

Managing an open source project

Tracy Teal

2023-08-29

Table of contents

Workshop overview	3
Overview	4
Workshop format	5
Pework	6
Instructors	6
Setting project goals	8
Project Goals Worksheet questions	8
The numbers	8
Start with why.	9
Your envisioned users	9
And onto how many	9
You and your team	10
Contributors	10
1 Managing expectations	11
1.1 Project setup	11
1.2 For people contributing to the project	12
1.3 For people using the project	12
2 Decision making	13
3 Surviving and thriving as an open source maintainer	14
References	15

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 effective 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

Setting project goals

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.

Project Goals Worksheet questions

The numbers

Let's start with a little 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, sunseting, something else?

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!

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?

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.

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?

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.

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?

1 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 experience with your project, it's good to be clear about expectations, 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 onboarding 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.

1.1 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.

There are two main categories of people who will engage with your project - contributors (people who are contributing code, documentation or issues to the project) and users (people who primarily are using the code for their own projects). We'll consider both categories and how to set and manage expectations for each group.

1.2 For people contributing to the project

1.3 For people using the project

2 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?

3 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.

References