What are the best practices for releasing an open source project? [closed]

Asked 15 years, 10 months ago Modified 13 years, 7 months ago

Viewed 667 times

4

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 10 years ago.

Improve this question

We have a pretty cool little web framework that we have used successfully on dozens of client projects. We are planning to release this software to the community. However, I am wringing my hands about what should/should-not go on a new open source software project page. What are the things the site must have? Docs? A Wiki? A link to download? What else?

And, a related but possibly different question is how do we begin marking release numbers. All we use internally

is the SVN stamp. Is there a good way to determine when to start calling something version 0.9 versus 1.0 and 1.1 and so-on?

open-source release release-management

Share

Improve this question

Follow

edited May 30, 2009 at 10:06

Ludwig Weinzierl

16.6k ● 10 ● 46 ● 49

asked Feb 17, 2009 at 8:47

Sam McAfee

10.1k ● 16 ● 62 ● 65

6 Answers

Sorted by: Highest score (default)

4

You can get an idea of what's required by what open source project hosting sites provide:

- A web site which acts as the "one stop shop" for the project
- Docs, potentially in wiki form
- A source repository allowing browsing, anonymous checkout, and authenticated and authorised commits
- Issue tracking and new feature requests

As for version numbers... I don't think *anyone's* worked out the best way of doing that yet :) With a bare minimum

of thought, I'd consider:

- v1.0 should be ready for production use
- Major version number changes can completely lose backward compatibility (if necessary - hardly a goal though!)
- Minor version number changes should usually be mostly compatible - deprecating is probably better than removing/renaming bits of API
- Smaller-than-minor version number changes should only include minor functional additions (if any) and bug/performance fixes

Share Improve this answer

Follow

answered Feb 17, 2009 at 8:56

Jon Skeet

1.5m ● 889 ● 9.3k ● 9.3k

2

On versioning, I think the absolute best place to start from is Semantic Versioning.

Share Improve this answer

Follow

answered May 26, 2011 at 15:13

Alex Miller

70.1k ● 25 ● 124 ● 168

The 0.9 / 1.0 / 1.1 / 1.0.1 / ... version labelling is for marketing purpose only (in the good sense of it). This

1

allows your users/customers to identify if the release is major, minor or bug-fix and whether you consider it mature or not yet.

The minimum to deliver is sources. Other deliverables depend on how you are willing to help your users and provide them support.

Share Improve this answer

Follow

answered Feb 17, 2009 at 8:56

mouviciel

67.8k ● 13 ● 108 ● 142

1

Choose a website to host the source on first (SourceForge, for example). Get the source up there on a version control system with anonymous checkout. Get an email address on there for people to contact you.

Call this first version 0.1. This is because you don't have docs yet to support the project.

Then breathe.

Then start looking at documentation, like a wiki. Once you have it all covered, at a basic level of detail, and you believe the release is ready for some primetime, then move to 1.0, and start providing binary downloads.

Share Improve this answer

Follow

answered Feb 17, 2009 at 9:03

Neil Barnwell

42k ● 31 ● 154 ● 226

1

Make sure you think about the license for the sources.

When I look at an open source project, one of the first things I check is the license. If the license is not GPL2/GPL3/BSD styles or similar, that's a demotivator for me.

The license means what people will do with it, how it can grow, and how much it is owned by the corporate who released it. As by choosing open source I try not to depend on corporations (who depend on their share holders), I really choose to use the software that is really free.

As the open source community is very sensitive to corporate power (Google seems a bit immune to that at the moment), so you really must make sure to deliver the message of **truly free** on your web site and other materials you release about the software.

See more on free software and open source definitions of the FSF.

Share Improve this answer

Follow

answered May 30, 2009 at 10:46

yhager

1,662 ● 15 ● 16

0

Take a look at GitHub or Google Code. they provide a very good starting point for own open source projects. You can describe your project, documentate in a wiki, use

git or svn as your repository, and provide downloads together with an issue tracking and multi-developer management. Nice environments out of the box to learn from and to use them.

For release numbers: I don't recommend 0.9 or something like this for pre-releases. The reason? What about release 1.9? Is it the 9th sub-release of the major release 1 or is it the last pre-release of release 2? My release standard is decribed here: http://code.google.com/p/tideland-eas/wiki/ReleaseStandard. I'm using a three-number-scheme, major, minor, and fix, together with a status code, alpha, beta, gamma, and the release date. So I'm able to handle multiple releases in parallel easily.

Hope this helps.

mue

Share Improve this answer Follow

edited Feb 17, 2009 at 9:08

answered Feb 17, 2009 at 9:02

themue

7,840 ● 2 ● 27 ● 28