We're Building On Hollowed Foundations:

Worrying Trends in Open Source and What You Can Actually Do About It

Heather Miller
@heathercmiller

scalacenter

Northeastern

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lambda Days, February 22nd, 2018

First, where is this perspective coming from?

scalacenter

A **not-for-profit** center established at EPFL

(the Swiss Federal Institute of Technology at Lausanne)

OUR MISSION

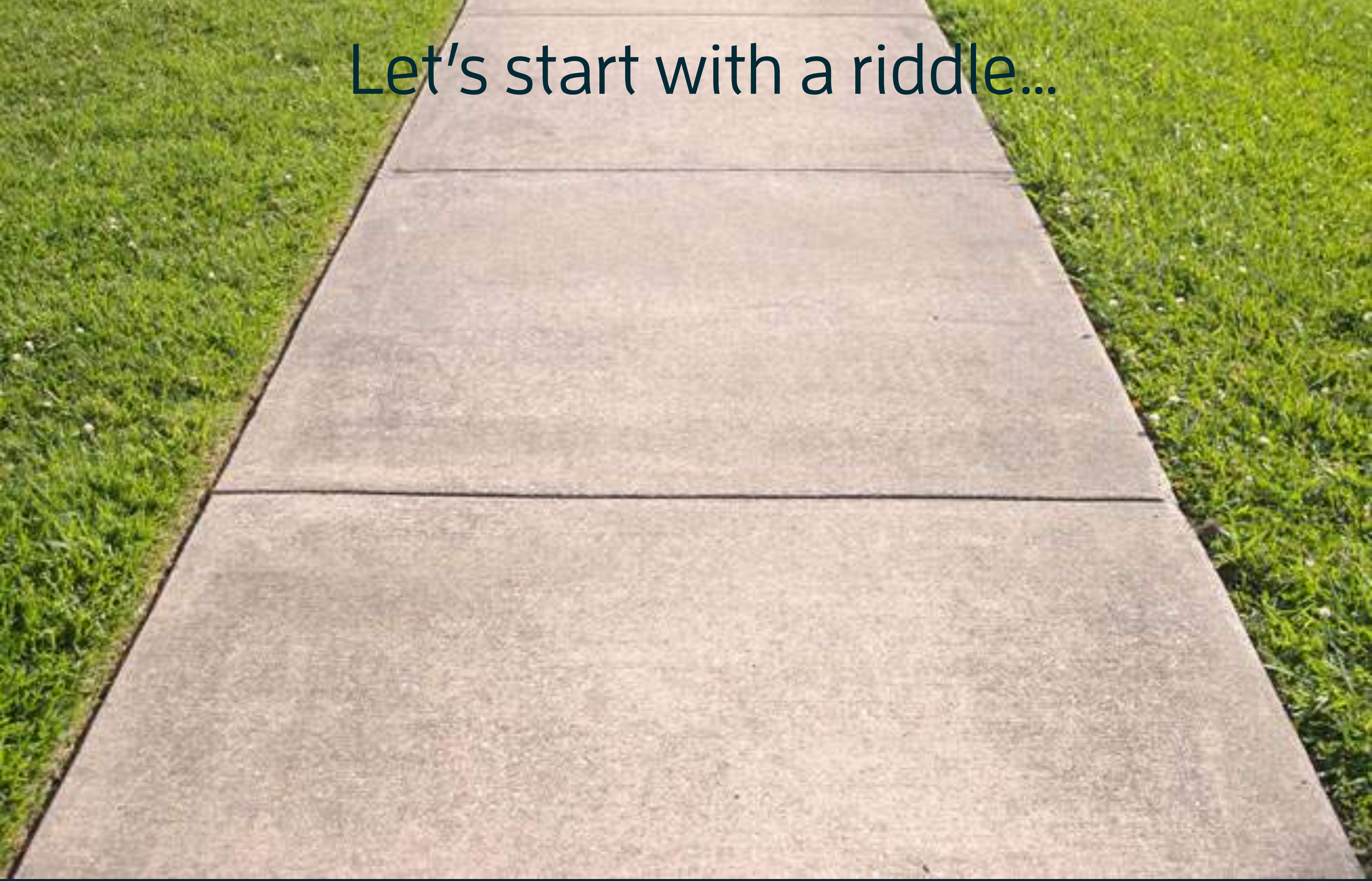
Independently guide and support the entire Scala community.

Coordinate and develop open source libraries and tools for the benefit of the overall Scala community.

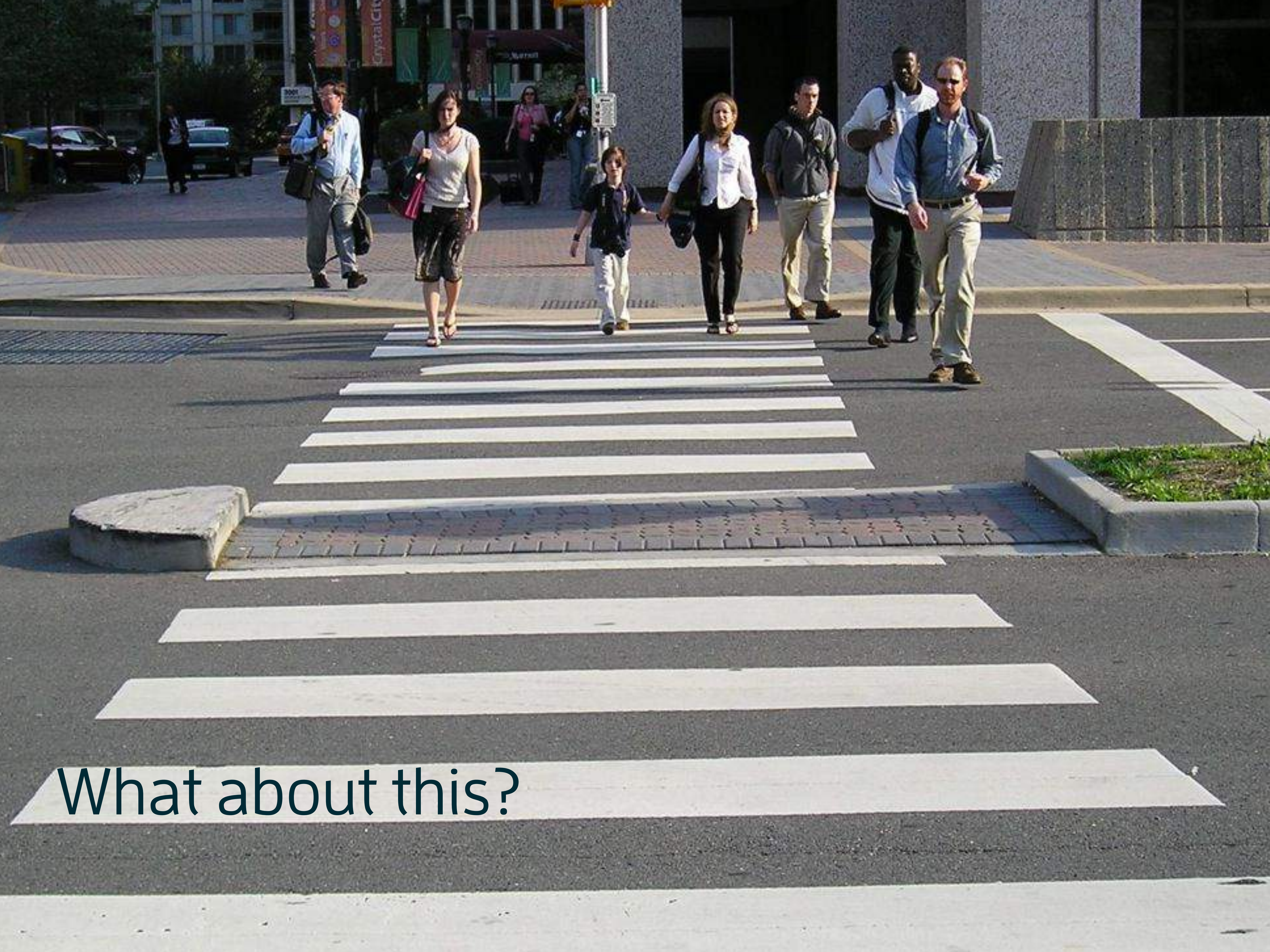
Provide deep, quality, and freely available educational materials for Scala.

Let’s start with a riddle…

Let’s start with a riddle…



What does **this** have to do with Scala?



What about this?

Or this?

These are all examples of
common, shared
physical infrastructure

Anybody can use these things.

The same is true for **digital infrastructure**

free/libre and open source = FLOSS

Open source software is our shared digital infrastructure.

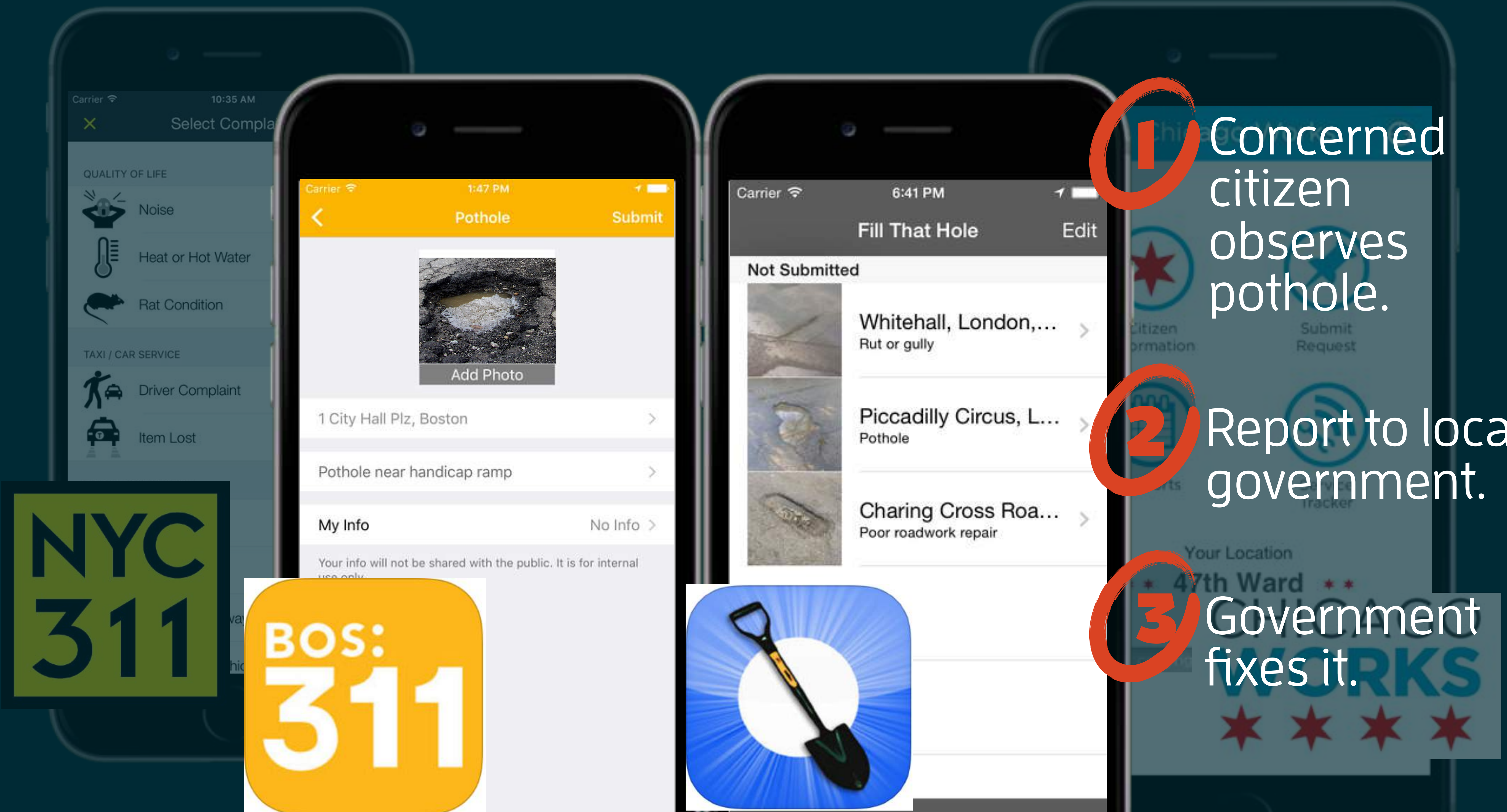
Like physical infrastructure, anybody can use these things.

Back to **physical infrastructure**

What do you do **when this happens to your infrastructure?**

(UK app)

"Fill That Hole contacts the highway authority to get it fixed."



Carrier 10:35 AM

Select Compla

QUALITY OF LIFE

Noise

Heat or Hot Water

Rat Condition

TAXI / CAR SERVICE

Driver Complaint

Item Lost

NYC 311

Carrier 1:47 PM

Pothole Submit

Add Photo

1 City Hall Plz, Boston

Pothole near handicap ramp

My Info No Info

Your info will not be shared with the public. It is for internal use only.

BOS:311

Carrier 6:41 PM

Fill That Hole Edit

Not Submitted

Whitehall, London,…
Rut or gully

Piccadilly Circus, L…
Pothole

Charing Cross Roa…
Poor roadwork repair

1 Concerned citizen observes pothole.

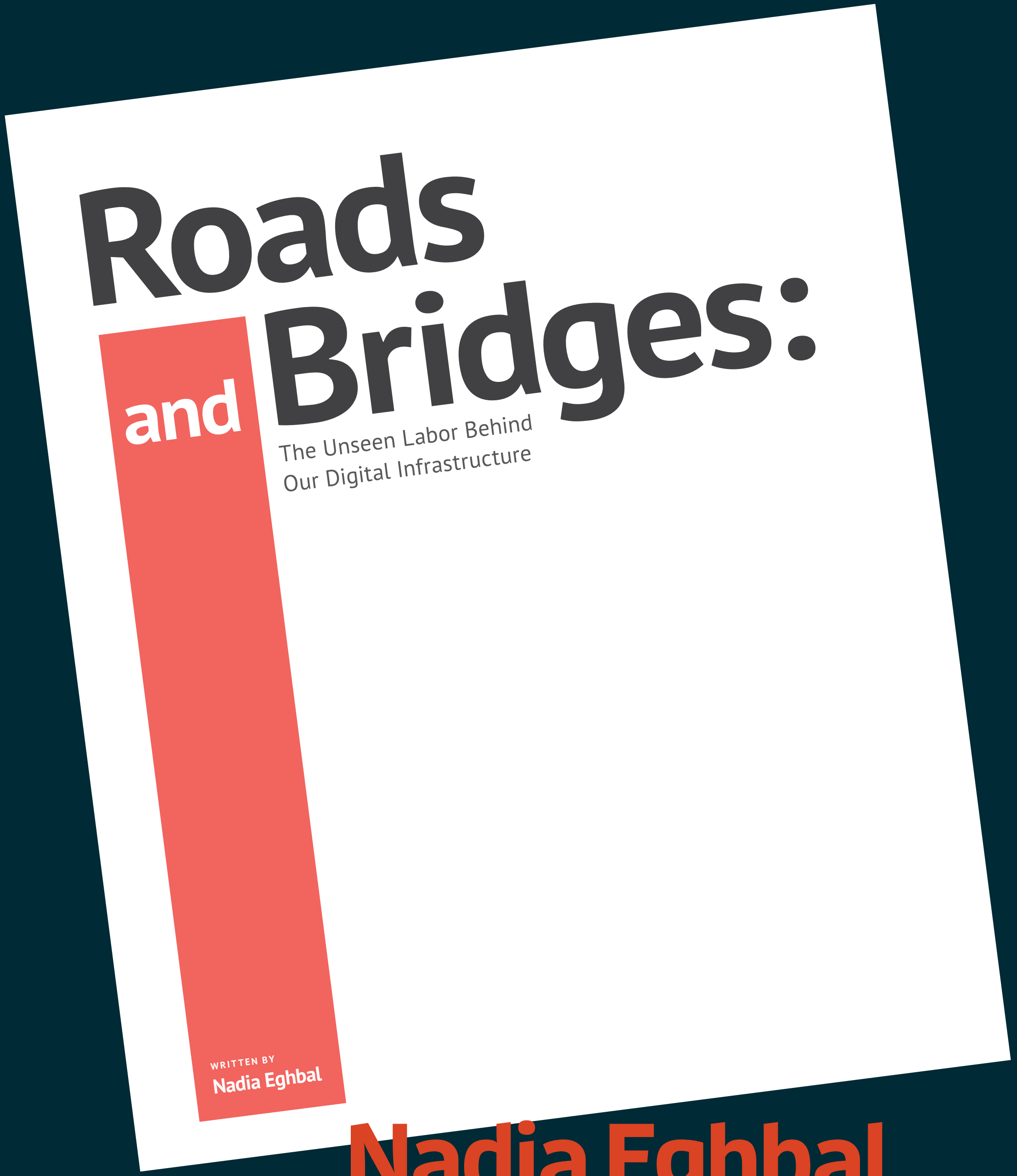
2 Report to loca government.

3 Government fixes it.

What do you do when this happens to your **digital infrastructure**?

It's fundamentally distributed. No central gov't.

Several sources of information for this talk:



Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure

WRITTEN BY Nadia Eghbal

Nadia Eghbal

+ many blog articles, surveys, etc, that will be cited in-line.

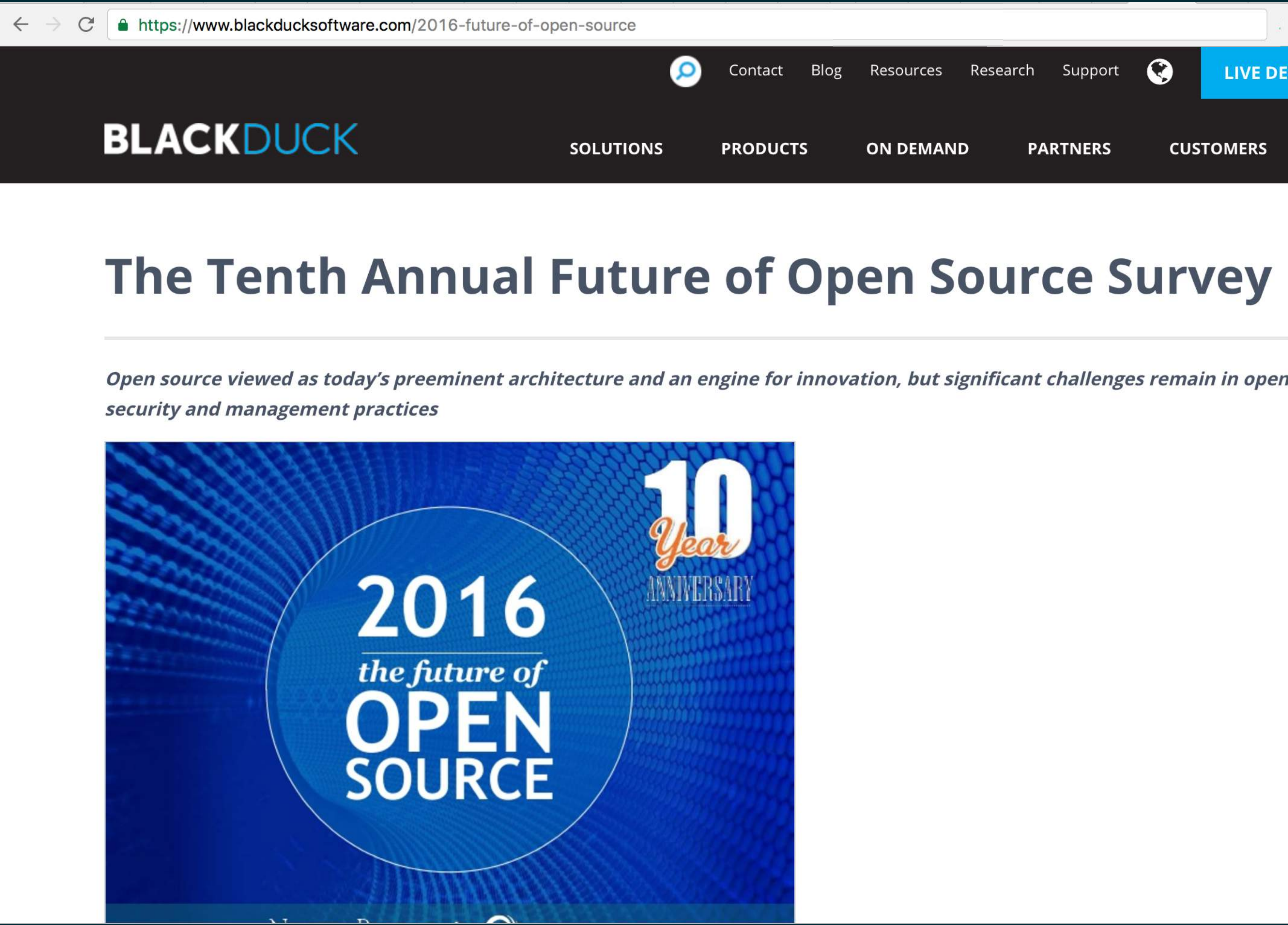
I found many of these references via Nadia's excellent ♥ report!

Remember these days?

Everybody is jumping on the FLOSS bandwagon.

We don’t need to encourage people to use open source anymore.

We need to encourage people to give back to it instead!



The Tenth Annual Future of Open Source Survey

Open source viewed as today's preeminent architecture and an engine for innovation, but significant challenges remain in open security and management practices

https://www.blackducksoftware.com/2016-future-of-open-source

OPEN SOURCE SURVEY

Black Duck 2015 & 2016 "Future of Open Source" Survey

Key points

- Interviewed **1,313** companies *(in 2016)*
  Interviewed **1,240** companies *(in 2015)*

  78% "of companies run on open source" *(in 2015)*

  This is up 2x over 2010!

  COMPANIES ARE DEPENDING MORE AND MORE ON FLOSS

https://www.blackducksoftware.com/2016-future-of-open-source

OPEN SOURCE SURVEY

Black Duck 2015 & 2016 "Future of Open Source" Survey

Key points

- **TOP 3 REASONS TO USE FLOSS** *(in 2016)*
  - **#1** Quality of solutions
  - **#2** Competitive features & technical capabilities
  - **#3** Ability to customize & fix
- 66% of companies consider FLOSS *(in 2015)* options before proprietary alternatives.
  ↖ **FLOSS IS THE DEFAULT CHOICE!**

https://www.blackducksoftware.com/2016-future-of-open-source

OPEN SOURCE SURVEY

Black Duck 2017 Survey

IN 2017, 60% OF COMPANIES SURVEYED INCREASED OPEN SOURCE USAGE.

Main attributed reason: *(in 2017)*

- Low cost with no vendor lock-in.

https://www.blackducksoftware.com/open-source-360deg-survey

FLOSS IS WIDELY VIEWED AS A WAY TO HELP STARTUPS SCALE QUICKLY AND SAVE MONEY

Mark Suster
Entrepreneur & VC

(in 2011)

> When I built my first company starting in 1999 **it cost $2.5 million in infrastructure** just to get started and another $2.5 million in team costs to code, launch, manage, market & sell our software.

> Open source became a movement – a mentality. **Suddenly infrastructure software was nearly free. We paid 10% of the normal costs for the software** and that money was for software support. A 90% disruption in cost spawns innovation – believe me.

https://bothsidesofthetable.com/understanding-changes-in-the-software-venture-capital-industries-b69a7e3a1ec7#.4mlgkd577

FLOSS IS WIDELY VIEWED AS A WAY TO HELP STARTUPS SCALE QUICKLY AND SAVE MONEY

Instagram

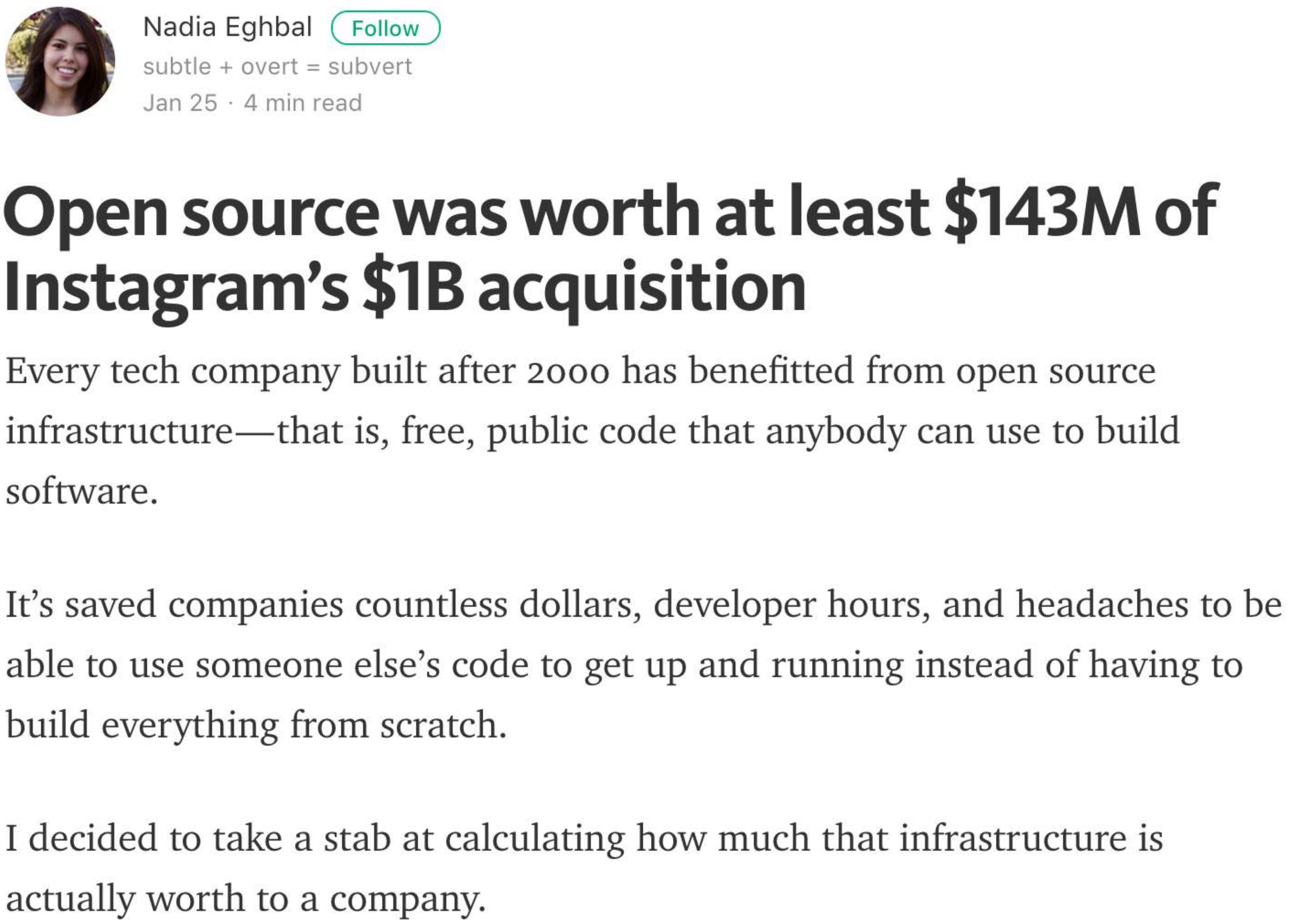
Blog article:
Advice on picking
tech for your startup

Mike Krieger
Instagram co-founder

> “Borrow instead of building whenever possible”

> “There are hundreds of fantastic open-source projects that have been built through the hard experience of creating and scaling companies; especially around infrastructure and monitoring…that can save you time and let you focus on actually building out your product.”

https://opbeat.com/blog/posts/picking-tech-for-your-startup/



Nadia Eghbal
subtle + overt = subvert
Jan 25 · 4 min read

Open source was worth at least $143M of Instagram's $1B acquisition

Every tech company built after 2000 has benefitted from open source infrastructure—that is, free, public code that anybody can use to build software.

It's saved companies countless dollars, developer hours, and headaches to be able to use someone else's code to get up and running instead of having to build everything from scratch.

I decided to take a stab at calculating how much that infrastructure is actually worth to a company.

https://medium.com/@nayafia/open-source-was-worth-at-least-143m-of-instagram-s-1b-acquisition-808bb85e4681#.d6gzzr9nk

Meanwhile…

Same # or less contributors.

OpenSSL
Cryptography and SSL/TLS Toolkit

In 2014,

66% of all web servers were using OpenSSL[1]

Meanwhile, OpenSSL was maintained by only a few volunteers

> “Steve Marquess, noticed that one contributor, Stephen Henson, was working full time on OpenSSL. Curious, Marquess asked him what he did for income, and was shocked to learn that Henson made one-fifth of Marquess’s salary.”[2]

> “Marquess had always considered himself to be a strong programmer, but his skills paled in comparison to Henson’s. … Henson had been working on OpenSSL since 1998.”[2]

[1]: https://news.netcraft.com/archives/2014/04/08/half-a-million-widely-trusted-websites-vulnerable-to-heartbleed-bug.html

[2]: https://fordfoundcontent.blob.core.windows.net/media/2976/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure.pdf

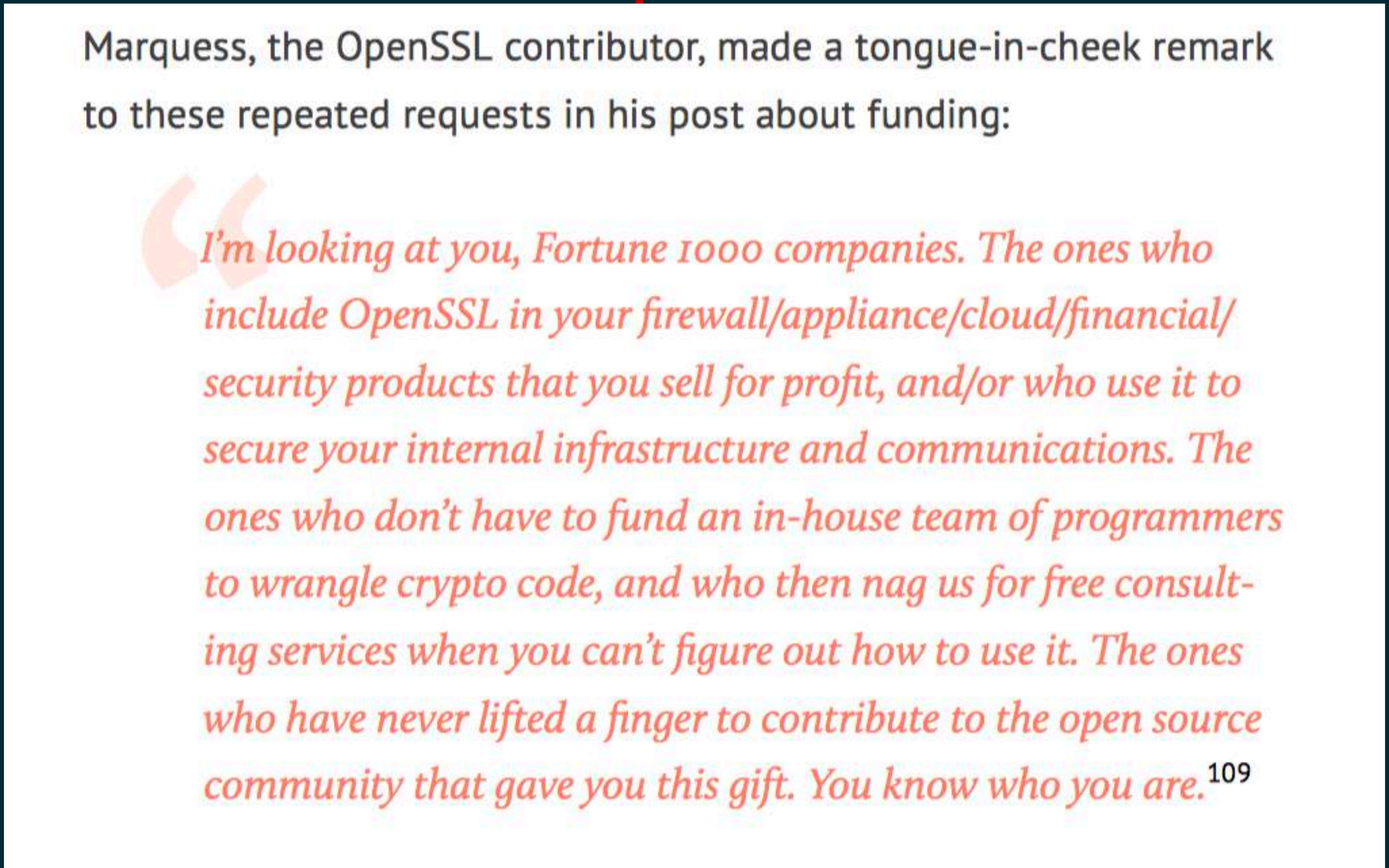
> “I had always assumed, (as had the rest of the world) that the OpenSSL team was large, active, and well resourced.”[2]
>
> – Steve Marquess

In reality, OpenSSL wasn’t even able to support one person’s work.[2]

AND YET, INDUSTRY, GOVERNMENT, ETC ARE LARGELY UNAWARE OF INFRASTRUCTURE’S FUNDING ISSUES.

[2]: https://fordfoundcontent.blob.core.windows.net/media/2976/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure.pdf

OpenSSL
Cryptography and SSL/TLS Toolkit

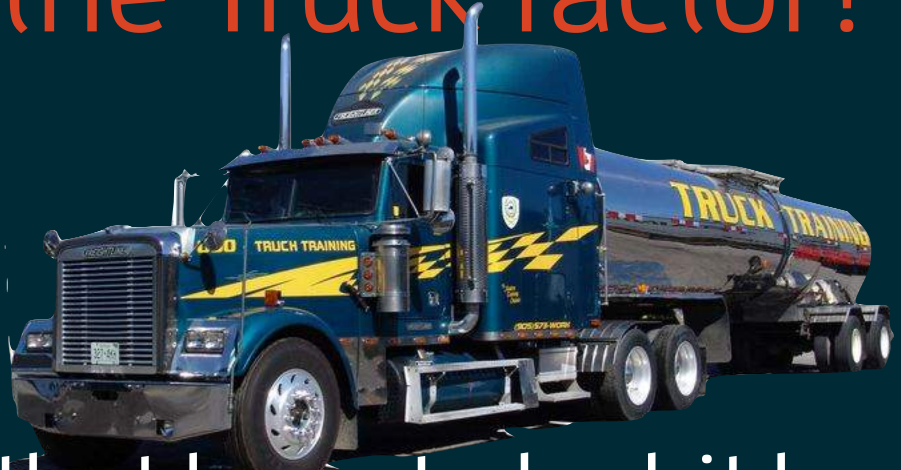


Marquess, the OpenSSL contributor, made a tongue-in-cheek remark to these repeated requests in his post about funding:

> *I'm looking at you, Fortune 1000 companies. The ones who include OpenSSL in your firewall/appliance/cloud/financial/security products that you sell for profit, and/or who use it to secure your internal infrastructure and communications. The ones who don't have to fund an in-house team of programmers to wrangle crypto code, and who then nag us for free consulting services when you can't figure out how to use it. The ones who have never lifted a finger to contribute to the open source community that gave you this gift. You know who you are.*[109]

http://veridicalsystems.com/ blog/of-money-responsibility-and-pride/

It gets worse.
Have you ever heard of the Truck factor?



The Truck Factor:

the minimal # of developers that have to be hit by a truck (or quit) before a project is incapacitated

- Look at the 133 most active projects on GitHub
- Determine the amount of information concentrated in individual team members from commits.

Result:

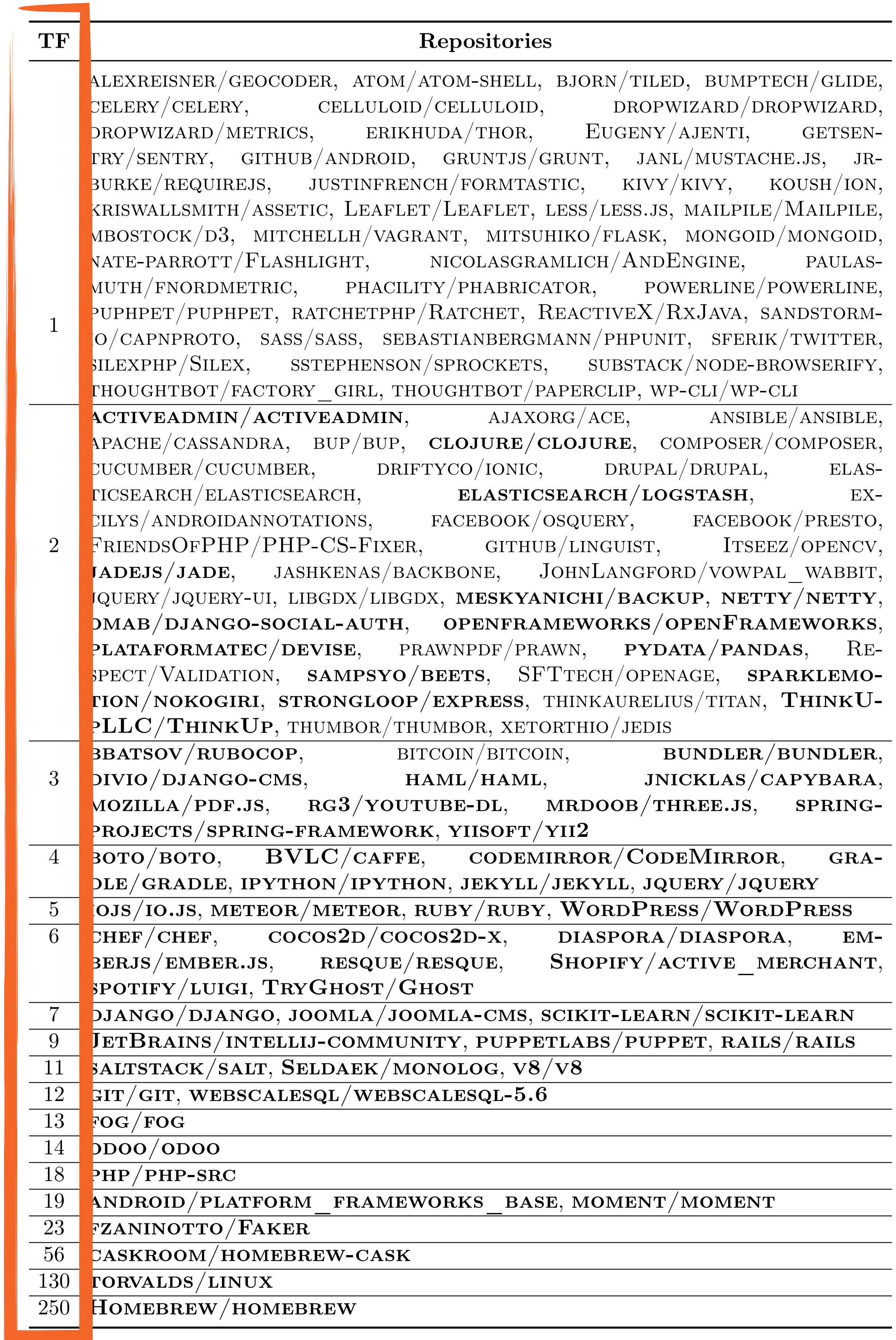
64% OF PROJECTS RELIED ON 1-2 DEVS TO SURVIVE.

https://peerj.com/preprints/1233.pdf

The Truck Factor Results:

The higher the TF the better!

Only a handful of projects with a high TF...



TF	Repositories
1	alexreisner/geocoder, atom/atom-shell, bjorn/tiled, bumptech/glide, celery/celery, celluloid/celluloid, dropwizard/dropwizard, dropwizard/metrics, erikhuda/thor, Eugeny/ajenti, getsentry/sentry, github/android, gruntjs/grunt, janl/mustache.js, jrburke/requirejs, justinfrench/formtastic, kivy/kivy, koush/ion, kriswallsmith/assetic, Leaflet/Leaflet, less/less.js, mailpile/Mailpile, mbostock/d3, mitchellh/vagrant, mitsuhiko/flask, mongoid/mongoid, nate-parrott/Flashlight, nicolasgramlich/AndEngine, paulasmuth/fnordmetric, phacility/phabricator, powerline/powerline, puphpet/puphpet, ratchetphp/Ratchet, ReactiveX/RxJava, sandstormio/capnproto, sass/sass, sebastianbergmann/phpunit, sferik/twitter, silexphp/Silex, sstephenson/sprockets, substack/node-browserify, thoughtbot/factory_girl, thoughtbot/paperclip, wp-cli/wp-cli
2	activeadmin/activeadmin, ajaxorg/ace, ansible/ansible, apache/cassandra, bup/bup, clojure/clojure, composer/composer, cucumber/cucumber, driftyco/ionic, drupal/drupal, elasticsearch/elasticsearch, elasticsearch/logstash, excilys/androidannotations, facebook/osquery, facebook/presto, FriendsOfPHP/PHP-CS-Fixer, github/linguist, Itseez/opencv, jadejs/jade, jashkenas/backbone, JohnLangford/vowpal_wabbit, jquery/jquery-ui, libgdx/libgdx, meskyanichi/backup, netty/netty, omab/django-social-auth, openframeworks/openFrameworks, plataformatec/devise, prawnpdf/prawn, pydata/pandas, Respect/Validation, sampsyo/beets, SFTtech/openage, sparklemotion/nokogiri, strongloop/express, thinkaurelius/titan, ThinkUpLLC/ThinkUp, thumbor/thumbor, xetorthio/jedis
3	bbatsov/rubocop, bitcoin/bitcoin, bundler/bundler, divio/django-cms, haml/haml, jnicklas/capybara, mozilla/pdf.js, rg3/youtube-dl, mrdoob/three.js, spring-projects/spring-framework, yiisoft/yii2
4	boto/boto, BVLC/caffe, codemirror/CodeMirror, gradle/gradle, ipython/ipython, jekyll/jekyll, jquery/jquery
5	iojs/io.js, meteor/meteor, ruby/ruby, WordPress/WordPress
6	chef/chef, cocos2d/cocos2d-x, diaspora/diaspora, emberjs/ember.js, resque/resque, Shopify/active_merchant, spotify/luigi, TryGhost/Ghost
7	django/django, joomla/joomla-cms, scikit-learn/scikit-learn
9	JetBrains/intellij-community, puppetlabs/puppet, rails/rails
11	saltstack/salt, Seldaek/monolog, v8/v8
12	git/git, webscalesql/webscalesql-5.6
13	fog/fog
14	odoo/odoo
18	php/php-src
19	android/platform_frameworks_base, moment/moment
23	fzaninotto/Faker
56	caskroom/homebrew-cask
130	torvalds/linux
250	Homebrew/homebrew

The Truck Factor Results:
A sampling of some low truck factors...

Truck Factor 1:

gruntjs/grunt

wp-cli/wp-cli

sass/sass

mbostock/d3

ReactiveX/RxJava

Truck Factor 2:

apache/cassandra

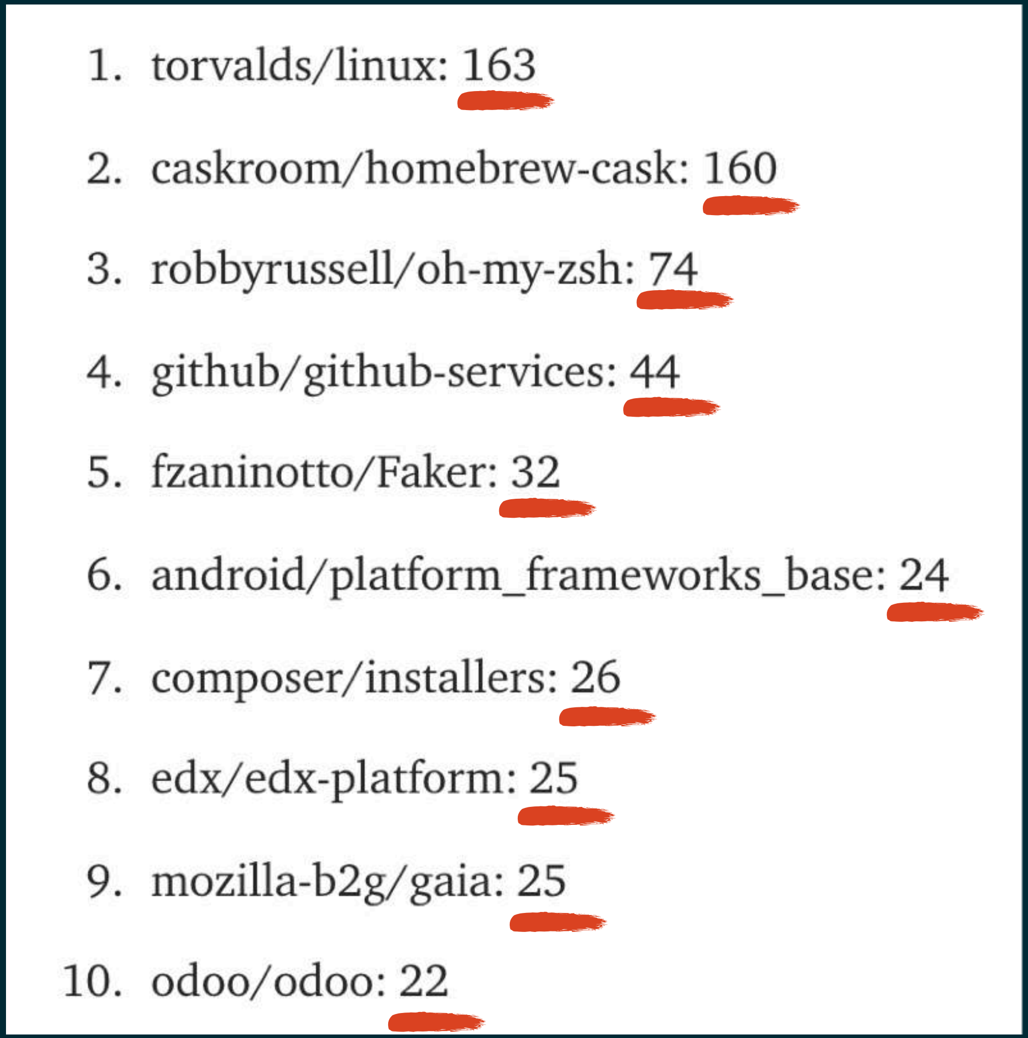
clojure/clojure

pydata/pandas

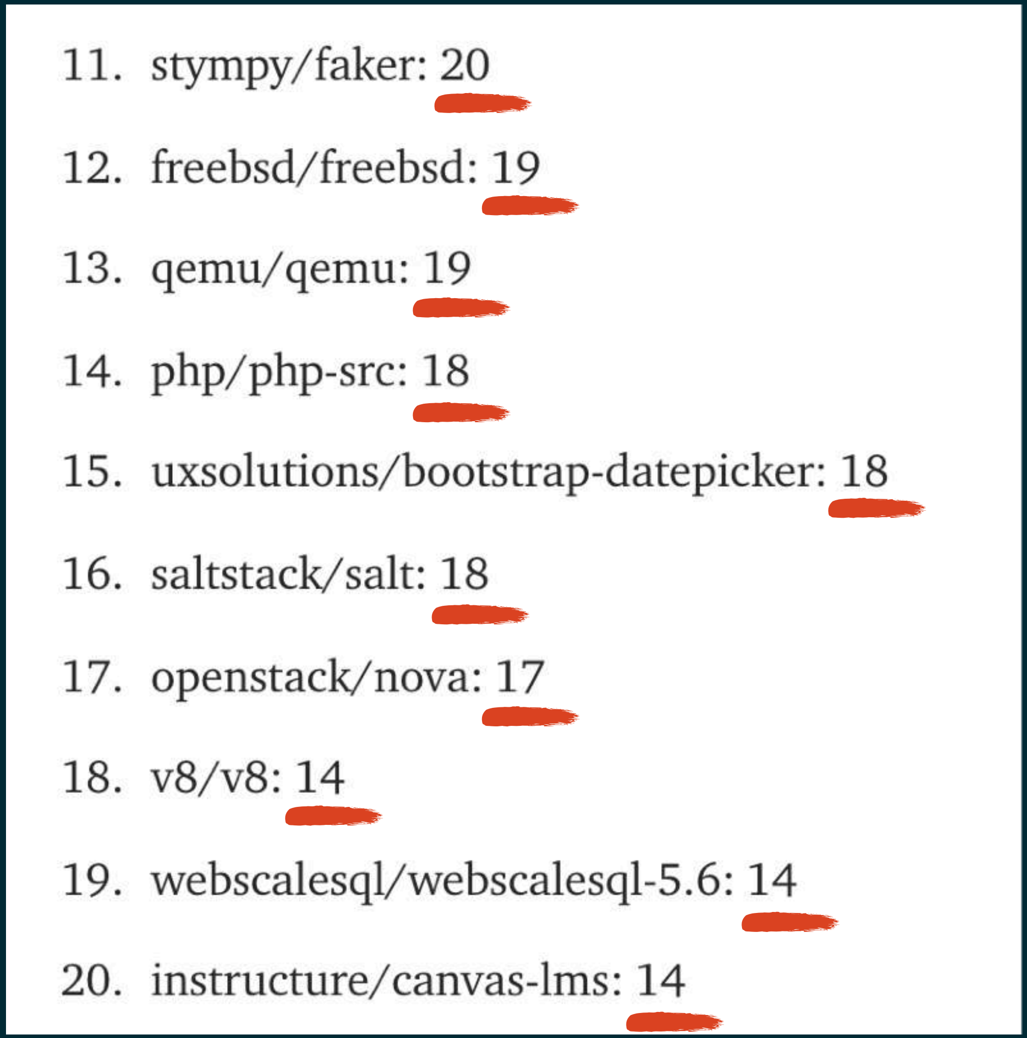
netty/netty

drupal/drupal

The Truck Factor Results:
There exist high truck factors too!



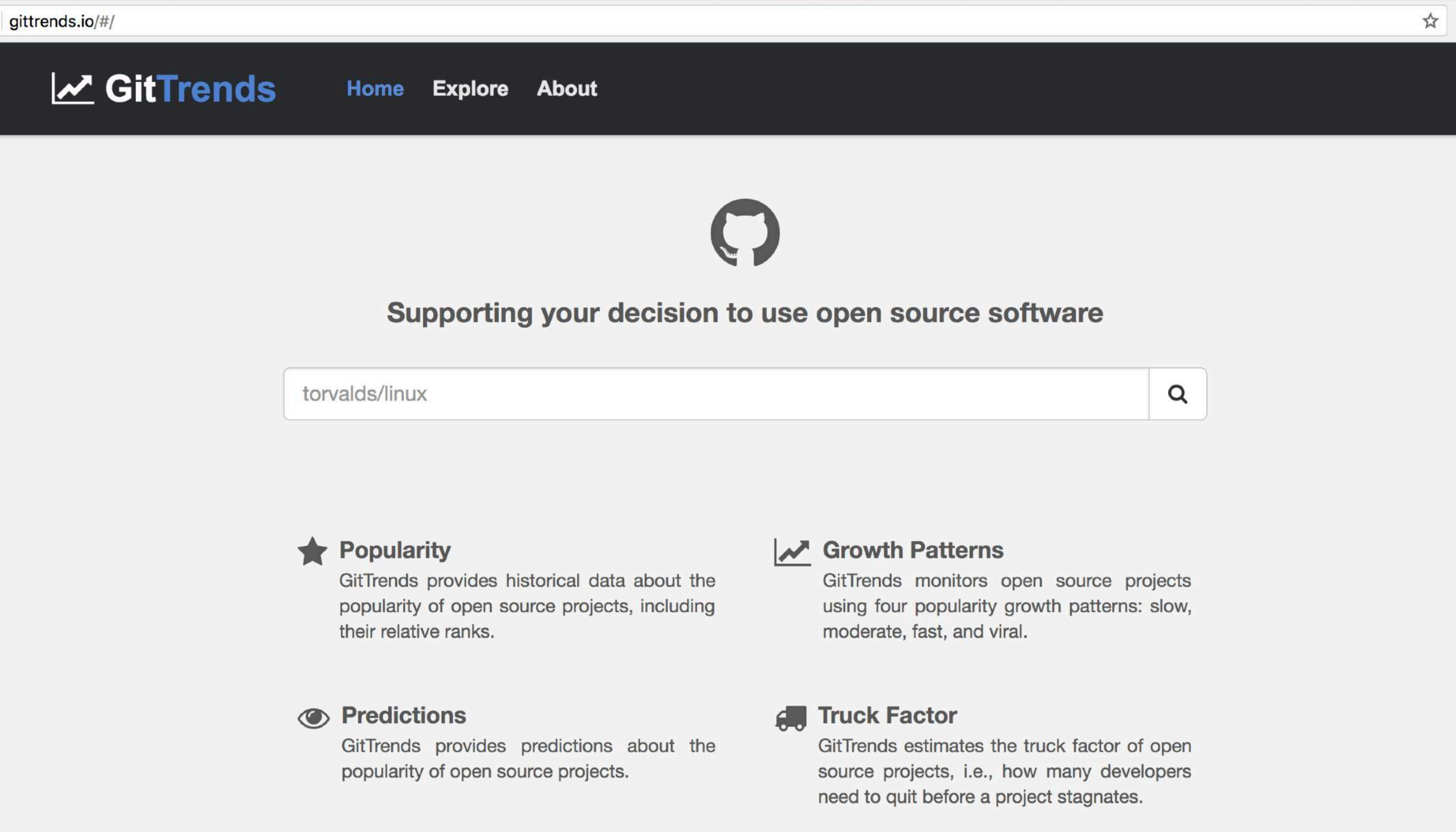
1. torvalds/linux: 163
2. caskroom/homebrew-cask: 160
3. robbyrussell/oh-my-zsh: 74
4. github/github-services: 44
5. fzaninotto/Faker: 32
6. android/platform_frameworks_base: 24
7. composer/installers: 26
8. edx/edx-platform: 25
9. mozilla-b2g/gaia: 25
10. odoo/odoo: 22



11. stympy/faker: 20
12. freebsd/freebsd: 19
13. qemu/qemu: 19
14. php/php-src: 18
15. uxsolutions/bootstrap-datepicker: 18
16. saltstack/salt: 18
17. openstack/nova: 17
18. v8/v8: 14
19. webscalesql/webscalesql-5.6: 14
20. instructure/canvas-lms: 14

The Truck Factor Results:

Check the calculated truck factor of your project.



gittrends.io/#/

GitTrends

Home Explore About

Supporting your decision to use open source software

torvalds/linux

★ Popularity

GitTrends provides historical data about the popularity of open source projects, including their relative ranks.

Growth Patterns

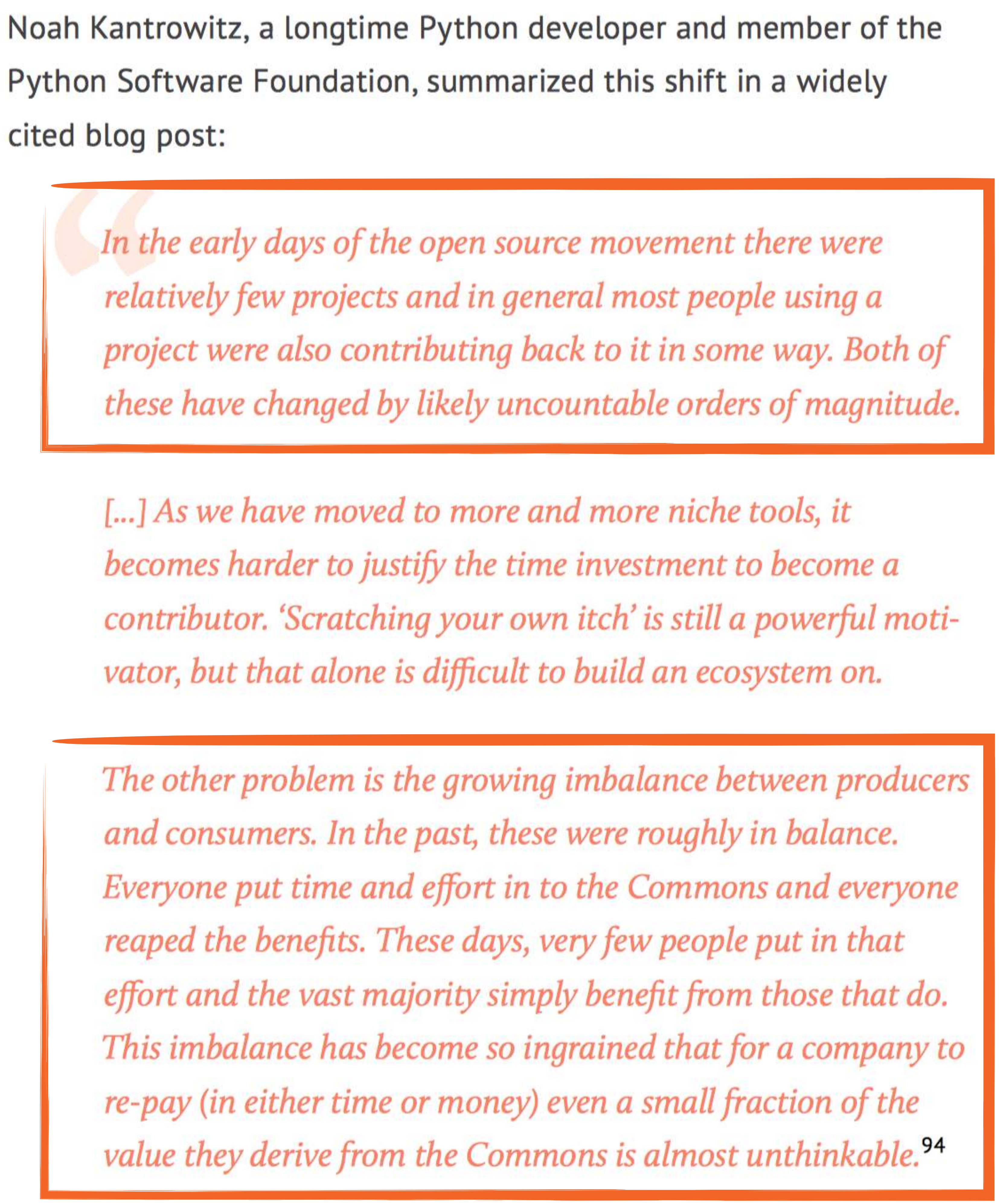
GitTrends monitors open source projects using four popularity growth patterns: slow, moderate, fast, and viral.

Predictions

GitTrends provides predictions about the popularity of open source projects.

Truck Factor

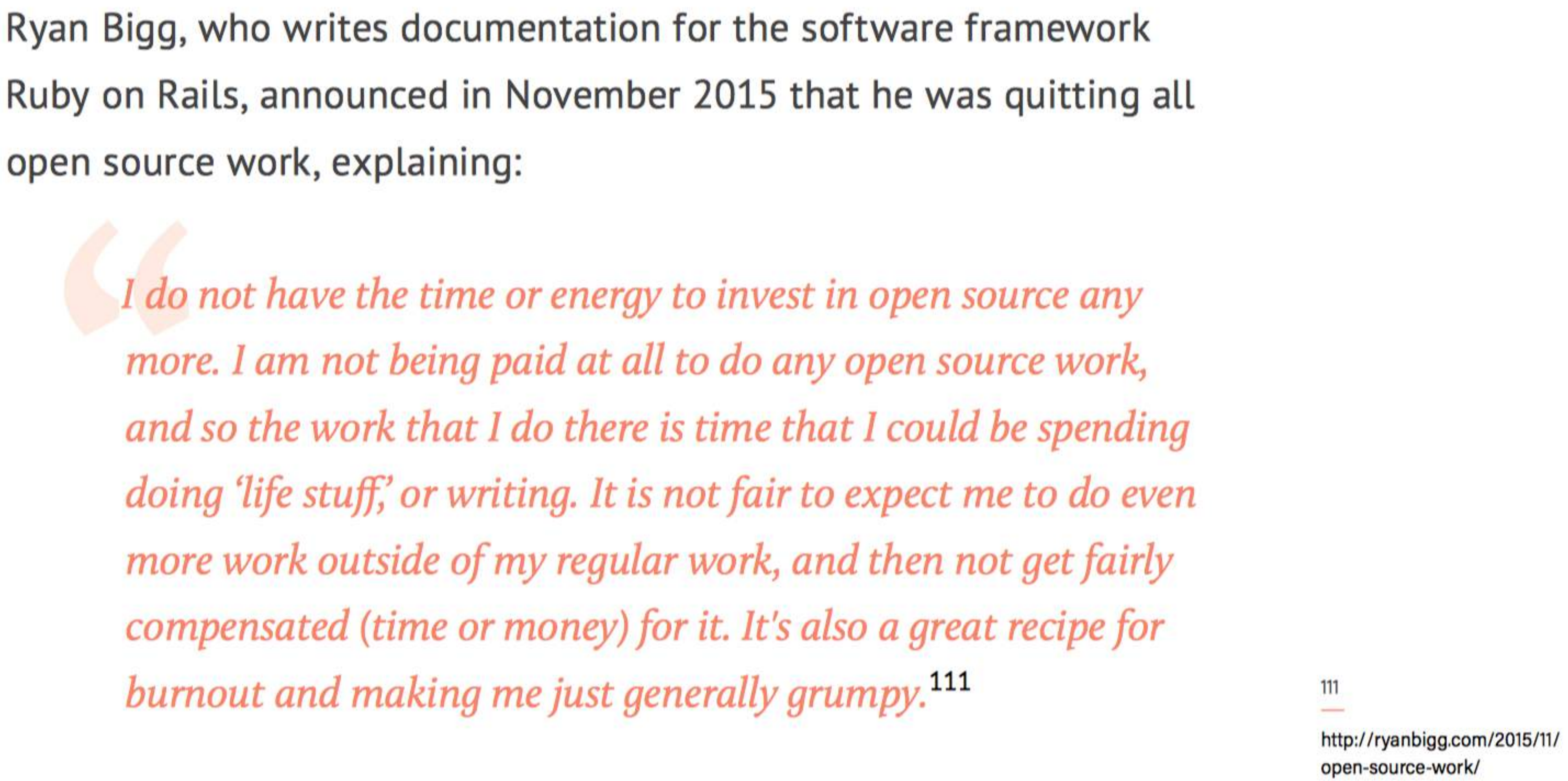
GitTrends estimates the truck factor of open source projects, i.e., how many developers need to quit before a project stagnates.



Noah Kantrowitz, a longtime Python developer and member of the Python Software Foundation, summarized this shift in a widely cited blog post:

> *In the early days of the open source movement there were relatively few projects and in general most people using a project were also contributing back to it in some way. Both of these have changed by likely uncountable orders of magnitude.*
>
> *[...] As we have moved to more and more niche tools, it becomes harder to justify the time investment to become a contributor. 'Scratching your own itch' is still a powerful motivator, but that alone is difficult to build an ecosystem on.*
>
> *The other problem is the growing imbalance between producers and consumers. In the past, these were roughly in balance. Everyone put time and effort in to the Commons and everyone reaped the benefits. These days, very few people put in that effort and the vast majority simply benefit from those that do. This imbalance has become so ingrained that for a company to re-pay (in either time or money) even a small fraction of the value they derive from the Commons is almost unthinkable.*[94]

From Roads
& Bridges



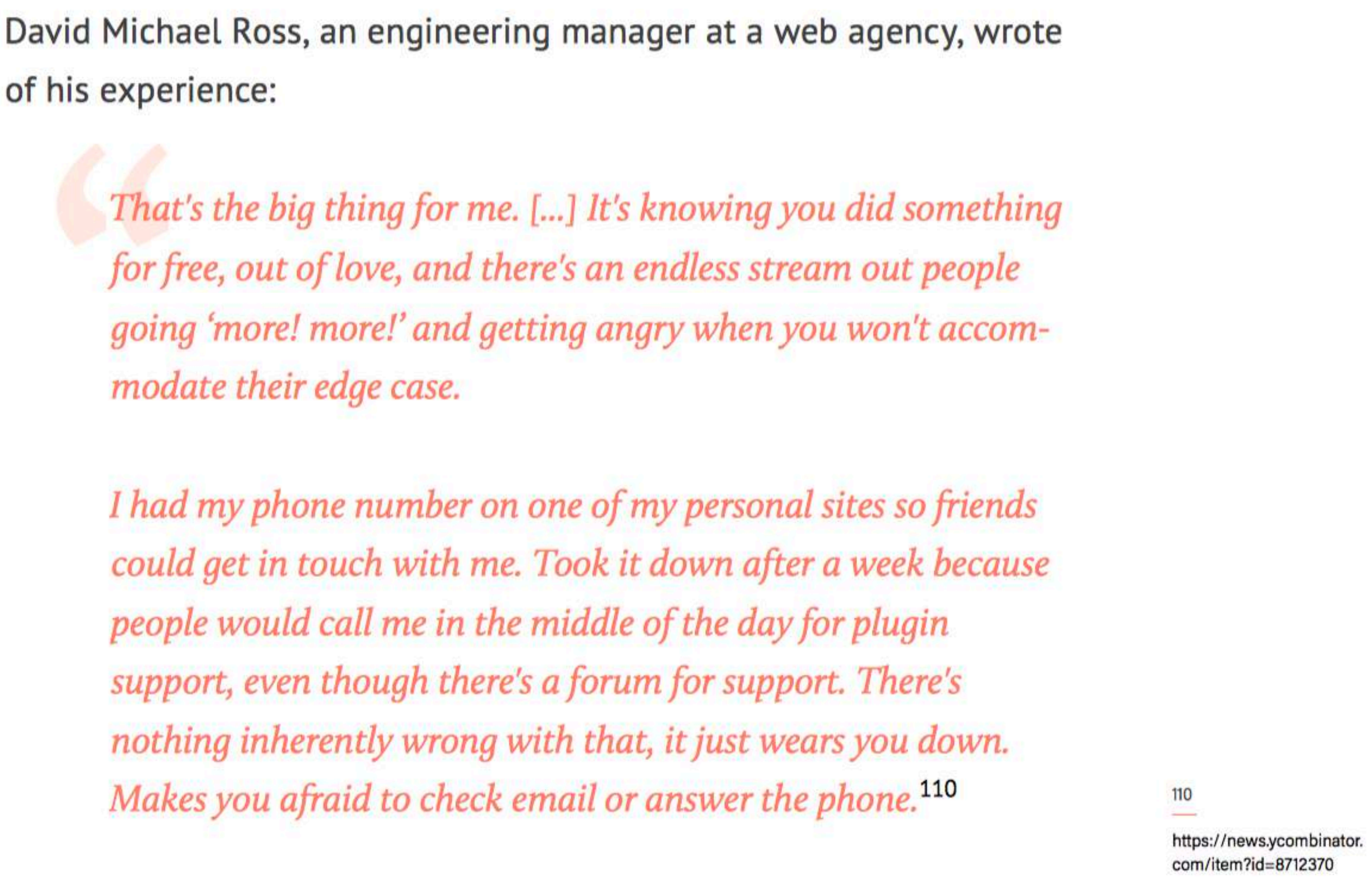
Ryan Bigg, who writes documentation for the software framework Ruby on Rails, announced in November 2015 that he was quitting all open source work, explaining:

> *I do not have the time or energy to invest in open source any more. I am not being paid at all to do any open source work, and so the work that I do there is time that I could be spending doing 'life stuff,' or writing. It is not fair to expect me to do even more work outside of my regular work, and then not get fairly compensated (time or money) for it. It's also a great recipe for burnout and making me just generally grumpy.*[111]

[111] http://ryanbigg.com/2015/11/open-source-work/

MAINTAINERS STRAINED!

From Roads
& Bridges



David Michael Ross, an engineering manager at a web agency, wrote of his experience:

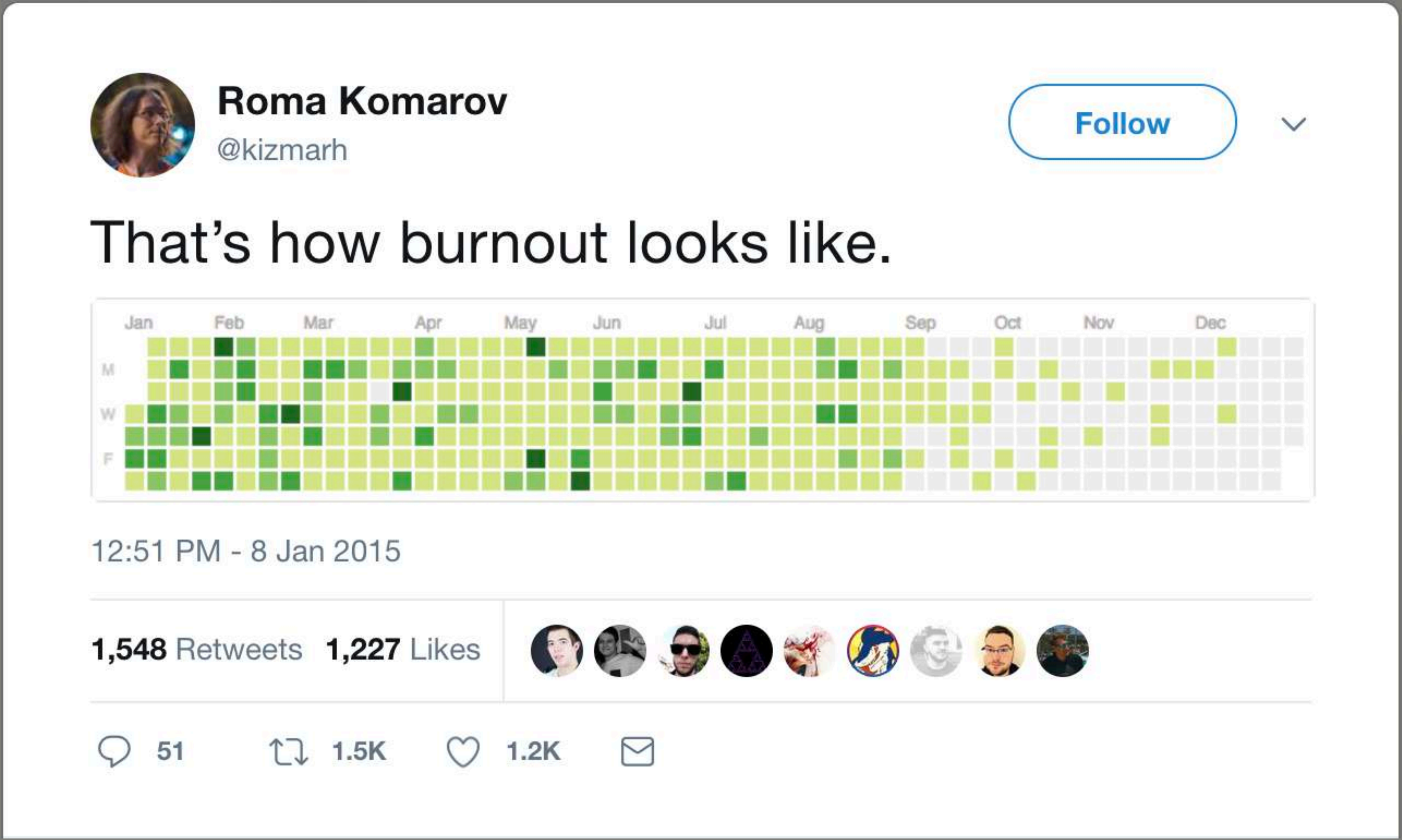
> *That's the big thing for me. [...] It's knowing you did something for free, out of love, and there's an endless stream out people going 'more! more!' and getting angry when you won't accommodate their edge case.*
>
> *I had my phone number on one of my personal sites so friends could get in touch with me. Took it down after a week because people would call me in the middle of the day for plugin support, even though there's a forum for support. There's nothing inherently wrong with that, it just wears you down. Makes you afraid to check email or answer the phone.*[110]

[110] https://news.ycombinator.com/item?id=8712370

MAINTAINERS STRAINED!

From Roads
& Bridges

Burnout happens. A lot.



Roma Komarov
@kizmarh

Follow

That’s how burnout looks like.

12:51 PM - 8 Jan 2015

1,548 Retweets **1,227** Likes

51 1.5K 1.2K

On that note…

Going FLOSS-crazy

All the cool kids are doing it:

US Dept of Defense:

A 2003 report showed that the US DoD was a major user of FLOSS. FLOSS even being used in mission-critical situations.

Report concluded:

- Unexpectedly, DoD security depends heavily on FLOSS.
- Don't ban FLOSS at the DoD!
- Instead, promote promote broader and more effective use of FLOSS at the DoD. **Moar!**
- "FOSS applications tend to be much lower in cost than their proprietary equivalents, yet they often provide high levels of functionality with good user acceptance."

http://dodcio.defense.gov/Portals/0/Documents/FOSS/dodfoss_pdf.pdf

Is the US government secretly writing software for firing missiles that is built on a FLOSS project that a maintainer has ragequit from?

Ok.

A few trends.

- Using FLOSS over proprietary is now mainstream.
- Startups, enterprise, and even governments are depending on this digital infrastructure.
- Orders of magnitude more users, same or fewer contributors.

MOST USERS OF FLOSS UNAWARE OF THESE TRENDS!

How can we reverse this?

Usefully funding a FLOSS project can be difficult!

Can’t just throw money at it without any coordination/planning/organization…

- There may not be any maintainer that could work more hours (even if paid) on the project.
- Contributed amount might be too small alone to make an impact.
- etc.

Many funding models tried! Mixed success.

By and large, the most common method of funding open source is via **subsidy**.

Two kinds:

personal labor subsidy: dev has a day job and works on FLOSS exclusively in their spare time.

corporate labor subsidy: FLOSS work funded by an employer either as a side-effect or as explicit allocation of time to FLOSS project.

(LESS FREQUENT)

https://blog.tidelift.com/open-source-whos-paying-the-bills

Many funding models tried! Mixed success.

Other models include:

Patronage. E.g., one-off contributions, or contributions via something like Patreon. Also includes personal sponsorship, corporate patronage, grants.

Products & Services. Commercial distribution & assurances, hosted services, consulting, merchandising/events/sponsorship.

https://blog.tidelift.com/open-source-whos-paying-the-bills

WHICH IS THE MOST SUSTAINABLE?

The jury is still out on this!

But if I had to bet my money, I'd bet on:

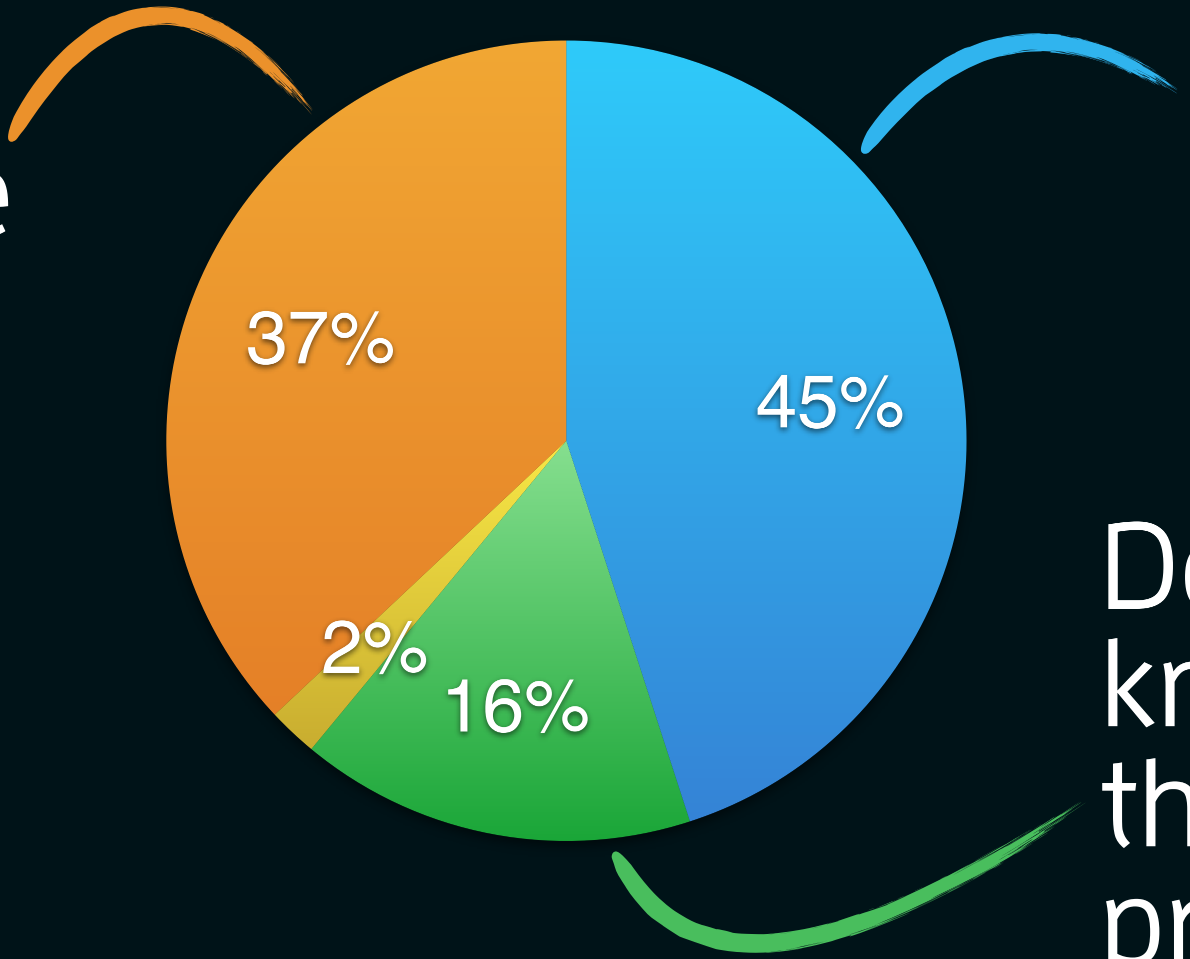
CORPORATE PATRONAGE

We make money off of software we obtained for free *(but which someone else paid for)*.

The least we could do is help one of these projects we depend on with a fix here and there, and we should do it on company time. After all, it's the company who profited.

One small issue though... 🤔

Most organizations don’t have a process for managing open source or are unaware of one in their org.



Yes, there is some kind of process

No process at all.

Don’t know if there’s a process

https://www.blackducksoftware.com/open-source-360deg-survey

Toward corporate-supported FLOSS work

Maybe you are a small contracting shop, and maybe it's easier (organizationally) to dedicate energy to FLOSS.

Or maybe you're part of a large organization with many rules and processes.

Change has to start somewhere.

The FLOSS you use and depend on every day deserves

What we can do right away, without the approval of a legal department?

We can teach people how to participate in open source projects.

Teaching Open Source:
Open Source Sprees

Soon: general recipe for these posted on http://sprees.io

Teaching Open Source

"Open Source Sprees"

Goal:

Portable, repeatable. For any library/project/ecosystem.

Ideal for running for half a day, before/during/after a conference. All you need are interested attendees and some folks who are committers to an open source project.

Also useful as a recipe for meetup group organizers to follow in their cities.

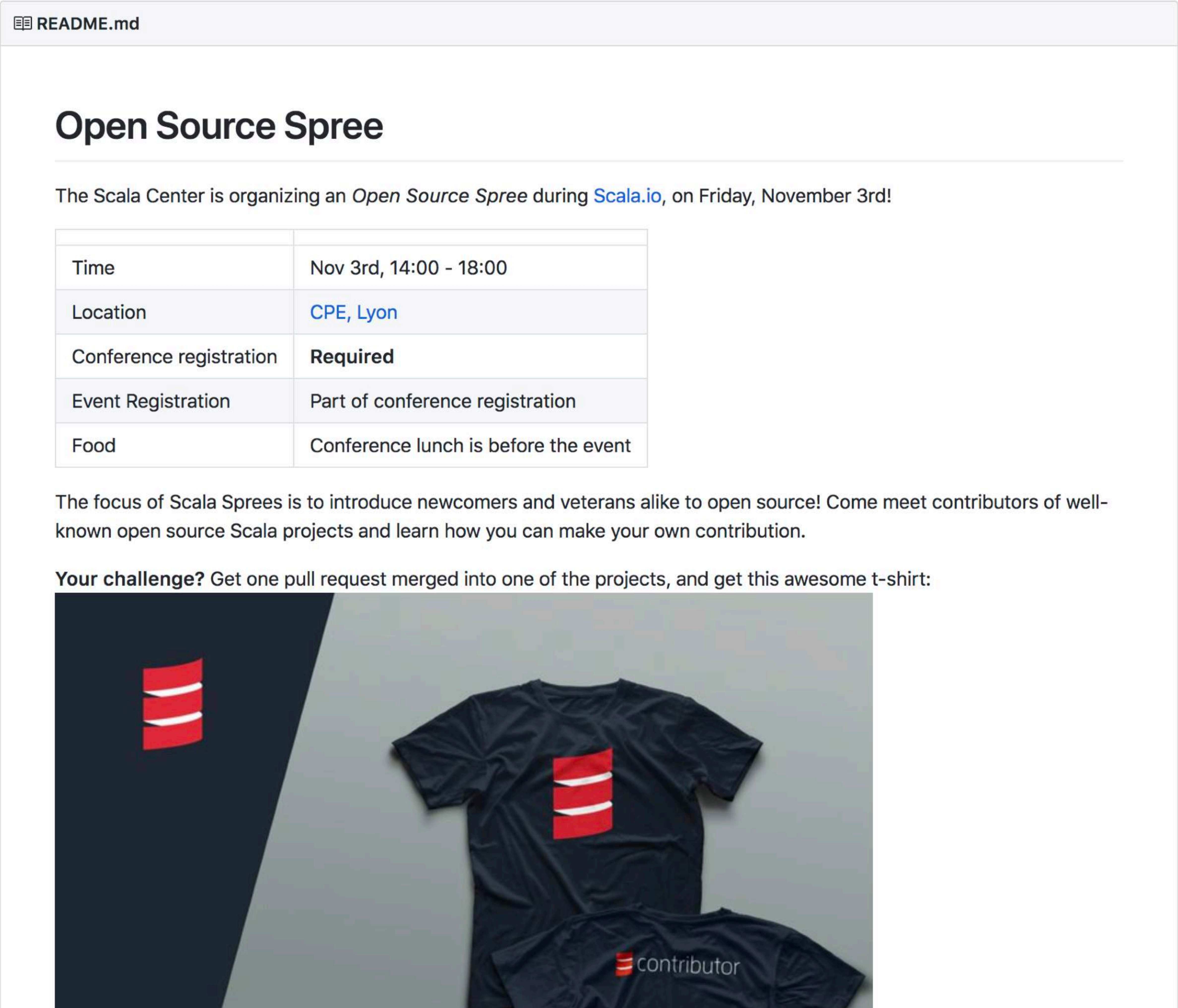
Open Source Sprees

Community-building exercise

Most importantly:

Structure and process for bringing people together to work on open source projects

- Library author in the room
- Library author curates ~5 todos, accomplishable in 3hrs by newcomers ahead of time
- Goal of participants: 1 PR merged by end of spree
- Small prize for those that make PRs!



README.md

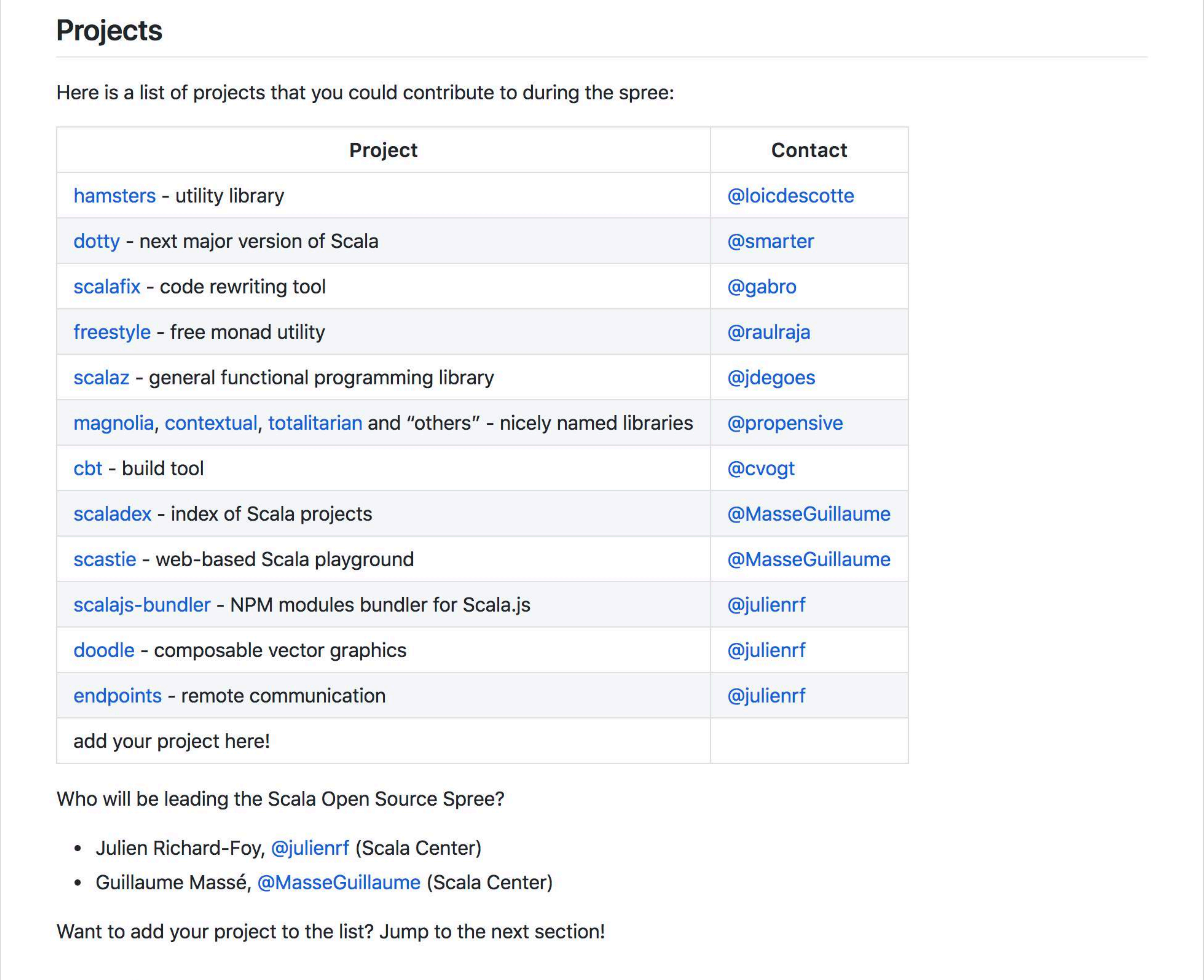
Open Source Spree

The Scala Center is organizing an *Open Source Spree* during Scala.io, on Friday, November 3rd!

Time	Nov 3rd, 14:00 - 18:00
Location	CPE, Lyon
Conference registration	**Required**
Event Registration	Part of conference registration
Food	Conference lunch is before the event

The focus of Scala Sprees is to introduce newcomers and veterans alike to open source! Come meet contributors of well-known open source Scala projects and learn how you can make your own contribution.

Your challenge? Get one pull request merged into one of the projects, and get this awesome t-shirt:



Projects

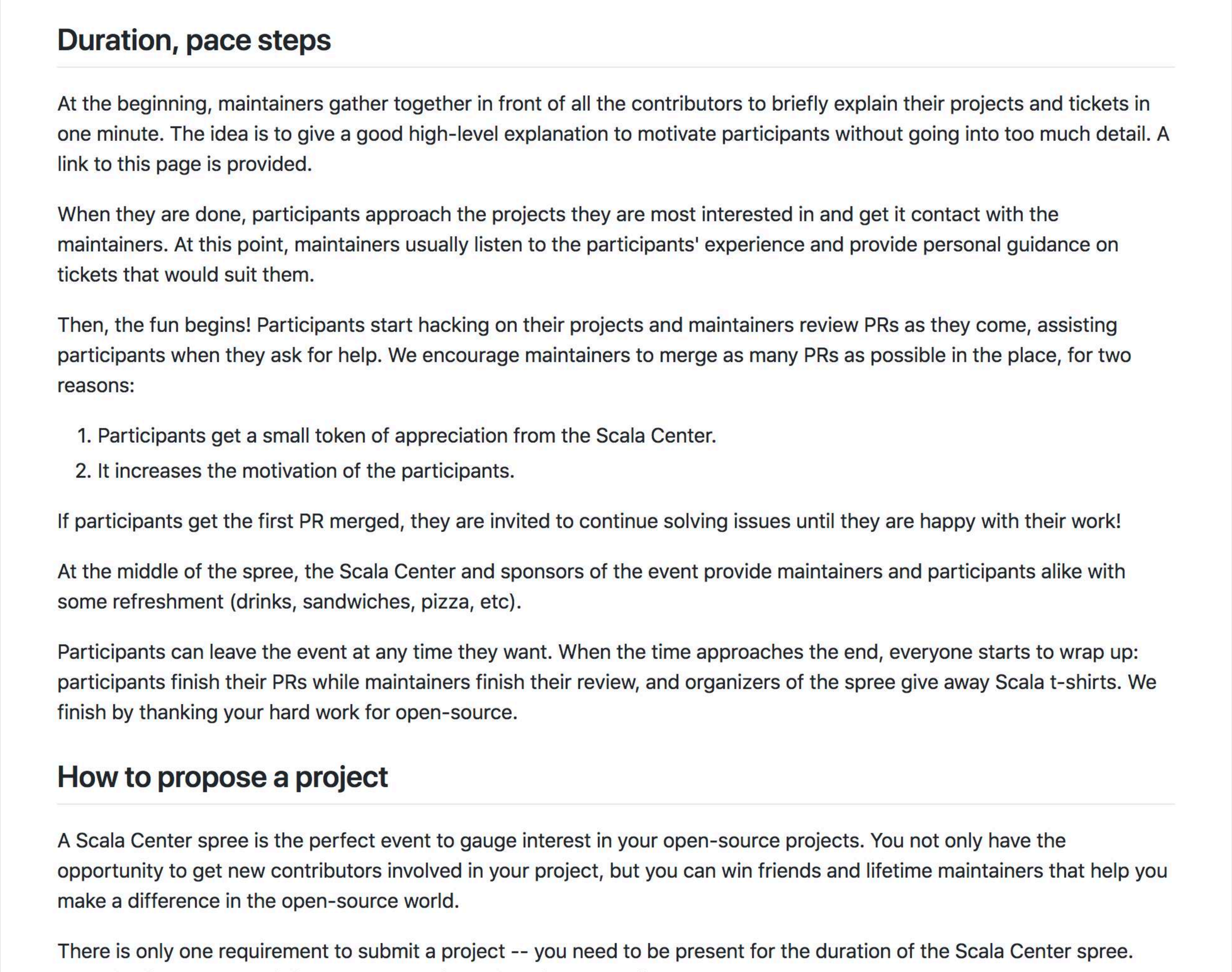
Here is a list of projects that you could contribute to during the spree:

Project	Contact
hamsters - utility library	@loicdescotte
dotty - next major version of Scala	@smarter
scalafix - code rewriting tool	@gabro
freestyle - free monad utility	@raulraja
scalaz - general functional programming library	@jdegoes
magnolia, contextual, totalitarian and “others” - nicely named libraries	@propensive
cbt - build tool	@cvogt
scaladex - index of Scala projects	@MasseGuillaume
scastie - web-based Scala playground	@MasseGuillaume
scalajs-bundler - NPM modules bundler for Scala.js	@julienrf
doodle - composable vector graphics	@julienrf
endpoints - remote communication	@julienrf
add your project here!	

Who will be leading the Scala Open Source Spree?

- Julien Richard-Foy, @julienrf (Scala Center)
- Guillaume Massé, @MasseGuillaume (Scala Center)

Want to add your project to the list? Jump to the next section!



Duration, pace steps

At the beginning, maintainers gather together in front of all the contributors to briefly explain their projects and tickets in one minute. The idea is to give a good high-level explanation to motivate participants without going into too much detail. A link to this page is provided.

When they are done, participants approach the projects they are most interested in and get it contact with the maintainers. At this point, maintainers usually listen to the participants' experience and provide personal guidance on tickets that would suit them.

Then, the fun begins! Participants start hacking on their projects and maintainers review PRs as they come, assisting participants when they ask for help. We encourage maintainers to merge as many PRs as possible in the place, for two reasons:

1. Participants get a small token of appreciation from the Scala Center.
2. It increases the motivation of the participants.

If participants get the first PR merged, they are invited to continue solving issues until they are happy with their work!

At the middle of the spree, the Scala Center and sponsors of the event provide maintainers and participants alike with some refreshment (drinks, sandwiches, pizza, etc).

Participants can leave the event at any time they want. When the time approaches the end, everyone starts to wrap up: participants finish their PRs while maintainers finish their review, and organizers of the spree give away Scala t-shirts. We finish by thanking your hard work for open-source.

How to propose a project

A Scala Center spree is the perfect event to gauge interest in your open-source projects. You not only have the opportunity to get new contributors involved in your project, but you can win friends and lifetime maintainers that help you make a difference in the open-source world.

There is only one requirement to submit a project -- you need to be present for the duration of the Scala Center spree.

Teaching Open Source

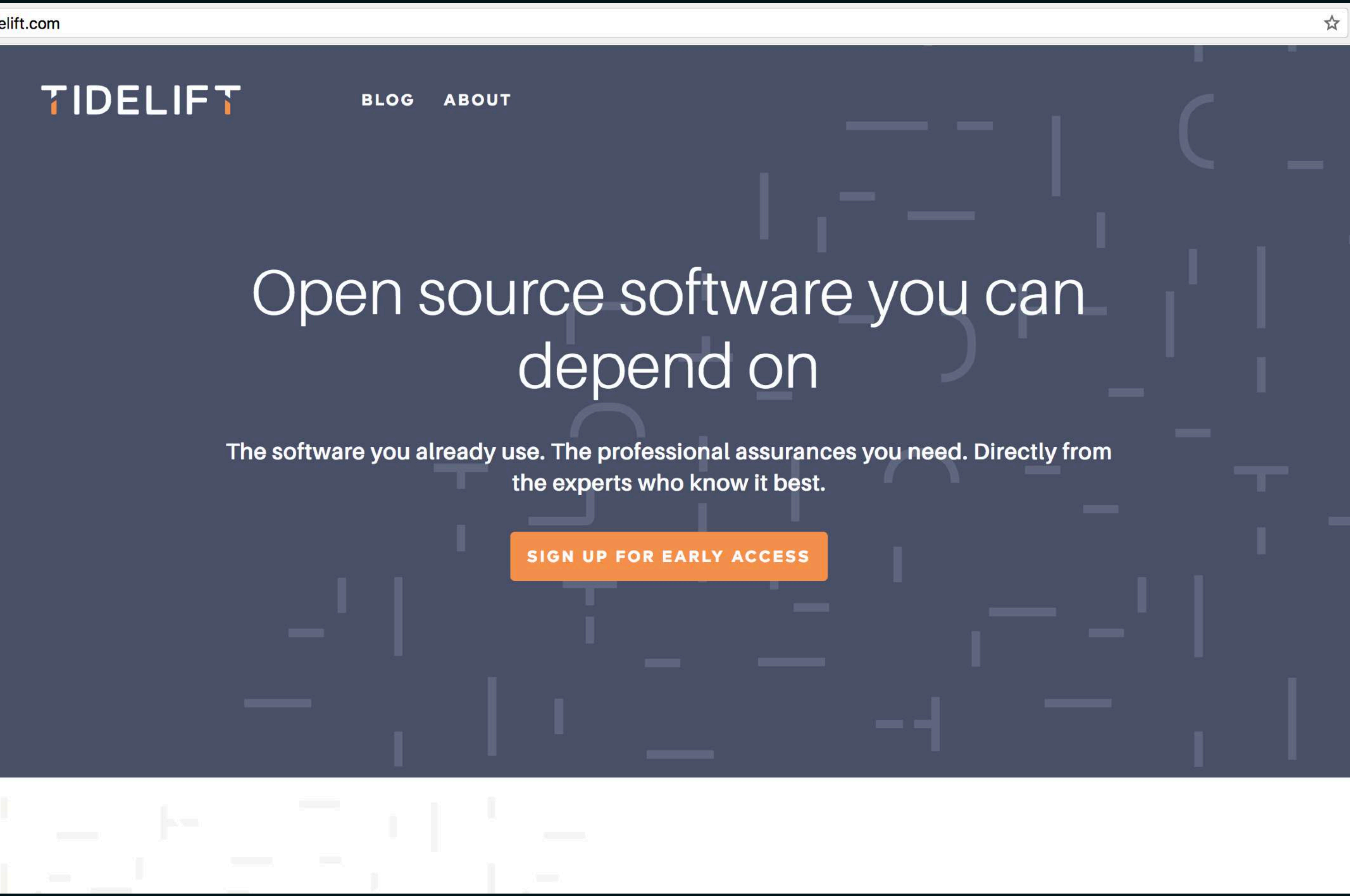
“Open Source Sprees”

Anecdotal:
It helped in the Scala community!

Many of our projects have seen a big increase in casual contributors over the past year that we've been running open source sprees!

Any FLOSS ecosystem can do it.

Worth being aware of:



Open source software you can depend on

The software you already use. The professional assurances you need. Directly from the experts who know it best.

Worth being aware of:

TIDELIFT

A new company that plans to facilitate exchange of funds relative to FLOSS project usage.

New model for funding FLOSS!

Stay tuned! News to come very soon :)

Phew.

I’ll leave you with that.

If you remember anything from this, remember this quote by Nadia Eghbal:

> “In the last five years, open source infrastructure has become an essential layer of our social fabric. But much like startups or technology itself, what worked for the first 30 years of open source’s history won’t work moving forward. In order to maintain our pace of progress, we need to invest back into the tools that help us build bigger and better things.”

Thank you
Questions? & a shameless plug

Thank you
Questions?

CURRY ON Amsterdam!

JULY 16–17TH, 2018

Two days, three tracks
with four exciting keynotes
and many invited talks

Alan Kay
Viewpoints Research Institute

Simon Peyton Jones
Microsoft Research

Ben Titzer & Andreas Rossberg
Google & Dfinity Foundation

Steve Francia
Docker