

Mentoring: Your Path To Immortality

Rich Bowen, (Apache|Red Hat)

Slides: BoxOfClue.com/mentoring





Echecrates

What did we learn?

- Two paths to immortality:

1) Leave an extensive body of work

2) Leave a student who is even more impressive than yourself

Option 1: Leave extensive work



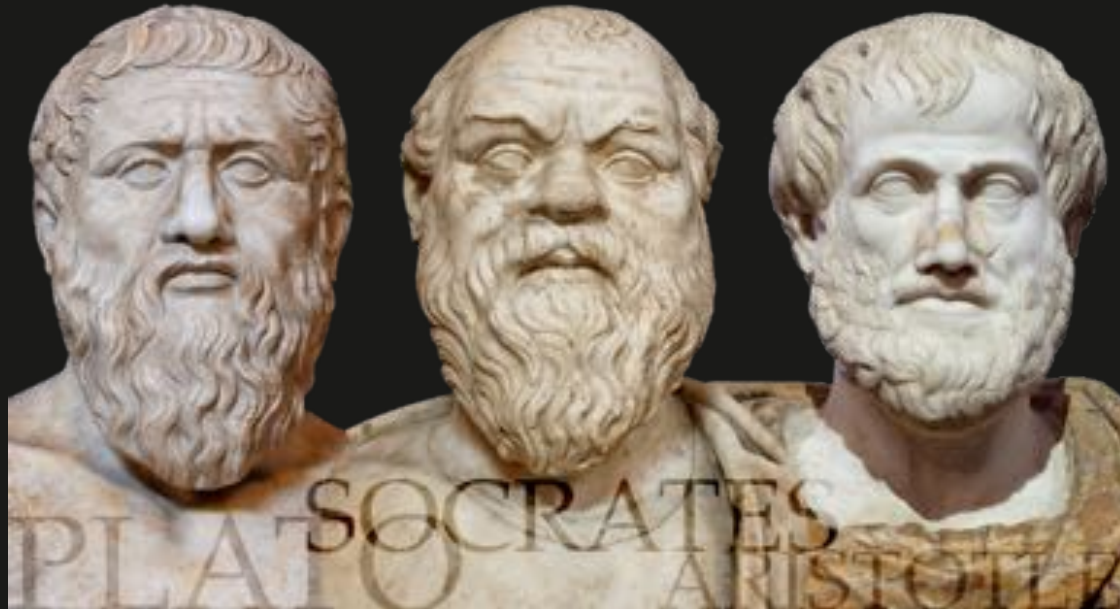
That's hard

- You have to actually **DO** that work
- And you have to pick a medium that is permanent
- Alas, we work in software, which is definitively *not* permanent

Option 2: Leave impressive students



Option 2: Leave impressive students



A word of caution ...



So ... how do you do that?

- Practical tips for active mentoring in Open Source

Give permission

- Social
- Technical

Technical permission

- "Commit bit"
- File permissions, or root, on your servers



Why do we not hand out commit rights?



Image: Creative

They're not trusted yet

- How will they become trusted if we don't give them space to make mistakes?
- Withholding trust is a great way to ensure that they don't stick around to earn trust.

They might break something



Photo Credit: [Dieter](#)

[Drescher Flickr cc](#)

But, but ... they might break something!

Some things are much more expensive when they get broken

- If it's infrastructure, let them work on dev/test first
- If it's code, let them work in a branch first

RTC vs CTR

- Review, then commit
- Commit, then review

Sometimes it makes sense to have different policies for your release branch vs your development branch, for example.

CI and commit hooks

- Every change goes through testing, so that rejected commits come with a detailed report of *why* it's rejected
- Ensure that someone is watching, and can help someone turn failure into success.



Photo Credit: [Mariana Montes de Oca](#) Flickr [cc](#)

Social permission

- This is a lot harder
- Give people *social* permission to participate
- There's many ways that we forbid people to participate

Give time to participate

When you see a bug, don't fix it immediately unless it's actually urgent

- Log a ticket
- Include a description of how you might fix it
- Talk about the ticket on the mailing list
- Ask if anyone has time to look at it
- Wait a bit

Corporate participation

- Waiting for new contributors to participate can be particularly difficult in projects with a large corporate presence
- This is hard, and is becoming more and more common in open source as a whole
- I don't know how to fix this. Suggestions welcome.



Ask people to do things

Directly asking someone to work on something serves a number of purposes

It lets them know that they're allowed

You might think they already know this. It's open source, after all.

They don't know.

They think that they have to be *someone special* to contribute to your project.

You need to tell them that they are already someone special.

It lets them know that you think they can do it

Telling someone that you believe they can do it is a very powerful thing.

It helps them believe in themselves.

It lets them know that they are trusted enough

Telling someone you trust them, and then actually trusting them, is very, very empowering.

It lets them know that the project is counting on them

Ask them to do it, and then expect that they're going to.

Feeling that someone depends on you is a huge encouragement to move out of your comfort zone.

**It models the behavior that they
should then follow for others**



This is Ken. In 1999 he encourage me to submit a paper to a conference. At the time, I was pretty sure I had nothing whatever to say.



These are just a few of the events that I've spoken at since then, as a direct result of Ken asking me. Every wall of my office is covered with these.



This is Jim. In 1998 or so, he told me to go fix it myself, rather than waiting for him to do it.



I did. This resulted in thousands of commits on the httpd docs. That, in turn, resulted in several books, and most of my jobs since then.



This is Sally. In 2006 she encouraged me to get more involved in the governance of the Apache Software Foundation.



Today I'm a director at the ASF. I've served in a number of different officer positions, and currently am the VP in charge of conferences.





Image courtesy of www.askideas.com

Do What You Can't



I'm always doing things I can't do.
That's how I get to do them.

— *Pablo Picasso* —

AZ QUOTES

Related: <https://youtu.be/jG7dSXcfVqE> ("Do What You Can't", by Casey Neistat and Max Joseph)

How to ask - Be specific

- Don't say "come help"
- Do say "please do this task", and point to a ticket
- Ask someone to answer a question on the forum/IRC/ mailing list, rather than just answering it yourself

How to ask - set expectations

- Set a time expectation
 - Tell them when it needs to be done by
 - Check in midway through the time
- Set code quality expectations
 - Coding standard
 - Test procedure

How to ask - Offer to help

- Offer to review whatever they produce
 - Be very specific in your critique
 - Be kind, but don't approve bad work in order to be kind. That's not actually being kind.
- Don't interfere or do it for them. Make it clear that you're willing to help, but don't butt in.
- Don't be condescending. Assume competence, and help only when needed

Give credit

- Celebrate first contributions
 - On Twitter
 - In your newsletter
 - In the release notes

OpenStack mentions every contributor to each release of the software. <https://www.openstack.org/software/queens/> lists all 1,925 contributors by name.

Wordpress does this too:

<https://wordpress.org/news/2018/05/wordpress-4-9-6-privacy-and-maintenance-release/>

Even if you helped ...

- If you mentored a contribution through to a commit, even if you did *most of the work*, give the credit to your student.
- Remember that it's not about you

Identify good starter tasks

- Related to earlier point, but more ...
- Actively tag tickets as "easyfix" or "good-first-bug", or similar
- Forbid the regulars to work on them
- Set notifications so you know when someone is working on it

#41155 new defect (bug)

Opened 3 weeks ago

Last modified 2 weeks ago

WordPress 4.8 Admin Sidebar Sub Menu Navigation Issue

Reported by:	 codexdemon	Owned by:	
Milestone:	Awaiting Review	Priority:	normal
Severity:	trivial	Version:	
Component:	Administration	Keywords:	good-first-bug has-patch
Focuses:	ui		

Description

Hello Team

As this is really awesome as working with the community of wordpress CMS.

While reviewing themes of WordPress Me and my colleague @amolebonde face issue about accessing sub-menu navigation when we are theme detail page Wordpress Admin > Appearance > Themes > Theme Details.

Once we are on Theme Detail page then in case if the user wants to access any Sub Menu Navigation it not shows well, which goes behind the popup theme detail page.

For Ref. Screen Shot Attach



GitHub repository interface for `redhat-openstack / easyfix`. The page shows 4 open issues and 0 closed issues. The issues listed are:

- Add with_doc conditional over -doc subpackage** (Issue #6, opened a day ago by `chkumar246`, 0 of 5 votes)
- Replace rm -rf requirements.txt step with proper rpm macro** (Issue #0, opened a day ago by `chkumar246`, 0 of 6 votes)
- Move package description to global variable and replace with it.** (Issue #4, opened a day ago by `chkumar246`, 0 of 6 votes)
- [DLRN] Proofread documentation** (Issue #1, opened 14 days ago by `javierpena`, 2 votes)

ProTip! Exclude your own issues with `-author:bowen`.

© 2017 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#) [Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)



chkumar246 commented a day ago

Member + 👤 ✎

In RDO packaging, we use `rm -rf requirements.txt/test-requirements.txt` in order to avoid python package installation from pip during rpm package building process. Since from Pike release, we have to replace this step with rpm macro `%py_req_cleanup`.

How to reproduce:

1. Clone openstack-nova spec file `git clone https://github.com/rdo-packages/nova-distgit`
2. Go to nova-distgit and open `openstack-nova.spec` in editor.
3. You can find `rm -rf {test-}requirements.txt tools/{pip,test}-requires` under `%prep` section in spec file.

How to fix:

1. Replace `** rm -rf {test-}requirements.txt tools/{pip,test}-requires **` with `%py_req_cleanup`
2. Add **BuildRequires: openstack-macros** just below description section of main package.
3. Add the changes file, commit it and send a Gerrit review.

Below is the list of packages which need same fixes:

- [openstack-nova](#)
- [openstack-glance](#)
- [openstack-cinder](#)
- [openstack-swift](#)
- [openstack-neutron](#)
- [openstack-keystone](#)

Easyfix: Reference

- <https://fedoraproject.org/easyfix/>
- <http://rdoproject.org/easyfix>
- <https://core.trac.wordpress.org/tickets/good-first-bugs>

Clearly document your process

You may think that your project contribution process makes sense

It doesn't

Document the process. Then have someone go through it for the first time, and write down everything that was assumed, or otherwise not obvious. Add that to the process documentation. Repeat.

Who should you mentor?

Be carefully who you invest your life in



People who ask good questions

- Encourage them to write up the answers they get and contribute to the documentation
- Documentation is the gateway drug to project participation



People who always argue about how things are

- People who complain tend to view themselves as the outsider, and you as the cause of their problem
- Welcoming them into the "us" who can fix the problem they're having changes the game, and makes them responsible for their own fix

People who are answering questions on other sites

- These people are demonstrating a passion to help others
- Encourage them to come help the whole world



People who are developing plugins/modules/patches for your project

- Finding related projects on Github is a good way to find talent
- Invite them to come join the party
- Plugins, modules, themes, additional documentation, etc, etc



Everyone else

- Be aware that mentoring can be a significant time investment
- Sometimes, the people that you mentor can become lifelong friends
- However ...

Mentees to avoid

- Some people are just looking for something to pad their resume
- Others are trying to bypass the hard work and get straight to the fame and adulation that invariably comes with open source contributions. (Or so they imagine.)



"Do my homework" mentoring

- And some are just looking for you to do your job for them
- It can be hard to detect these folks before you've already sunk hours of your life into it.

Amplify yourself

- You have only a certain number of hours in your day, and days in your life
- Amplify your overall influence on the world by cloning yourself.



Touch the future

- If you only look out for yourself, your influence ends the day you step away from the project.
- Mentoring allows you to have an influence on the future, both of your project, and of the individuals involved.



finis

rbowen@apache.org

[@rbowen](#)

[@CentOSProject](#)

[@RDOPROject](#)

[@ApacheCommunity](#)

Slides: BoxOfClue.com