So You Want to be an Open Source Hero?

Turn your app or library into a thriving open source project.

CTO

@aembler

Andrew Embler

PortlandLabs

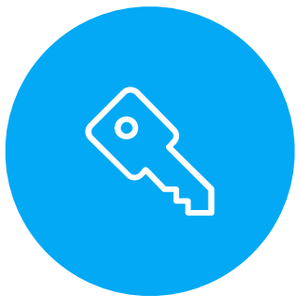
- CTO and co-founder of PortlandLabs
- Web Development Pro since 1998
- Native Oregonian
- Core Team Leader and Top Contributor for concrete5

Why Embrace Open Source?

Benefits to Users/Customers

Typically Free

Malleable for Developers



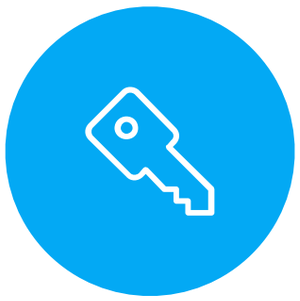
More Eyes, More Secure

Long-Term Viability

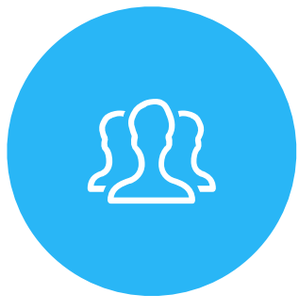
Benefits to Developers

More Reporters

More Doers



More Eyes, More Secure



Wide Audience

Convinced?

Unfortunately, Open Source is less this…

And more this:

GET
Noticed

I

Stand out from the crowd

EVOLVE
Your Software

II

Improve and release rapidly based on feedback.

EMBRACE
Your Community

III

Recognize the strong leaders amongst your users

CEDE
Some Control

IV

Give up to get more.

GET NOTICED

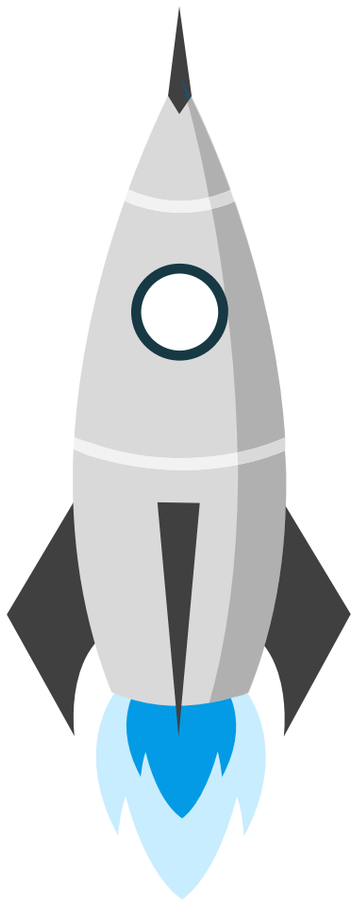
Go from one user to one thousand.

Get Noticed: The Good Ways

Use Project Hosting
Put yourself out there

Documentation
Even just a README

Package Managers
Be in them



Videos
Make them

Forums & Social Media
Introduce Yourself

Blogs & Articles
Spread the word

Use Open Source Project Hosting

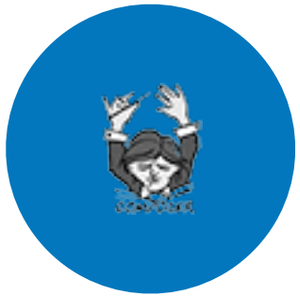
GitHub

GitLab

Phabricator

Bitbucket

Register with Package Managers



Composer

npm

Bower

Homebrew

Write some Documentation

1. #1 Metric for whether to choose a library.
2. More isn’t always better.
3. Lean toward practical examples.
4. Keep it up to date!
5. Something is **always** better than nothing.

Make some Videos

Write Blogs & Articles

- Different than docs.
- Give people a reason to choose your software.
- Promote yourself as a modern alternative.
- Long tail, lasts forever.

Embrace Social Media & Forums

- Promote your articles, not your software.
- You don’t need to own every social network. Establish a presence on Reddit, StackOverflow.
- Start a public Slack.
- Respond to every forum post. You won’t be able to forever.

Get Noticed: The Not-So-Good Ways

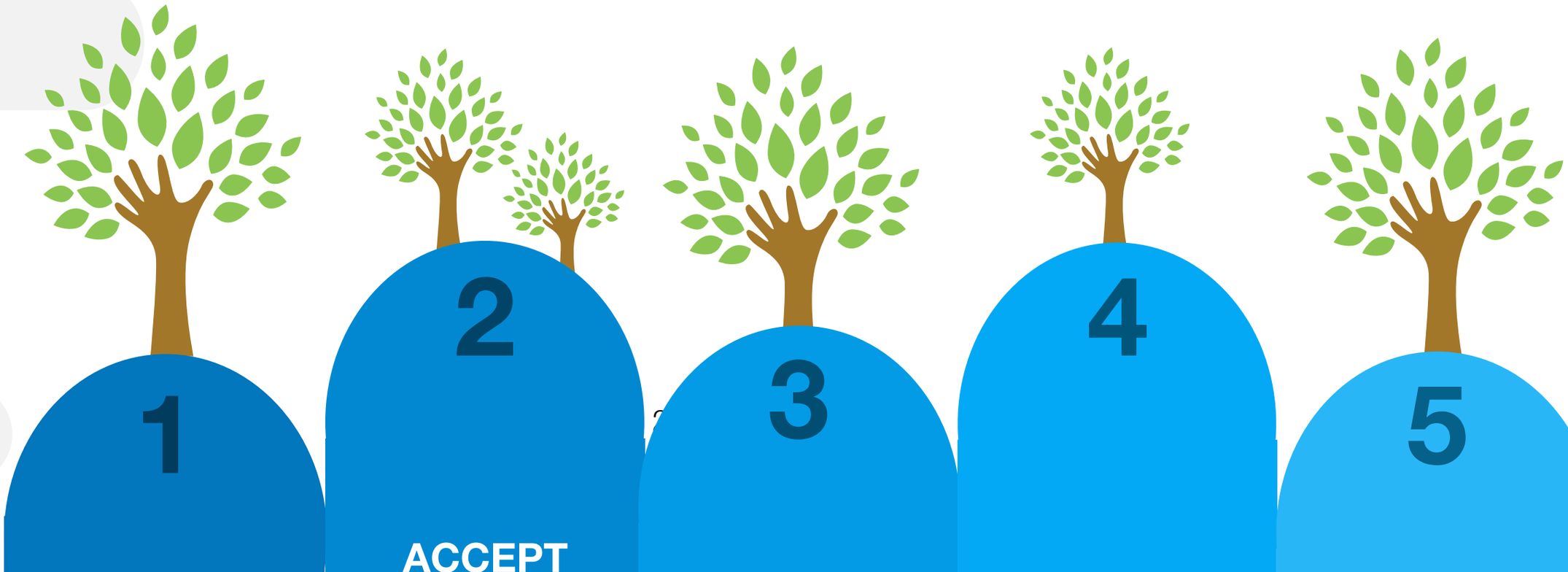
Talking Trash

Lavish Spending

Concept Videos

EVOLVE YOUR SOFTWARE

Keep the pace up.



ROADMAP
Make a plan for your software.

ACCEPT HELP
Get code submissions from your community.

MAINTAIN QUALITY
Don’t sacrifice for the sake of speed.

DOCUMENT
Make sure your docs keep pace with your changes.

SHIP CODE.
Release early. Release often.

Make a Roadmap

- Listen to your early adopters.
- “Eat your own dog food.”
- Maintain your vision.
- Use an issue tracker to make this vision clear.

Accept Help

- Pull requests are your friend!
- Label issues to help your community know what to work on.
- Create a Contributing.md file.
- Give your community permission to help.

Maintain Code Quality

- High speed = more mistakes.
- More developers = more mistakes, more inconsistency.
- Embrace Unit Testing
- Use Continuous Integration.

Don’t forget about your Docs!

- As your software improves, document your new features.
- Try to keep screenshots up to date.
- Time taken now = 10x time saved later.
- Embrace a platform for user contributed documentation.

Release Early. Release Often.

- Your software is new. Take advantage.
- Every release will drive traffic to your software.
- Every release validates the work of your early adopters.
- Use the release tools built into your software hosting platform.

LATHER. RINSE. REPEAT.

Do this over and over and over.

EMBRACE YOUR COMMUNITY

Recognize your leaders.

COMMUNITY

Let’s meet your new best friends.

THE GRATEFUL

Loves your software and lets you know.

THE SPOUSE

With you through thick and thin.

THE WORK HORSE

Helps with everything, with a minimum of fuss.

THE STRANGER

One fix, then… **poof!**

THE HERO

Does so much for you.

The Work Horse

- Hopefully the most common of your community heroes.
- Submits detailed bug reports, fixes and new code.
- Exhibit a strong understanding of how your software works, and also how you think about solving problems.
- Helps you test, stays in contact.

The Grateful

- Uses your software and **lets you know!**
- The unsolicited thank-you’s keep you going.

- Comes in unannounced and delivers a huge fix.
- Oftentimes this accompanies a detailed bug report.
- Ghosts you – in and out without a sound.

- Long-time user.
- Never tires of telling you what they think.
- Unlike most marriages, you can have more than one.
- You will probably hate them on occasion.
- They will do more for you than almost anyone else.

- Limited to a handful of people.
- Devote significant amounts of their life to working on your software.
- Evolves to become as much of an expert as you.

IT’S SUPPOSED TO BE HARD. IF IT WASN’T HARD, EVERYONE WOULD DO IT.

- TOM HANKS, *A LEAGUE OF THEIR OWN*

THE VAMPIRE

Requires all your attention.

THE PESSIMIST

Finds the downside in every decision.

THE GROUCH

Never in a good mood.

THE TROLL

I smell the blood of an open source project.

The Vampire

- A sucker of time and attention.
- Not necessarily overtly negative, just draining.
- Can derail otherwise productive threads and conversations.

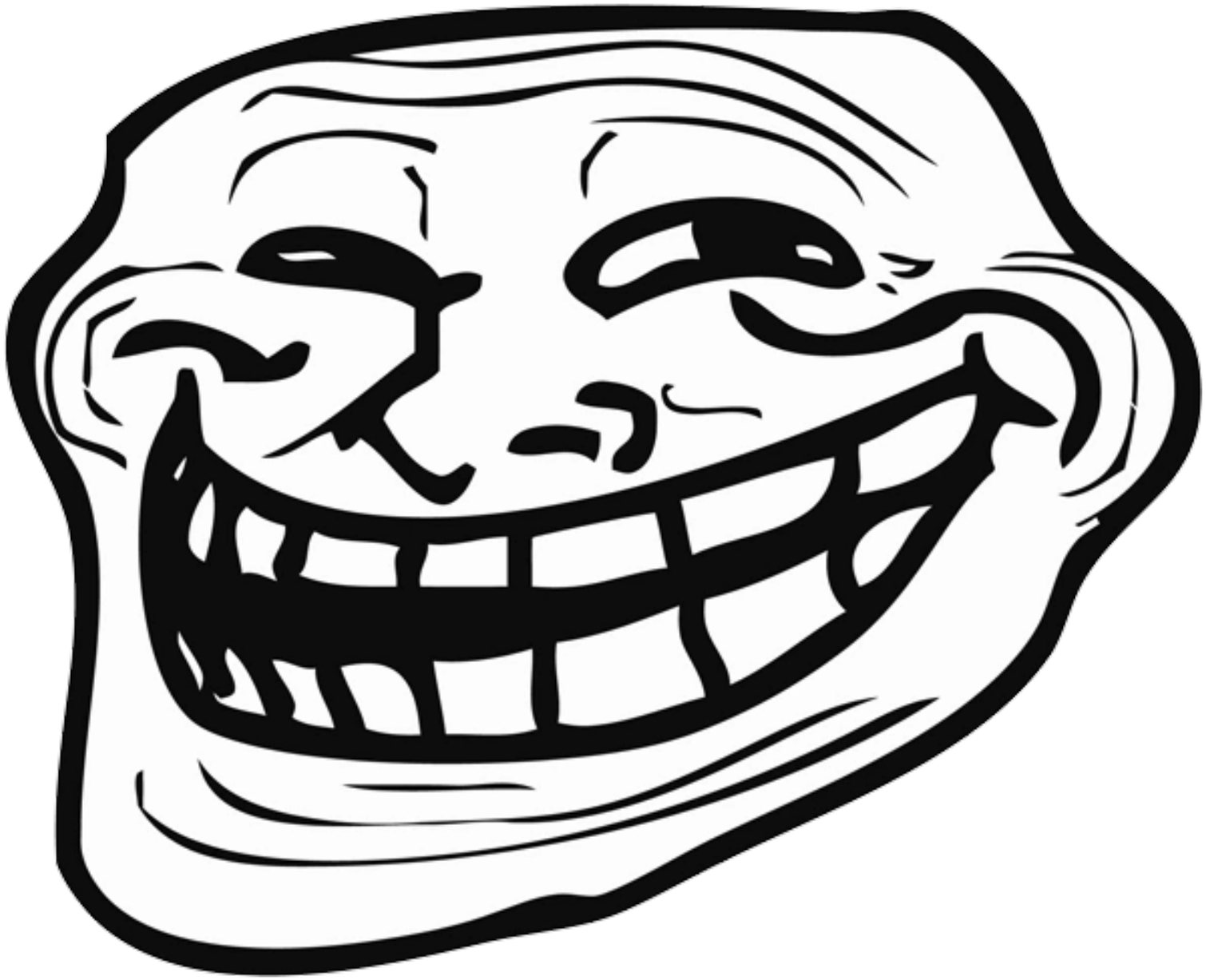
The Pessimist

- Invades threads with negativity.
- Second-guesses most decisions.
- Is frequently difficult to please.

The Grouch

- Every thread a negative one.
- Frequently derails unrelated threads.
- Is equally negative about the product and the decisions (and decision makers) behind it.
- Evolves over time.

The Troll



The Troll

- Don’t feed the…yeah yeah, we get it.
- Block and move on.
- Probably perverted logic behind it, but not worth the effort to deduce.

Handling the Negatives

- Don’t get hung up on the negative.
- Always try to clearly communicate your decisions.
- Don’t let negative threads fester. Lock them if you have to (but always with a reason.)
- Both positive and negative roles can shift over time.
- Consider a code of conduct – and enforce it.
- Don’t let bad apples spoil the bunch.

General Feedback & Communication Tips

- Respond in discovery mode, not defensive mode.
- Get your story straight.
- Stay connected and don’t drop out of sight.
- Focus first on positive aspects, then the negatives.
- You’re dealing with a real, living, breathing human – try and remember that.

Surprise! You’re a politician now!

CEDE SOME CONTROL

Give up some to gain much more.

Ceding some Control: Why?

- Not enough hours in the day.
- Everything takes longer now.
- Many more systems to worry about
- Making changes to a larger project takes more time.
- You still must release code.

Ceding some Control: What?

- Look for easy wins: places where you have neither expertise **nor** desire for ownership.
- Find places not impacted by community turnover and slower process.
- Look for perfect fits within your community.
- Be honest with yourself about what you want to do.

Ceding some Control: Who?

- Look at your community leaders.
- Heroes? Obviously. Workhorses? Absolutely? But don’t forget the others.
- Continually evaluate for new talent.

Ceding some Control: How?

- This is the toughest question.
- Use existing boring tools first (Email, Slack, GitHub/Jira/Whatever)
- Build if you must.
- Informal committees at the start.
- Look for inspiration elsewhere.

When?!

Yesterday!

(Or as soon as possible)

THANK YOU

QUESTIONS?

@aembler