

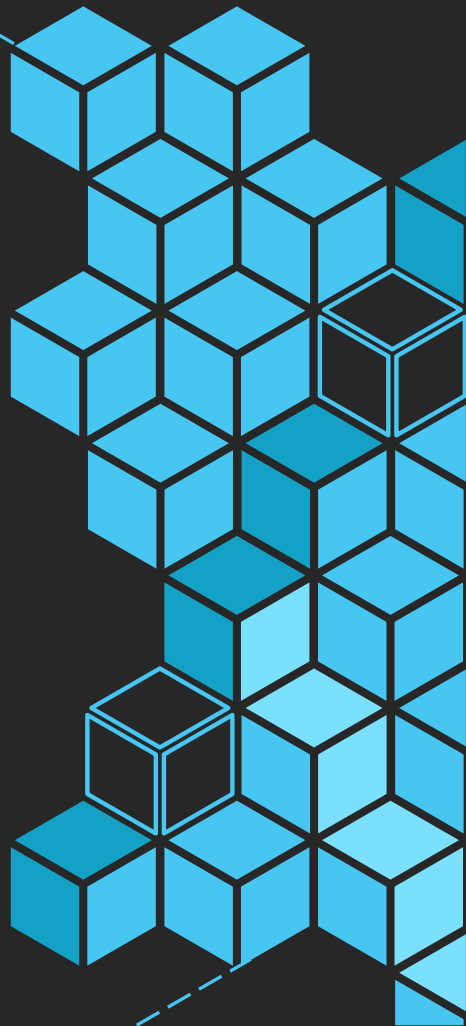


# Please Do Not Feed the Maintainers

(How to Run Away and Join the Free Software  
Circus!)

Michael Place







Licensed under Apache 2





# Would you like Salt with That?

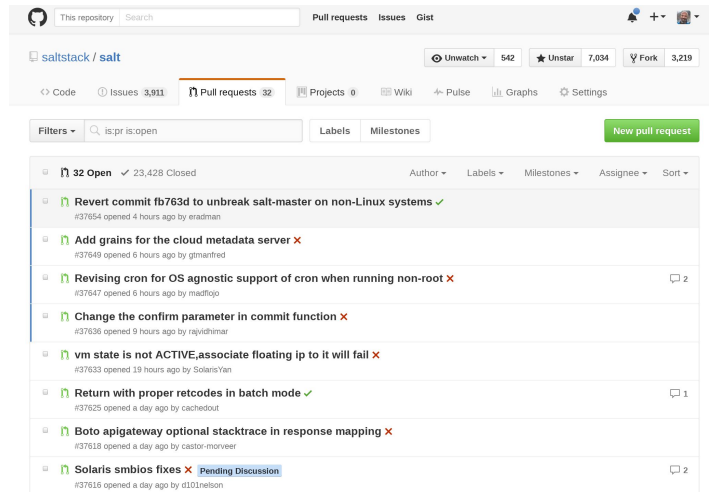
## Stats from a big Python project

-  <https://github.com/saltstack/salt>
-  Started in 2011
-  Currently around 75,000 commits
-  4,000 open Github issues
-  100-150 pull requests a week
-  50-100K LoC change every seven days
-  Support for over a dozen platforms
-  Python 2.6 (yes, really) and 2.7. Support for 3.4 is coming quickly.
-  Around 1,700 developers and 7,000 GH stars
-  The Salt OSS ecosystem makes up about 750,000 LoC
-  On a typical day, our test infrastructure executes about 6 million automated tests.



# Day in the Life of a Project Maintainer


Ringmaster in the circus that never sleeps



**Not common**

KEEP OUT





The first step to  
contributing to free  
software is using free  
software.



# GUIDE FOR NEW PROJECT CONTRIBUTORS

This guide provides tips and suggestions for new *contributors* to Apache projects. A contributor is anyone who wants contribute (code, documentation, tests, ideas, anything!) to any project hosted here at the Apache Software Foundation (ASF). Note: if you are interested in contributing **financially**, please see the [Sponsorship and Donations](#) page.

There is a separate [New Committer Guide](#) and [FAQ](#).





## CONTENTS

- [Guide For New Project Contributors](#)
- [Contents](#)
- [Community Development Is Here To Help!](#)
- [Everything Happens On Mailing Lists](#)
- [How Open Source Works At Apache](#)
- [How to get source code from revision control \(SVN or git\)](#)
  - [Nightly code / Development code](#)
  - [Getting source code on Windows: Use cygwin](#)
  - [Getting source code on Windows: Use the command-line tools](#)
  - [Getting source code on Windows: Use TortoiseSVN](#)
  - [Getting source code on Linux: Use the command-line tools](#)
  - [Getting source code with IDE/Editors: Using an IDE for source repository access](#)
  - [How to update your checked-out module](#)
- [How to Provide feedback to Apache projects](#)
- [How to Send in a Bug Report](#)
- [How to Send in Patches](#)



# How to Join the Circus

OK, I'm a user. What now?

-  Talk about it. Most projects like feedback. Give them some.
-  Tell your story.
-  Don't be afraid to file a bug.
-  If you want to contribute and you don't know how you can, ask



# On virtue

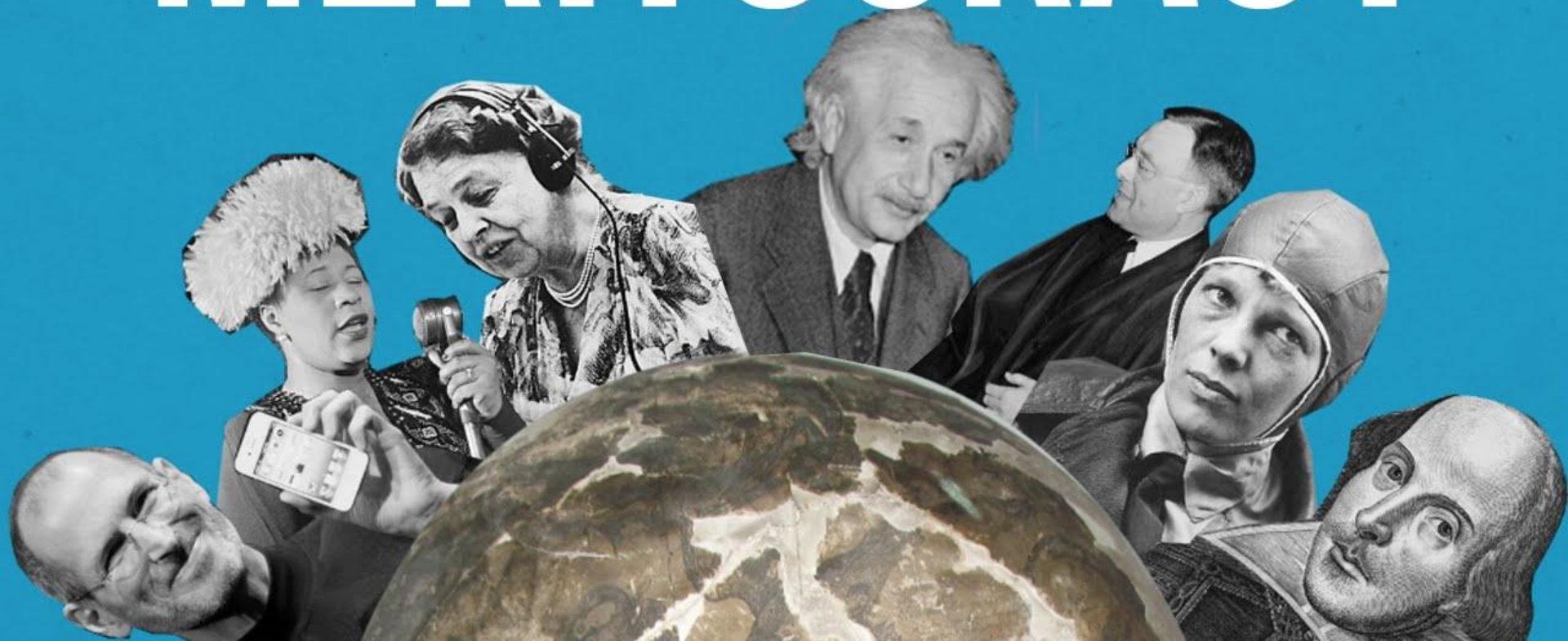
*What can everyone do? Praise and blame. This is human virtue, this is human madness.*

**-Friedrich Nietzsche**

(On virtue and maybe also on git)



# MERITOCRACY





*"How to behave when you don't  
know how to behave."*

Trustworthy

Loyal

Courteous

Thrifty

Clean

Helpful

Friendly

Kind

Brave

Respectful

---

*I praise loudly. I blame softly.*

*-Catherine the Great*

[Creative Commons Attributions License;  
Wikipedia]



# Best practices for open-source

*A brief, ambitious and opinionated guide to making your mark in the Python world.*

*Building art is a synthesis of life in materialised form.  
We should try to bring in under the same hat not a  
splintered way of thinking, but all in harmony together.*

**-Alvar Aalto**



# How to file a bug so that it gets fixed

## One chance to make a first impression

- ⬡ Knowing how to file a bug is the **best** skill for an open-source developer to have.
- ⬡ Bug reports are the most common interaction that people have with OSS developers
- ⬡ Your reputation in the community that you're in will have a lot to do with how you handle bug reports
- ⬡ Bug reports aren't just about bugs. Sometimes they can be about documentation, features that you wish existed, or questions about intended behavior.





# What makes up a bad bug report?

There's so, so much.

*The time and effort expended by a developer to acquire the information she or he needs to begin troubleshooting to is inversely proportional to the quality of the bug report!*

$$q=1/t$$

# Search first

- Always search the issue tracker for something similar.
- Consider expanding the search to other resources, like StackOverflow or IRC
- Ask more experienced co-workers if you can, or a local mailing list.
- Remember that the maintainer will have to do this even if you don't, so be a good citizen and do this before you file a bug.
- You might even find that it's solved!

“Google can bring you back 100,000 answers. A librarian can bring you back the right one.”

— Neil Gaiman

# | Is this *really* a bug?

- It's easy to forget that sometimes what feels like a bug really is a documentation error or something that's just unclear. Be open to that possibility.

***Docs > Tests > Code***

*A man can fail many times, but he isn't a failure until he begins to blame somebody else.*

*-John Burroughs*

# Building a test case

- The goal is to get a developer to be able to see and *reproduce* with a minimal amount of code.
- A good test case should contain as much natural language as code, if not more.
- Bugs in the develop branch are often easier to fix but bugs in release branches are often more critical.
- Tests should be done against both the branch where you see the bug *as well* as the branch where active development is happening.

*It is not the beauty of a building you should look at; it is the construction of the foundation that will stand the test of time.*

*- David Allan Coe*



# Packaging test cases

- Consider the use of Docker as a mechanism for packaging test cases.
- If there are lots of dependencies necessary, consider packaging them in a virtualenv.
- Automated test cases, if possible, are best

*But I think my mistakes became the chemistry for my miracles. I think that my tests became my testimonies.*

*-T. D. Jakes*

# Does this bug matter?

- How does this affect you specifically?
- How might it affect others?
- The best kind of bug report reads like a use case in a design document.

*At a certain point in their existential experience, the oppressed feel an irresistible attraction toward the oppressor and his way of life. Sharing this way of life becomes an overpowering aspiration.*


**- Paulo Freire**

# C'est bon?

- Know the language and cultures that you're dealing with. It matters more than you realize.
- Feel free to ask for help on the project mailing list if you need help with translation. There are also translation-specific groups out there.
- Respect time zones.
- Respect personal and professional boundaries.

*A warm smile is the universal language of kindness.*

*- William Arthur Ward*



Whatever you do,  
don't be a stranger.

*"I may not have been sure about what really did interest me, but I was absolutely sure about what didn't."*

— *Albert Camus, The Stranger*



# Contributing

To merely observe your culture without contributing to it seems very close to existing as a ghost.

***-Chuck Palahniuk***

# Contributing checklist

- CONTRIBUTING.md
- Does the bug have an issue number
- Have you read the style guide?
- Is there a contributor agreement?

*I woke up an hour before I was supposed to, and started going over the mental checklist: where do I go from here, what do I do? I don't remember eating anything at all, just going through the physical, getting into the suit. We practiced that so much, it was all rote.*

*-Alan Shepard*

# Good contributions

- Adhere to the style guide
- Have been tested against all applicable versions.
- Only contain changes relevant to the issue in question.
- Don't try to show off
- Do not mix and match functional and style changes.
- Have already been run through the test suite.
- Are up-front about possible downsides.
- Are readable

*Good sense is both the first principal and the parent source of good writing.*

*-Horace*

# Avoid these things!

- Adding yourself to the CONTRIBUTORS.md file before you've really contributed.
- Merge conflicts
- Modifying code without notifying the appropriate people.
- Submitting broken tests in PRs.
- WIP pull-requests without a good reason.
- Using vulgar or inappropriate language in any form, even in jest.

*A resolution to avoid an evil is seldom framed till the evil is so far advanced as to make avoidance impossible.*

*-Thomas Hardy*



# Agreeing to disagree

- Recognize that the maintainers might have considerations you aren't aware of.
- Keep conversations on-topic
- Go out of your way to be polite. Be *irrationally* polite.

*Honest disagreement is often a good sign of progress.*

**- Mahatma Gandhi**

# Good Python

- Is a good citizen to its neighbors
- Uses configuration values to gate changes when appropriate
- Doesn't change around keyword ordering
- Is visually pleasing
- Is free of “weasel words”
- Is modular
- Is modular
- Is free of “weasel words”

*A good person can make another person good; it means that goodness will elicit goodness in the society; other persons will also be good.*

**-Bhumibol Adulyadej**



# Setting out on your own

So you want to make a project?

*You have brains in your head. You have feet in your shoes. You can steer yourself in any direction you choose.*

*You're on your own, and you know what you know. And you are the [one] who'll decide where to go.*

*-Dr. Seuss*




SALTSTACK



# Kintsugi

Not only is there no attempt to hide the damage, but the repair is literally illuminated... a kind of physical expression of the spirit of mushin....Mushin is often literally translated as “no mind,” but carries connotations of fully existing within the moment, of non-attachment, of equanimity amid changing conditions. ...The vicissitudes of existence over time, to which all humans are susceptible, could not be clearer than in the breaks, the knocks, and the shattering to which ceramic ware too is subject. This poignancy or aesthetic of existence has been known in Japan as *mono no aware*, a compassionate sensitivity, or perhaps identification with, [things] outside oneself.


— Christy Bartlett, *Flickwerk: The Aesthetics of Mended Japanese Ceramic*”



2.8M+ people created  
their *first* GitHub repo  
last year.

*I'd never just want to do what everybody else did.  
I'd be contributing to the sameness of everything.*


*-Captain Beefheart*



# What is the problem to which this is a solution?

*Happiness is having a scratch for every  
itch.*


*-Ogden Nash*



# When should I share my work?

*Program construction consists of a sequence  
of refinement steps.*

**-Niklaus Wirth**  
*[Inventor of Pascal]*



A project is ready to be  
shared as soon as it has the  
*simplest* possible answer to  
the problem.

*Problem solving is hunting. It is savage  
pleasure and we are born to it.*

*-Thomas Harris*





What should a new  
project contain?

# README.md

- Discuss the problem you are trying to solve.
- Explain how your project proposes to solve it.
- Caveats
- How to get started
- What to do if you get stuck
- How to contribute

*Either write something worth reading or do something worth writing.*

*-Benjamin Franklin*

# AUTHORS.md

- Should contain your name and email address.
- A short note at the top explaining how to add your name.
- You can add people to it as they contribute if they don't want to do it themselves. Be sure to ask first!

*Great authors are admirable in this respect: in every generation they make for disagreement. Through them we become aware of our differences.*

*-Andre Gide*



# CONDUCT.md

- Many to choose from
- We use the Contributor Covenant.
- Easy to select one ahead of time and just re-use for every project.

*Remember upon the conduct of each  
depends the fate of all.*

*-Alexander the Great*

# Packaging

- Follow the best practices at **packaging.python.org**.
- You may need a **MANIFEST.in** if you wish to package files which fall outside the scope of what **setup.py** will do for you.
- At a minimum, create a **setup.py** that has enough information in it that you (or a contributor) can upload your package to PyPi.

*Information about the package is as important as the package itself.*

**-Frederick W. Smith**  
FedEx CEO

# | Packaging (cont)

- Consider the use of **twine** to make all of this easier on yourself.  
[<https://github.com/pypa/twine>]
- Create a **requirements/** directory as soon as your project has dependencies.

# INSTALLATION.md

- Be sure to mention again how they can get help if they get stuck.
- This can be as easy as pointing somebody to **pip install my\_great\_project**.
- If the project requires installation steps, document them in this file.
- If there are certain platforms which may not work, document those here.
- Be certain to mention supported versions of Python.

# LICENSE.md

- Select licenses wisely. It's not always easy (and sometimes impossible) to switch licenses once you've selected one.
- Some businesses are resistant to certain licenses. If this is a concern to you, do research.
- Simply not licensing your work doesn't mean that you'll be free of licensing concerns.

*Sharing is good, and with digital technology,  
sharing is easy.*

*- Richard Stallman*



# | Promoting your project

*Kintsugi in practice*

*Word of mouth is the most valuable form of marketing, but you can't buy it. You can only deliver it. And you have to really deliver.*

*-G-Eazy*

# Promotional ideas

- Volunteer to talk about it at user group meetings
- Announce it briefly on a mailing list
- Help out in places where your solution might be helpful.
- If there's no community around your idea, make one!

*I don't need anybody to market or promote me. If people don't want to hear this music, then it's not for them. You cannot please everybody.*

*- Lauryn Hill*

[Pull requests](#) [Issues](#) [Gist](#) [ToDo](#)

**Pedro Algarvio**  
s0undt3ch

Follow

Block or report user

@saltstack

Portugal, Lisboa

<http://blog.algarvio.me/>

Joined on Jun 9, 2010

Organizations

Overview

Repositories 109

Stars 38

Followers 58

Following 12

### Pinned repositories

[saltstack/salt](#)

Software to automate the management and configuration of any infrastructure or application at scale. Get access to the Salt Open software package repository here:

Python 7.1k 3.2k

[pytest-helpers-namespace](#)

Forked from saltstack/pytest-helpers-namespace

PyTest Helpers Namespace

Python

### 5,217 contributions in the last year



### Contribution activity

Jump to ▾

2016

November 2016

*When you do good work, people will notice you.*

**- Ram Kapoor**

African users on  
GitHub increased  
from 4,000 to  
12,000 in just *two*  
*years*.

Data via **blog.ona.io** and **Yeleman** in Bamako, Mali.

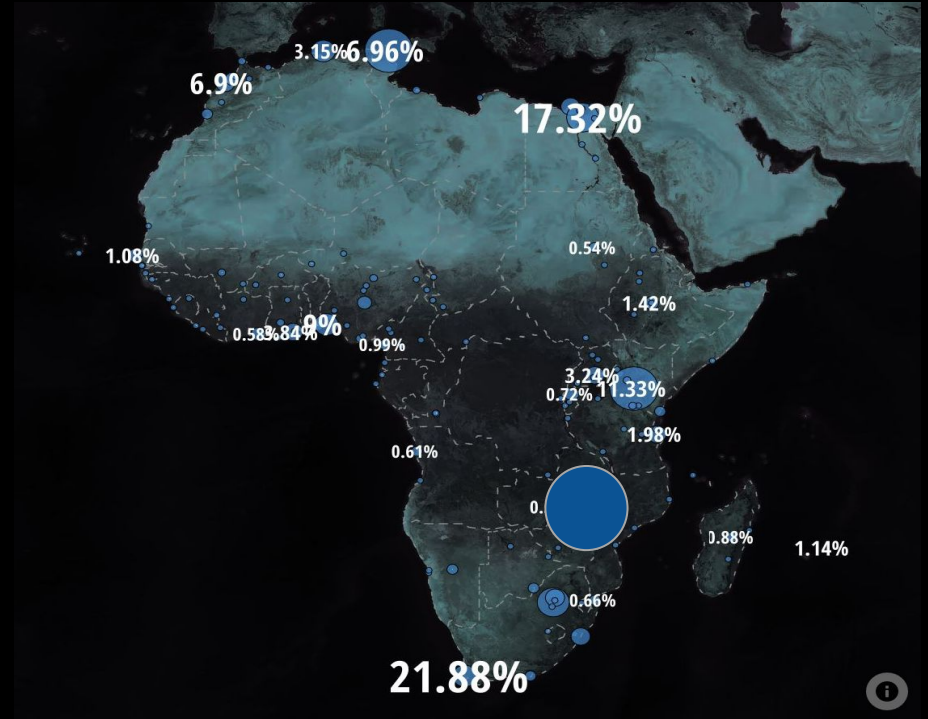


Image vi [blog.ona.io](https://blog.ona.io)



# <https://github.com/cachedout>

All slides are available in various formats, along with examples of the concepts we've discussed today.

*Absence of proof is not proof of absence.*

*-William Cowper*



Thank you!