

# Contributing to Open Source

# **Part 1: Your Expectations, Project Selection, and Protocol**

# OSS Project Expectations

You've published a lot of code but it's time to become real contributors. You will work solo to contribute to a single public open-source project by doing **all** of the following:

**1. Reproduce and confirm or contradict one  
bug report**

**2. Submit one sizable/difficult or two smaller/moderate code-centric pull requests that are worth merging**

**3. Submit one documentation patch  
(README, CONTRIBUTING, code  
documentation, etc)**

**4. Write a single blog post that explains the whole process**

**5. Have the blog post in your M4 portfolio  
(Individual Work)**

**How do you find a good  
project to work on?**

**Small (fewer than 50 contributors)**

**Active Development (last merge to master  
within 60 days)**

**Responsive/Maintained (issues/PRs aren't piling up "stale", show conversation)**

**Approachable (small libraries are better than big frameworks)**

# **A General Protocol**

**Once you've selected a project:**

→ Read the README.md and the CONTRIBUTING.md if there is one

→ Read at least five issue threads

→ Keep track of your steps (anything need updating in the README?) while you get the project running in dev

→ *Start by trying to reproduce a reported bug*

→ Join the conversation first -- you don't want your first post to be "here's a pile of code!"

→ Avoid introducing new dependencies to a project  
(i.e. gems, node modules, etc)

→ Work on a branch that is well named

→ Be conscious and clean with your git commits,  
consider squashing them down into one

→ Look for tools they use (CI, code style review with Code Climate, etc) and use them yourself to highlight any issues

# Submitting A PR

- Once work is done, make sure to get the latest master from the repo and merge it into your branch and rerun tests and manual QA
- Ask a peer or mentor to review your work

## **Submit a pull request with your changes.**

- Reference the issue if there is one related.
- Make sure your pull request comment includes what the change is, why you made it. Give the repo owner any information they need to merge the pull request up front.
- Consider @-tagging the mentor you had review your work in the PR so they can leave feedback

A woman with long, wavy brown hair is shown from the chest up. She is wearing a silver choker necklace and a dark jacket. She is looking off to her right with a neutral expression. The background is a dimly lit room with a lamp and some furniture.

→ Take PR comments and suggestions with grace...

**BRACE YOURSELF**

## **Make changes as they are requested**

- make sure newest master is merged into your local branch, rerun tests, manual QA
- push the changes to your remote branch
- tag the person who requested the changes to review them.

## **Part 2: Research & Pitches**

**It's time to find a project and write up three pitches.**

# Finding a Project

Need some ideas? Generate a shuffled list of the gems you have installed on your system by running this code in IRB:

```
> puts `gem list`.lines.reject{|x| x.include?('rails')}.shuffle
```

# Here are other places to find ideas:

→ CodeTriage

→ Issue Hub

## And here are a few concrete ideas:

→ Faker Gem

→ VCR Gem

→ Figaro

→ Slack-Ruby-Client

## And here are a few concrete ideas:

→ Faraday

→ Launchy

→ MrSpec

→ Adding to / Improving Ruby's Documentation  
(harder than you think)

# **Finding an Issue/Idea**

→ You're *not* expected to be dreaming up new features

→ Fixing and refactoring are generally better than adding new functionality when you first start contributing to a repo

→ Read through the posted Issues and you'll likely find feature requests/ideas. Take note of any tags used by the repo maintainers

→ Non-Code contributions are always appreciated - documentation is hard. Update the README with any set up instructions that you went through that weren't in it already, make spelling and grammar fixes.

# **Project Pitches**

Put together three pitches and we'll help you pick the best one.

Each pitch should contain the following information:

## ## Contributing to VCR

- \* Why: I use VCR regularly in projects, am interested in HTTP details, and would like to dig in deeper.

- \* What: I found a feature request at <http://github.com/vcr/vcr/issues/1> that sounds approachable. It asks for VCR requests to add support for the X-AUTH header.

- \* I Need: to dig into VCR and get the dev setup running locally. I need to read about X-AUTH to understand more about how it's used and the format.

- \* When I'm Done: In the original GitHub issue it sounds like the developers are in favor of the feature, so I think I can get this accepted for the next version of the library.

**You're expected to create a PR (linked in lesson plan) updating your individual markdown file to submit your proposals by 9 AM day one of week two.**

You're encouraged to talk with each other for ideas, inspiration, and collaboration opportunities.

## **Additional Resources**

- Reading: The origins of "OSS" terminology and culture in 1998
- Book: Open Sources: Voices from the Revolution  
(1999)