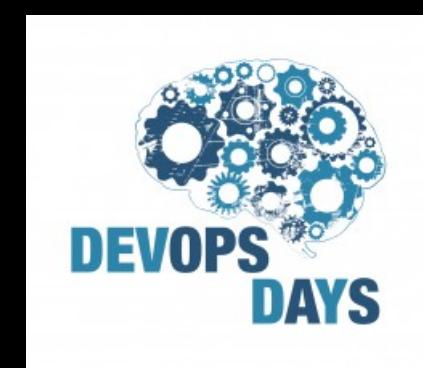


# **THE HANDWAVER'S GUIDE TO CONTRIBUTING TO OPEN SOURCE**

**@MATTSTRATTON - CHEFCONF 2017**

# WHOAMI



@MATTSTRATTON - CHEFCONF 2017

# WHO IS THIS TALK FOR?

- » Enterprise Architects
- » Solution Architects
- » Customer Architects
- » are you seeing a pattern?

@MATTSTRATTON - CHEFCONF 2017

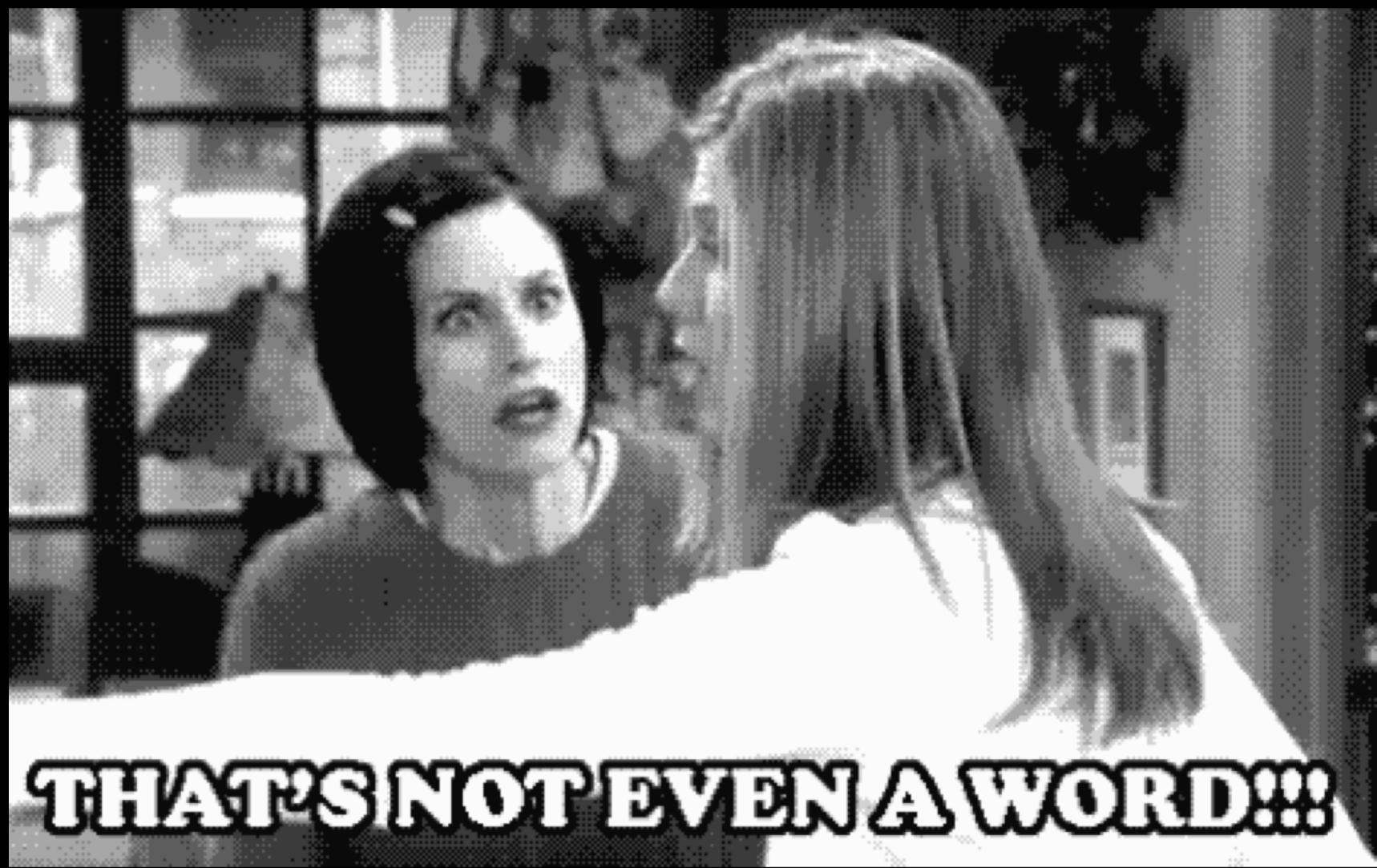


Sup with the wack PlayStation sup

We can definitely feel fraudulent with some technologies that we are trying to understand and design around.

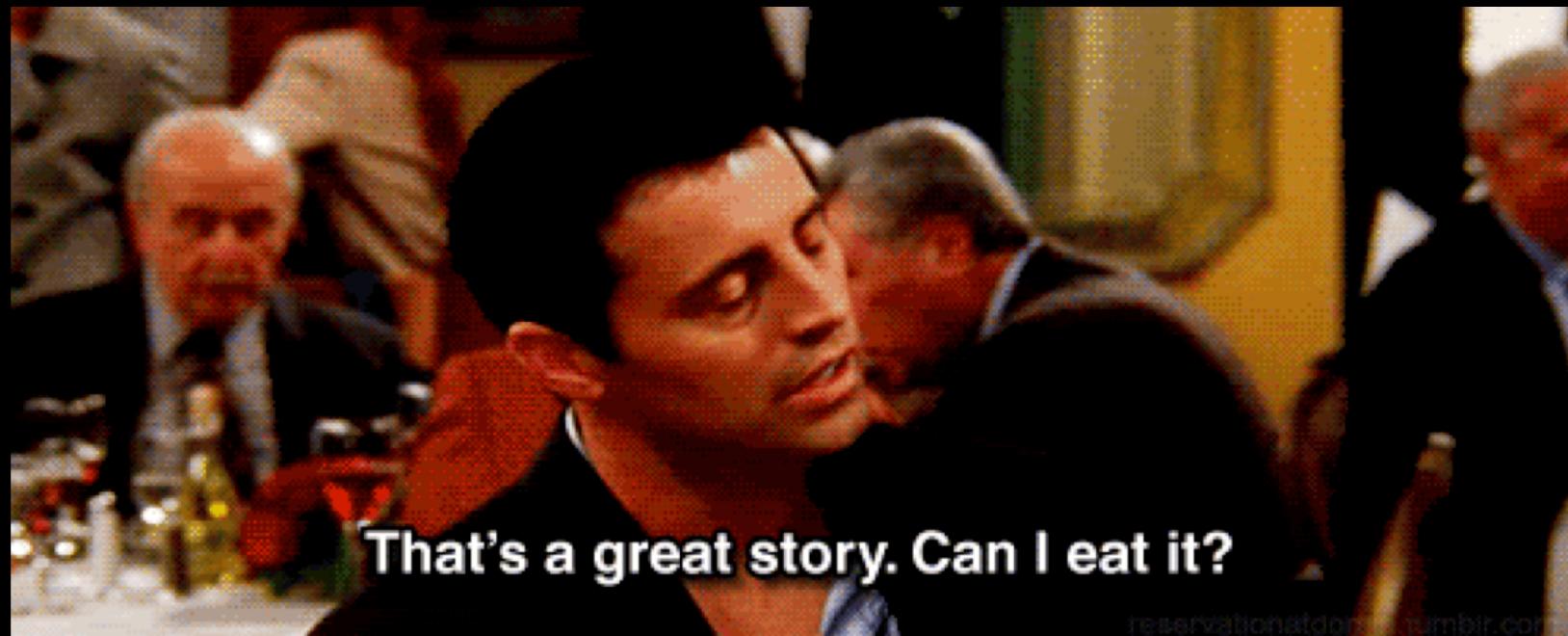
# WHY CONTRIBUTE RATHER THAN START FRESH?

- » You don't really know what you're doing
- » You're working totally alone
- » Chance are, you'll follow some major anti-patterns
- » Heck, you might invent some of your own!



@MATTSTRATTON - CHEFCONF 2017

# A SMALL EXAMPLE FROM MY OWN HISTORY



mattstratton/devopsdays  
vs  
chef-cookbooks/delivery-sugar

@MATTSTRATTON - CHEFCONF 2017

When I decided I wanted to learn Go, it was because I had a command-line application I had been writing in Ruby, and decided "hey, let me try to completely change languages mid-project" (nobody does that, right?). Of course, my use case was an edge case for almost all the Go books I was reading (which were about writing services, etc) and I spent hours and hours every night crawling the googles to find anything that looked right. And I wrote some truly terrible code. It's still not done. Contrariwise, I had an itch. I wanted the delivery-sugar cookbook to support test kitchen with an Azure driver. There already was a test-kitchen resource, and all I wanted to do was extend it. Yes, I know Chef, so this wasn't about me learning "Chef", per se, but it was my first dive into a custom resource in a build cookbook. And because of taking it on that way, I had a pattern to follow, and my fix was released quickly. And then, when I wanted to start working on a brand-new resource for use in a build cookbook, I was leveled up to the point that I could do that reasonably.

# DON'T CALL IT AN AGENDA

- » Determining when to contribute
- » Don't go it alone
- » What not to contribute
- » How to work with ~~containers~~ maintainers

@MATTSTRATTON - CHEFCONF 2017

Think of this as our roadmap through this talk. We are architects. We like roadmaps. See also my clever devops = containers joke.

# WHERE DO I START?



@MATTSTRATTON - CHEFCONF 2017

# SOLVE SOMEONE ELSE'S PROBLEM



@MATTSTRATTON - CHEFCONF 2017

Look through the issues on the repo. See if there are any that seem a) interesting, and b) reasonable for you. Don't take on something huge for your first attempt. Find something small, but it should resonate with you. The reason is, you're going to have to deal with this patch for a while. And if you don't really care about the issue, you'll abandon it.

# SCRATCH YOUR OWN ITCH



The fridge broke,  
so I had to eat everything.

<https://www.torontodrake.tumblr.com>

@MATTSTRATTON - CHEFCONF 2017

Find something that bugs YOU about the project. The best thing ever, as a maintainer, is to see an issue get opened by someone, and then see them close their own PR. Depending upon the way the project works, they may want issues open first. There are no hard and fast rules across the Open Source world. But I can assure you, trying to fix a limitation you see in a project, rather than just dashing off an issue, will see faster traction on making that itch of yours go away.

# DON'T JUST MAKE YOUR OWN THING

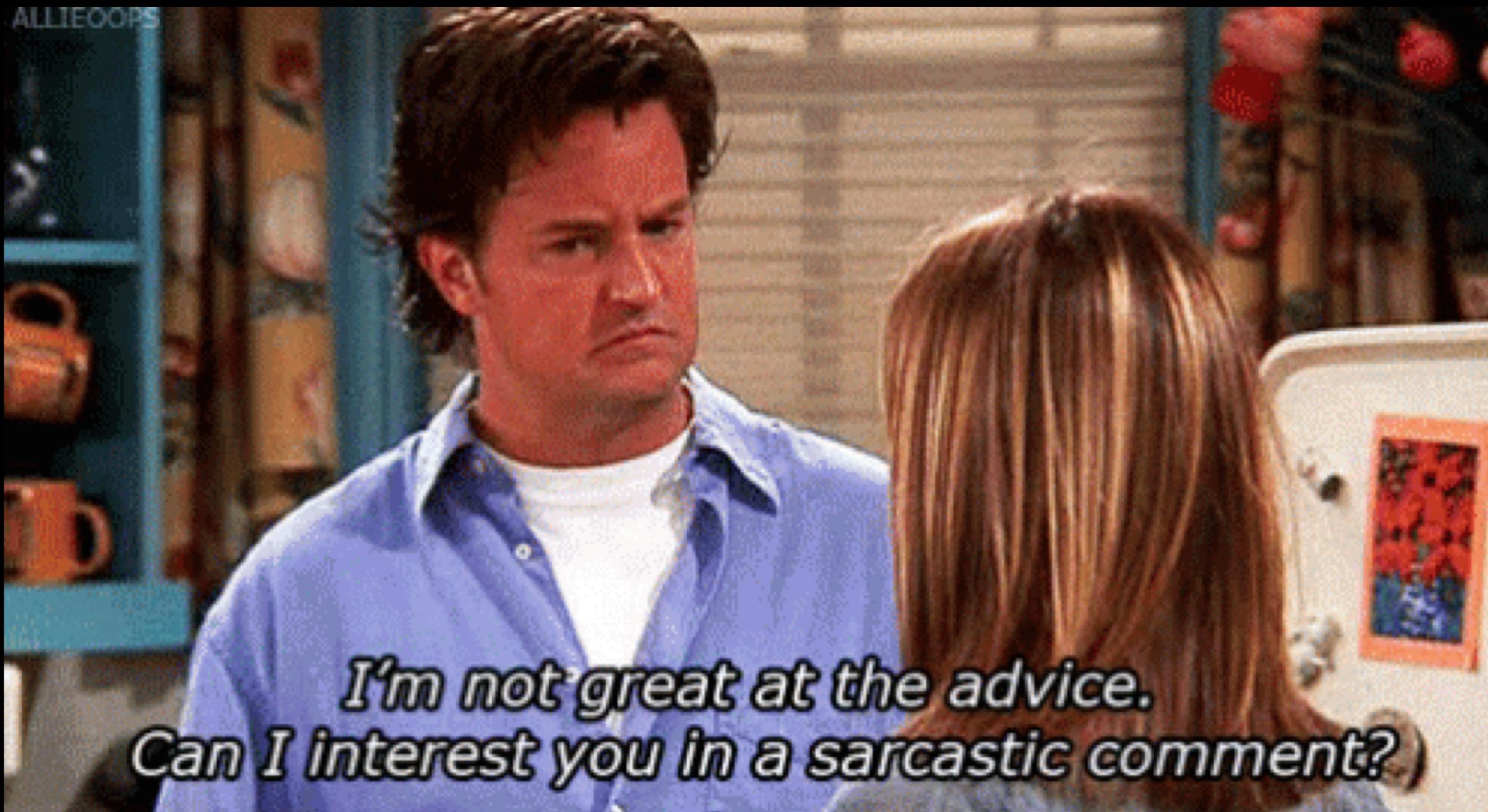


Being alone, sucks!

@MATTSTRATTON - CHEFCONF 2017

Your first project shouldn't be in isolation. As discussed before, you're just out on a raft, floating by yourself, and nobody is there to help you. Take something out there and make it BETTER. I promise you will learn 100x more that way. Or at least learn better things.

# WHAT NOT TO CONTRIBUTE



@MATTSTRATTON - CHEFCONF 2017

A lot of folks will say "A good way to get started is to contribute docs!" That's totally true. And you can do that. I won't tell you not too! But it's really not what we are here for today. Updating an FAQ isn't going to help you understand the limitations (or possibilities) of a technology you are thinking big thoughts about.

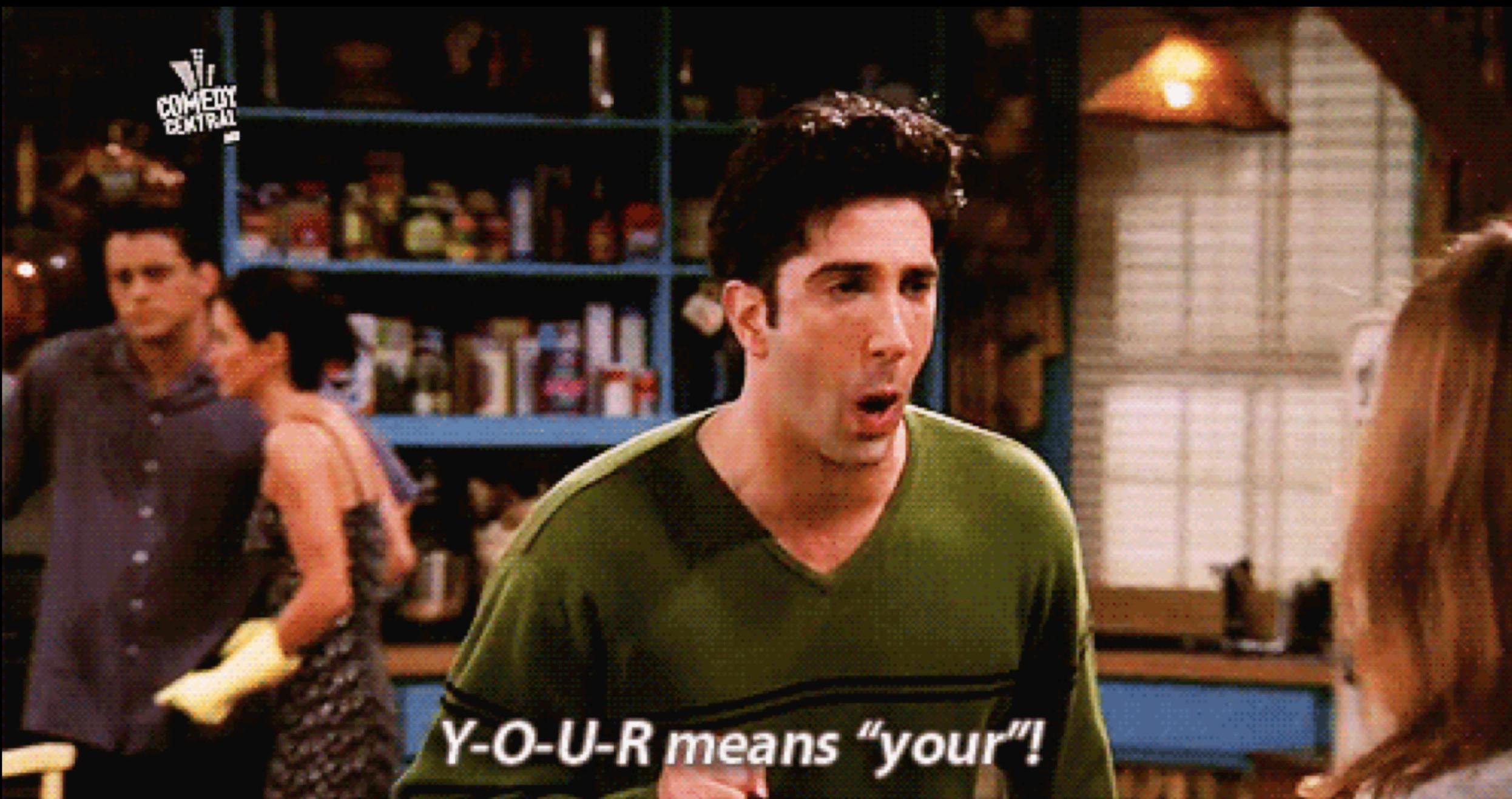
# WORKING WITH MAINTAINERS

- » Follow the contributing guidelines
- » Make your changes small and understandable
- » Test your stuff
- » Don't take it personally



@MATTSTRATTON - CHEFCONF 2017

# FOLLOW CONTRIBUTING GUIDELINES



@MATTSTRATTON - CHEFCONF 2017

# CONTRIBUTING.MD

## From spf13/hugo

### Code Contribution Guidelines

To make the contribution process as seamless as possible, we ask for the following:

- Go ahead and fork the project and make your changes. We encourage pull requests to allow for review and discussion of code changes.
- When you're ready to create a pull request, be sure to:
  - Sign the CLA.
  - Have test cases for the new code. If you have questions about how to do this, please ask in your pull request.
  - Run go fmt.
  - Add documentation if you are adding new features or changing functionality. The docs site lives in /docs.
  - Squash your commits into a single commit. git rebase -i. It's okay to force update your pull request with git push -f.
  - Ensure that make check succeeds. Travis CI (Linux and macOS) and AppVeyor (Windows) will fail the build if make check fails.
  - Follow the Git Commit Message Guidelines below.

@MATTSTRATTON - CHEFCONF 2017

# MAKE YOUR CHANGES SMALL AND UNDERSTANDABLE



@MATTSTRATTON - CHEFCONF 2017

Don't make too many changes at once. What's strange is that in our "real world" jobs, we totally understand this, and would rage at people we work with who try to rewrite the entire application in one release. But we are the same people who will submit a PR that contains 15 different features.

Break that work down into small chunks. And most importantly, make them UNDERSTANDABLE. As Jez Humble has said, version control is just a communication tool. I can't tell you how many PR's I get that have no body, just title. TELL ME WHY YOU ARE DOING THIS THING. Because I can guarantee you, it won't get merged until I do. And you're saving me having to ask you if you tell me up front. Why is this a good change? Who cares about it?

# A BAD CHANGE:

PR#341 - Fixes some stuff

# A GOOD PR:

PR#341 - Enable the use of TOML in addition to YAML  
in configuration files

# SOMETIMES BEING PEDANTIC ISN'T A BAD THING

There's a reason we have conventions in commit messages. Good commit messages follow these guidelines[^1]

- » Initial caps
- » Short introductory summary
- » Describes the change in an imperative manner ("Move class foo out of main" vs "Moved class foo

# TEST YOUR STUFF



@MATTSTRATTON - CHEFCONF 2017

Yes, I know that it's not fun. And sometimes it feels like it's more work than necessary. But make sure you perform the testing guidelines as laid out in the project. And you know what? Maybe there aren't any. That doesn't mean you get a free pass. YOU SHALL ALWAYS TEST.

# SHARE THE RESULTS OF YOUR TESTING



**JOEY DOESN'T SHARE FOOD!!!**

@MATTSTRATTON - CHEFCONF 2017

This may be spelled out in the CONTRIBUTING.md (travis tests, etc), but even if the project doesn't have explicit checks built into the pull request system, add the results of your tests into the PR.

# DON'T TAKE IT PERSONALLY



[reservationsatdorsia.tumblr.com](http://reservationsatdorsia.tumblr.com)

@MATTSTRATTON - CHEFCONF 2017

“One of the hardest parts of being a maintainer is saying “No” to patches you don’t want.

- Jessie Frazelle<sup>2</sup>

 Solomon Hykes  
@solomonstre

Rule #1 of open-source: no is temporary, yes is forever.  
3:38 PM - 30 Mar 2016

425 498

 Arnaud Porterie  
@icecrime

Follow

One of the numerous examples of information asymmetry in open source: contributors put effort in a pet PR, but maintainers manage cattle. 

10:34 AM - 20 May 2016

27 33

<sup>2</sup> <https://blog.jessfraz.com/post/the-art-of-closing/>

@MATTSTRATTON - CHEFCONF 2017

This is the hardest one. You've poured your heart and soul, and spent hours trying to get this PR ready to merge. And then the big bad maintainer says "WONTFIX" and closes it, unmerged.

Despair overtakes you. Or maybe rage. You probably go out onto twitter and start talking smack. The reality is, the maintainer is a human, and when we don't treat git, etc, as a communication tool, we stop communicating as humans. It's totally okay to ask WHY the PR wasn't merged. A good mainainer will actually tell you why. What you can't do (well shouldn't do) is throw a fit. We're all adults here. The maintainers of the project are thinking larger than your one change, and the capability you are adding might conflict with some larger plans. This should make sense to you, ARCHITECTS!!!!

# ADVICE FROM THE FIELD

 **Matt Ray**  
@mattray

Following

Replies to @mattstratton

be descriptive with examples of what you're fixing and be polite. Ask for guidance if there are any questions.

LIKES  
**2**

3:58 PM - 19 May 2017

 **Annie Hedgpeth**  
@anniehedgie

Following

Replies to @mattstratton

Volume! The more you do, the less nervous you become.

LIKES  
**2**

3:58 PM - 19 May 2017

 **Chief Chaos Monkey**  
@DrNemski

[Follow](#)

Replies to @mattstratton

Read the contributions doco

4:14 PM - 19 May 2017

Reply  Retweet  Like  Message 

 **Andrew Suderman**  
@SudermanJr

Following

Replies to @mattstratton

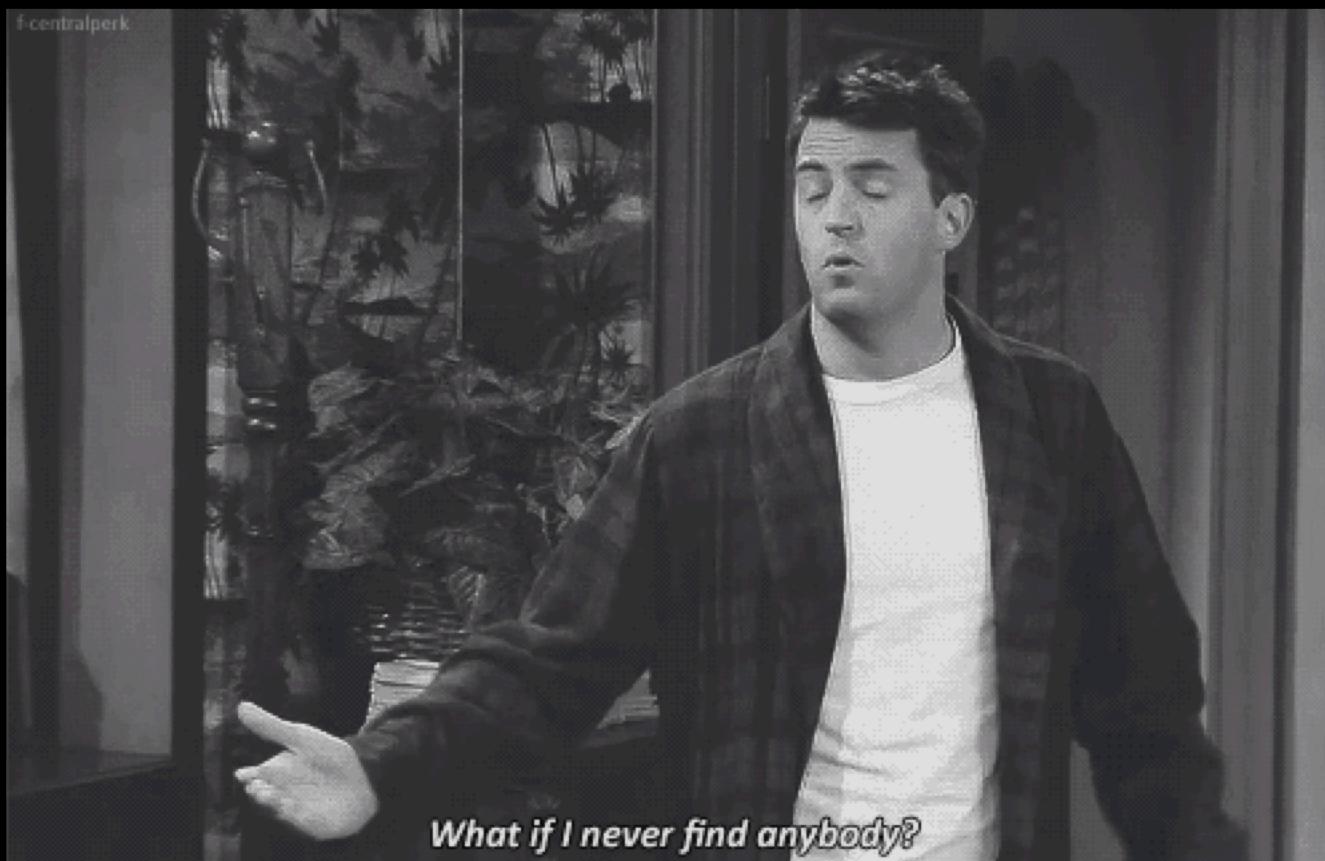
Ask as many as people as possible to find as many problems as they can. That's how we get better.

9:09 PM - 19 May 2017

Reply  Retweet  Like  Message 

@MATTSTRATTON - CHEFCONF 2017

# HERE ARE SOME FRIENDS

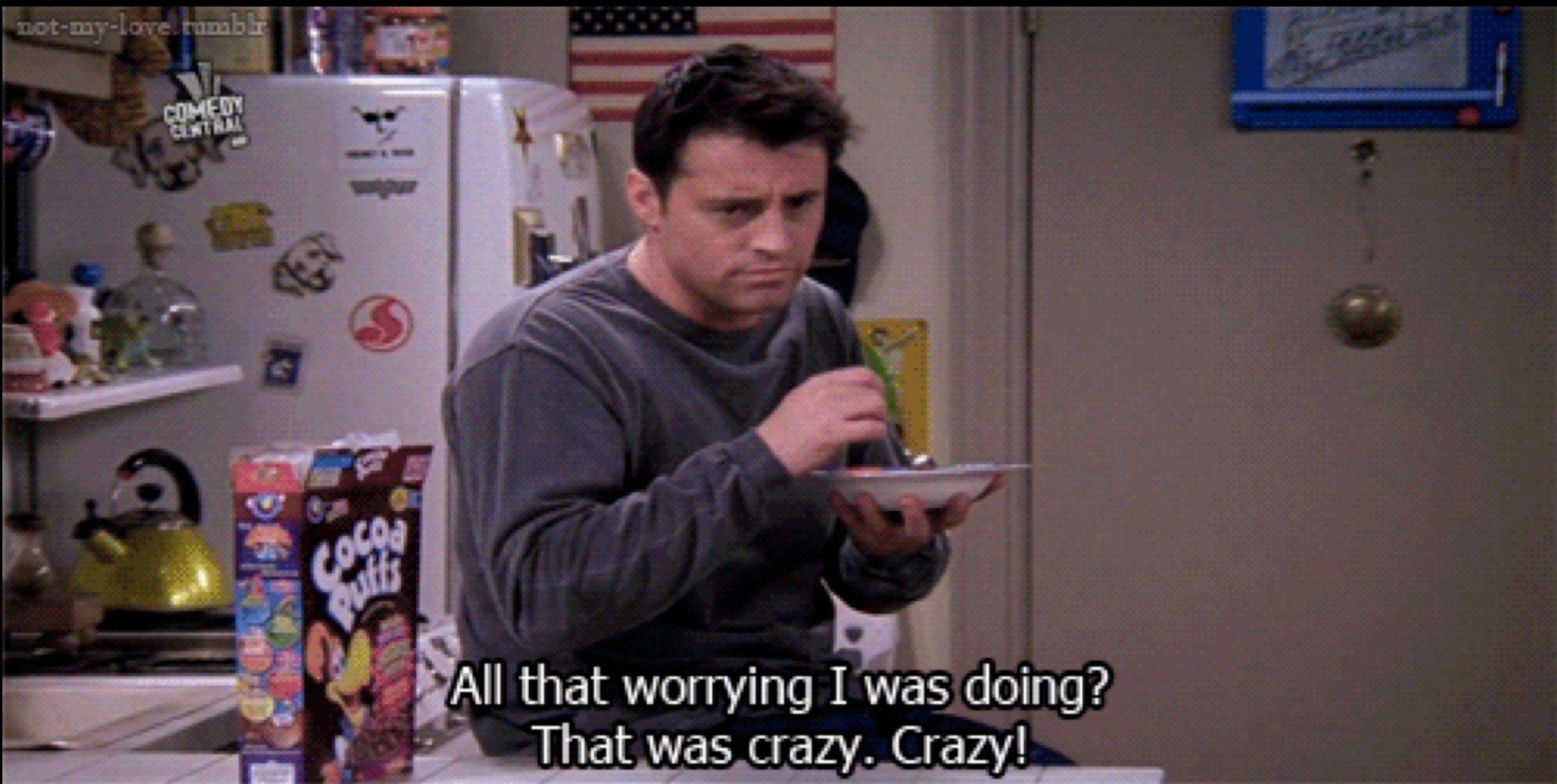


[github.com/sous-chefs](https://github.com/sous-chefs)

@MATTSTRATTON - CHEFCONF 2017

You might not even have ideas of a project to work on.  
Luckily, the sous chefs can help you.  
The work on making community cookbooks better. They  
have assured me they would love your help.

# IT'S ALL GOING TO BE FINE



@MATTSTRATTON - CHEFCONF 2017

This isn't hard. It's not scary. Just don't be an asshole.

# FURTHER RESOURCES

- » [learn.chef.io](http://learn.chef.io) (Open Source Collaboration module)
- » [gitforteams.com](http://gitforteams.com) (Emma Jane Hogbin Westby)
- » <http://codeinthehole.com/tips/a-useful-template-for-commit-messages/>
- » <https://github.com/m1foley/fit-commit>
- » Open Source Governance 101 by Nell Shamrell-Harrington <https://www.youtube.com/watch?v=iBhJBAdPbEE>

# EVEN MORE RESOURCES!

- » [arresteddevops.com](http://arresteddevops.com)
- » [github.com/mattstratton/speaking](https://github.com/mattstratton/speaking)
- » [twitter.com/mattstratton](https://twitter.com/mattstratton)
- » [speakerdeck.com/mattstratton](https://speakerdeck.com/mattstratton)

@MATTSTRATTON - CHEFCONF 2017