, and resources to put behind these different interests. This is where governance really shines. You need to be clear about how decisions for the project are made that serve the community as a whole. Otherwise you can end up in situations where work gets done just where people can put effort, and that might not be the top set of priorities for the project as a whole.

> **Exercise**
>
> In your project, who makes the decisions? Draft a couple of sentences about who makes the decisions for your project. This isn't something you'd necessarily share yet, it's a first draft for your own clarity. Pair up and talk about this with a partner.

3.2 The 'how' of decision making

Once you know who is making decisions, you can think about how you make them. What input do you include, what rubric do you use for making decisions (even if informal)? Having project goals and knowing roles within the team can already have put you on a path to make this process easier.

> **Exercise**
>
> How would you say decisions are made on your project now? Write a couple of sentences that describe how decisions are made.

3.2.1 Resolving conflict

You have some clarity on who makes decisions and how, and that's a big help! There will always though be disagreement, either within the group of decision makers, or with community members. So how do you resolve conflict? This is one of the top thing we're usually uncomfortable with. Conflict isn't always bad! It can help us get to better ideas, but it isn't usually something we look forward to encountering.

A lot of the general writing on conflict resolution applies here.

4 Surviving and thriving as an open source maintainer

> Lesson objectives
>
> In this lesson we'll discuss how to not just survive as an open source maintainer, but to thrive! We'll explore what's most important to you to make this possible, and the systems, structures and communities to connect with to help make this possible.

Hopefully some of the things we've talked about already, will help you have some clarity on your goals for your project and yourself, and therefore let you have a decision making process for where and how you spend your time. That's a big part of surviving and thriving!

It's one thing to know these things in theory though, and another to put them into practice, and most importantly make these new patterns for you or your project.

We'll discuss some approaches you can try, have a discussion around where you've had challenges, and find shared ways to support each other. (Including if content like this is valuable)

4.1 Approaches to try

There are a million and one books on 'habits', 'boundaries', 'self-compassion' and other related topics. There's not a new concept I can add here, and I'll share that I'm still on a journey to figuring it out myself.

So, what I'll do is try to give you a framework for you to find your own strategies that work for you, and the resources that have resonated the most for me. My best advice is to be cautious of advice. :) People are different from each other, have intersectional identities, and different backgrounds and experiences that led them to where they are, so something that worked for someone else is not necessarily something that will be right to you. So, maybe my advice is to listen to other people's stories and what they share, and take from it what makes sense for you. I have yet to read a book or hear about someone's journey where I agreed with everything, but I almost always take away at least one thing (even if it's a 'not to do' thing).

One thing to mention though is that **you are a leader**. If you are taking this course, it means you're working on a project and learning more about supporting the project and people you

work with. Often the transition is subtle, so you don't **feel** like a leader yet. But leadership isn't through title or position, it's about wanting to serve. (ok, everyone might not agree with this, but it's my definition of leadership).

A quote I like from one of my favorite show Numb3rs is "a teacher is a leader at the front of the room". Open source maintainers are leaders in the front of the code base. Thank you for what you do.

> **Exercise**
>
> Where you have had challenges in a misalignment with how you decided you were going to work, with how you did?
> What are some approaches that have worked for you in aligning your goals with your actions?

> **Exercise**
>
> What things would be useful after this workshop?

4.1.1 Some resources

- Good to Great
- John Wooden on Leadership
- Leadership is an Art
- How to be a great boss
- 5 Principles of concious leadership
- Things from Lara Hogan

References