

SECTION THREE

SUSTAIN



"If you fail to plan, you are planning to fail!"

- Benjamin Franklin,
Inventor, eventual abolitionist

HOW TO OPEN SOURCE

Except where otherwise noted, this book is licensed under Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) by Richard Schneeman.

Illustrations Copyright © 2022 Travis Stewart

Publisher	Richard Schneeman
Illustrator	Travis Stewart
Cover Designer	Travis Stewart
Editor	Ruby Ku

Building a contribution practice

If you've found yourself thinking, "I would love to contribute, but I just don't have the time," you're not alone. Time is constantly in short supply. The day always seems to end before you can write that issue response or start researching that feature idea. Unfortunately, I don't have a way to give you more hours in the day. Instead, we'll look at how successful contributors find the time and how you can do it too.

Successful contributors don't rely on willpower to find time. Instead, they work to craft systems that support their contributions and lifestyle. If they find themselves overwhelmed, they don't push through. They find ways to cut scope and floss one tooth. They don't wait to work on open source "when they feel like it". They build habits, block-out time, and rely on tools to keep them on track. Building a successful contribution practice won't give you more time to work, but it might feel that way.

In this section, we will look at how four personas contribute to open source. Pick the one you resonate with the most and learn how to adapt their practices to your life. Keep your SMART contribution goals in mind while you're working on your plan.

There is no "one right way" of contributing to open source. Committing to a plan that fits your lifestyle sets you up for success.

Once-a-week dabbler

Meet Kayla (she/her). Kayla likes React.js, pamplemousse flavored sparkling water and has carved out an hour to contribute to open source every week.

She set a recurring calendar entry for herself on her corporate account so no one else would schedule time over this block. When she first started, Kayla would choose to take the hour off over her

lunchtime, but she gradually found that morning work and unexpected incidents would bleed into this time. She would forget about her open source work until it was too late.

Kayla's calendar notification now fires first thing in the morning every Thursday. Some days, she's excited to see what new issues she will encounter. Some days, she is terrified she won't know what to do. But after a few minutes of sipping sparkly water and reading issues, she always finds something interesting.

Kayla has been working on open source contribution for a while. She took some vacation time to go backpacking, but she's about to hit a forty-week streak. That means in the last year; she's dedicated almost a whole work week to practicing open source contribution. Way to go, Kayla!

Her time is starting to pay off, too. The work is beginning to feel natural, and all those communication and code skills are showing up in the workplace. She heard a rumor that one of the projects she contributes to might give her commit access pretty soon!

Friday Funday (Or some other day)

Lemar (he/him) loves coffee and working with Python. He started by carving out an hour a week but found that the closer he looked at the repos, the more contribution ideas he got. He expanded from once a week to once a day. This pattern worked for a while, but he found the end of the hour would come too soon, and he had to spend too much time getting his head back in the contribution space each time.

Initially, Lemar tried to schedule time on Monday. Unfortunately, he found that too many little fires around the office demanded his attention. They would keep him from getting uninterrupted open source time.

Now, he prefers to sit down and be focused for larger chunks of time. He noticed that co-workers were likelier to head out of the office early on Friday or take the whole day off as part of a three-day weekend. He decided to devote a half-day to open source every Friday.

Like Kayla, he found that he needed to start early, so he worked from morning until lunch. He has a growing backlog of feature contribution ideas that he gets from helping to maintain documentation and triage issues.

He's had so much success with big time chunks that he started subdividing them into specific contribution goals. For example, he always reads and responds to at least one issue and pulls a task from his backlog. Later he explores new contribution ideas.

At four hours a week, he's been able to make a huge impact. Lemar works with his manager to track contributions and balance his team's needs. He already has access to a significant project and is considering taking on maintenance of a small library.

Lemar's co-workers comment on his quality of work and how they wish more people would write such detailed commit messages. Great job, Lemar!

Weekend/weeknight Warrior

Alena (they/them) and Rami (they/him) both program in Java. Alena listens to techno music while they program, while Rami prefers classical. Alena goes to a major university, while Rami is in the final weeks of a hacker bootcamp. They're both comfortable with their programming language. With a few years of coding experience, they have been looking for ways to get more time on practical projects that they can show off to employers.

Because they've got school during the day, they can't commit to working on open source from nine to five.

Alena prefers to work evenings. They've got a Tuesday-Thursday cadence. After they return from classes, they eat a snack, unwind with some cat videos, and then have a thirty-minute session blocked out for open source. They like to set a visual timer to keep them on track. Alena used to get distracted by social sites and the news. After using

a focus app to block those pesky diversions, she now finds she has built more self-control. The work is engaging enough to stay focused without any apps.

Rami is too tired to do extra work in the evenings and has kids to put down for bed. He found that a one-hour session while the kids are napping on the weekend, works well. That practice helped level up their debugging skills. This strategy leaves room for them to have some downtime to pick up the house and watch nature documentaries.

Alena and Rami, you are both putting in the work. Keep it up!

Your turn: pick a plan

These are some made-up scenarios to dramatize some real-world strategies that I've seen. Try to visualize what contributing looks like for you, and then make a specific time commitment on a specific day and put it on the calendar.

To get you started, here are a few questions:

- **What days work best?** Weekdays, weeknights, or weekends?
- **What time of the day do you prefer?** Mornings, afternoons, or evenings?
- **Can you block out a large chunk of time or only tiny chunks?** Big blocks mean you'll have more time to build context, but they are harder to schedule.
- **Can you work toward your goals one day a week or several?** More time means faster growth, but don't burn yourself out.
- **What reminder systems work best to keep you on track?** Calendars, to-do apps, or something else?
- **Where is your optimal contribution environment?** Will you prefer to be at your computer desk or your favorite chair?
- **Are there any foreseeable conflicts that might clash with your plan?** Above we saw incidents, bedtime routines, and workplace norms get in the way.
- **Can you contribute during your workday?** If you're not able to contribute during your workday see [Where contributions meet career](#) for advice.

Take this moment to answer these questions. Don't worry if you don't have all the answers. Your life will change and we will talk about how you can adapt your plan. Once you've answered the questions, put your plan on your schedule. Do it today. Do it right now.

You might think, "but I'm not even done with the book yet." You don't even have to wait to finish to build a schedule. When your contribution time comes up, you can spend it reading about how to contribute. Once you're done with the book, you'll already be in the habit of working toward your open source goals.

You need time and repetition to build good habits. Carving out time for your contribution habit will set you up for success.

Adapt your plan

Almost every one of the examples I gave started with one plan and moved to another. It's okay if you find out your system isn't working for you or that you need to make changes. You don't have to stick to a rigid schedule that you don't like forever.

That said, don't just delete your calendar entry and wing it. That's the same as canceling a personal trainer and saying, "I'll run when I feel like it." If you cancel one set of plans, replace them with another.

Here are some tips that can help you adapt:

- If you have to miss a contribution block, move it to another time instead of deleting or skipping it.
- If the same scenario makes you miss a block several times, consider if it's a temporary or systemic failure. What can you adjust to make sticking to your plan easier?
- Revisit the "pick a plan" questions any time to re-evaluate. Over time your preferences and schedules will change. Forcing yourself to stick with a plan that doesn't work for you can't work in the long run.
- Set aside 5-10 minutes to reflect on the reality of your plan and how well it's meeting your needs. Did you hit your goals or fall short? Why? Adjust your plan based on this feedback. When you're getting started, revisit after your first week. Then after your first month, and regularly every three months after that. Set up a reminder for these contribution plan check-ins, so you don't miss out.
- Consider switching up your schedule just to see what happens. Plan a week where you try something new. Maybe you'll like it. Maybe you won't. Either way, you'll gain valuable feedback.

When I'm frustrated that my plans aren't working, I think of this Bruce Lee quote: "Notice that the stiffest tree is most easily cracked, while the bamboo or willow survives by bending with the wind".

Instead of trying to be perfect, try to be flexible. As you grow, adapt your contribution plan to grow with you.

Recap

1. There is no “one flavor” of open source contributor. Some developers work nights and weekends, while others work during the workday. Pick a schedule and craft a practice that works best for you.
2. If your situation changes or things aren’t working, revisit your goals and contribution practice. Open source contribution is about growth and learning. Find a practice that supports you in your journey.
3. When making plans, rely on systems, not willpower. Use repetition and tools like recurring calendar invites to build sustainable habits.