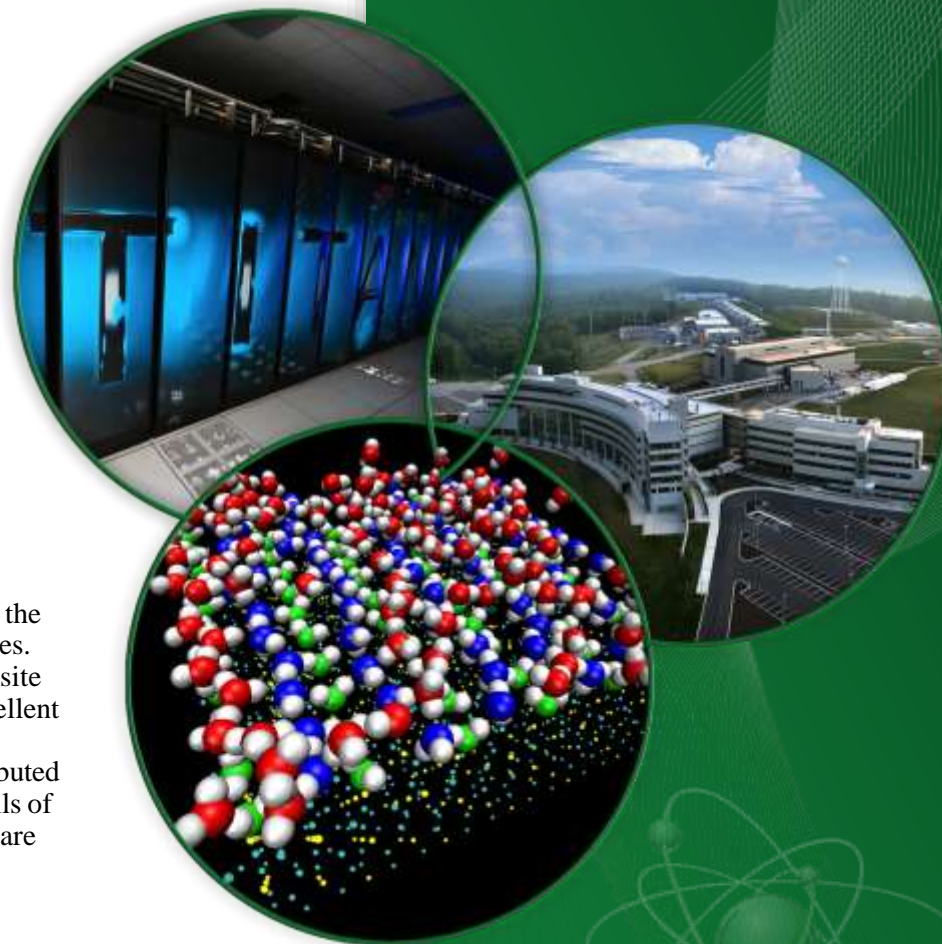


Open Source vs. Collaborative Software: FOSS is Not Enough

Peter F. Peterson

Much of the software at user facilities is developed is released by making the source code available and decorated with one of many open source licenses. Some projects go further than this and publish the source on a third party site with links to issue tracking and automated build results. While this is excellent for the user of the software, it doesn't go far enough to engage the larger community and ensure a well maintained project that can be easily contributed to and exist beyond the term of a post-doc. This talk will discuss the details of how to convert an open source software project into a collaborative software project.



What we want

- Written by a post-doc
- The source is the documentation
- It's on the cluster
- The post-doc runs it for everybody
- Post-doc gets a “real job”
- Archaeology reveals a VMS binary
- Irreproducible results



The solution?

Free

Open

Source

Software

Free as in beer - gratis

```
/*
 * -----
 *   * "THE BEER-WARE LICENSE" (Revision 42):
 *   * <phk@FreeBSD.ORG> wrote this file. As long as you
 *   * retain this notice you can do whatever you want with
 *   * this stuff. If we meet some day, and you think this
 *   * stuff is worth it, you can buy me a beer in return
 *   * Poul-Henning Kamp
 * -----
 * /
```

BEERWARE

Free as in speech - libre



5 You should write collaborative open source software

Open Source Software



Open Source Licenses

- Apache License 2.0
- BSD 3-Clause “New” or “Revised” license
- BSD 2-Clause “Simplified” or “FreeBSD” license
- GNU General Public License (GPL)
- GNU Library or “Lesser” General Public License (LGPL)
- MIT License
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

<http://opensource.org/licenses/>

Keep on trying

- Written by a post-doc
- The source is the documentation
- It's on the ~~cluster~~ department cvs server
- The post-doc runs it for everybody
- Post-doc gets a “real job”
- Archaeology reveals a VMS binary
- Irreproducible results



Collaborative software

Intentional group processes plus software to support them



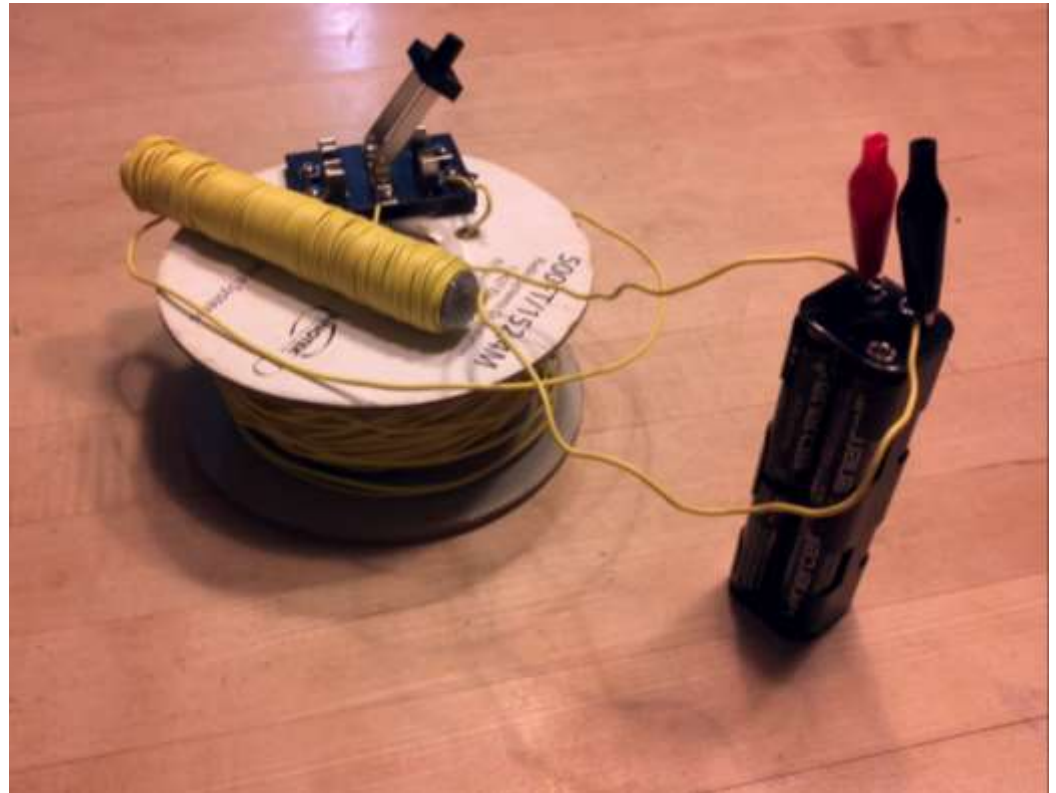
Software with a shared source

Collaborative Open Source Software



Required tools

- Source Control Management
- Issue Tracking
- Build System
- Communication
- Documentation



SCM

- RCS - <http://www.gnu.org/software/rcs/rcs.html>
- CVS - <http://www.nongnu.org/cvs/>
- Subversion - <http://www.apache.org/dev/version-control.html>
- Git - <http://www.git-scm.com>
- Mercurial – <http://mercurial.selenic.com>
- Bazaar - <http://bazaar.canonical.com>

Issue tracking

- Bugzilla
- Fossil – <http://fossil-scm.org>
- Github issues
- Jira
- Trac



Build system

- Ant
- Autoconf/Autotools
- Cmake
- Make/NMake
- Maven
- Ninja
- Rake
- Scons



Continuous Integration Systems

- Buildbot - <http://buildbot.net/>
- Continuum
- Hudson/Jenkins
- Team City - <http://www.jetbrains.com/teamcity/>
- Travis-CI



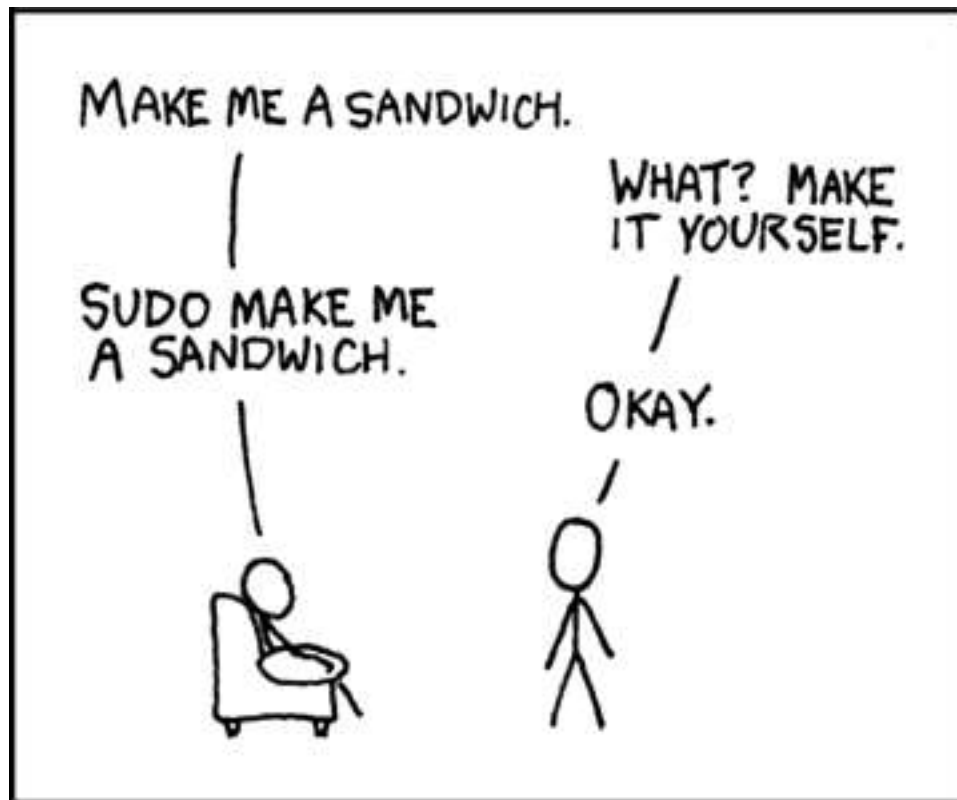
Communication

- Meetings
- Phone calls
- Chat
- IRC
- Blogs



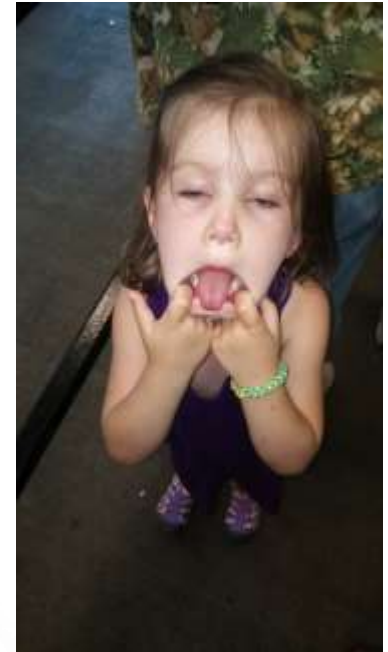
Developer documentation

- Doxygen
- Javadocs
- Sphinx
- Wiki



User documentation

- Blog
- Docbook
- LaTeX
- Sphinx
- Wiki
- Word



Success

- Written by a couple of post-docs
- ~~The source is the~~ documentation is in the source and here is a manual
- It's on ~~cluster department cvs server~~ github™
- The post-doc runs ~~it for~~ from everybody
- Post-doc gets a “real job”
- ~~Archaeology reveals a VMS binary~~ Build system
- ~~Irreproducible~~ results



The CommitStrip Test



THE COMMITSTRIP TEST

Coding environment quality criteria

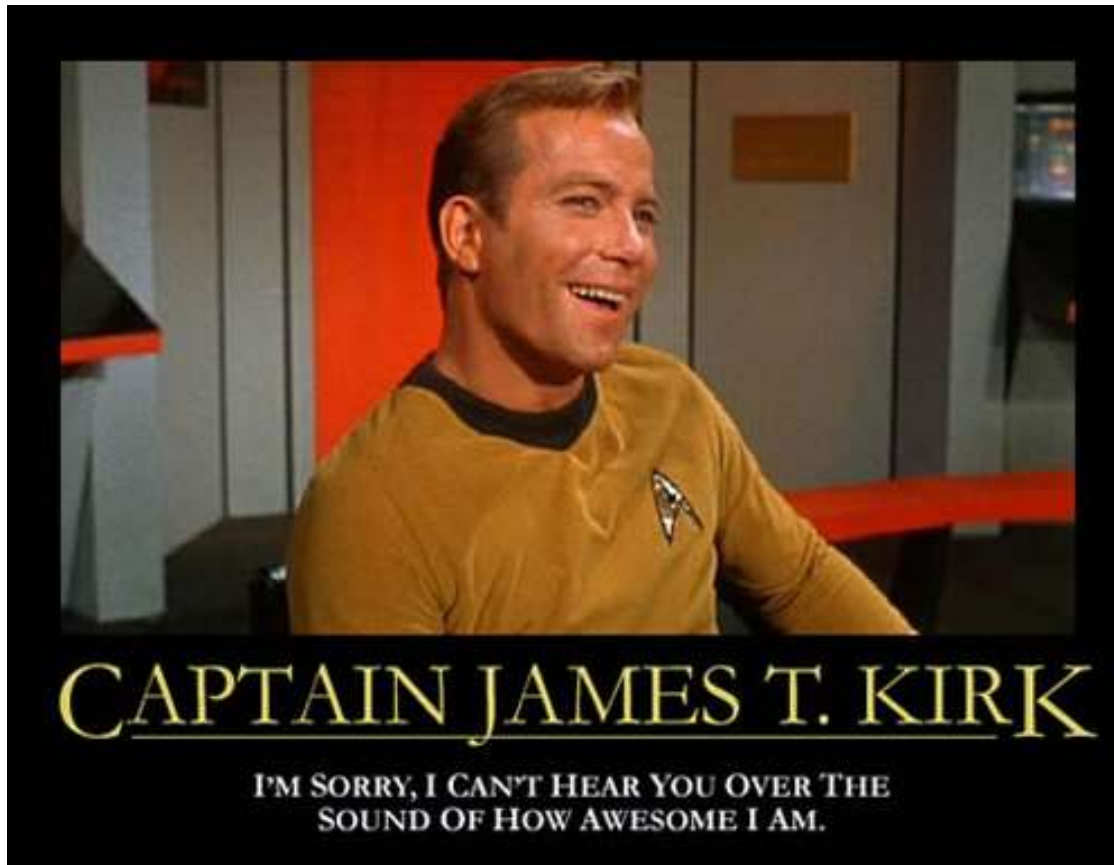
- Unlimited snacks and sodas. Coding burns down calories
- A workstation where nobody can see your screen. It's all about respecting privacy
- 50% of coders should be women. Equal rights are important
- A Gigabit Internet connection, to commit at decent speeds
- A high-end 2GB graphics card, in order to compile faster!
- An intern to fix trivial bugs

- An office you can lock yourself into. Because sometimes you need more focus
- 4 screens: one for coding, one for rendering, one for debug and another one for... well, 4 is better
- A second week-end. Because coding is hard
- USB and ethernet plus in the bathroom. Because each commit counts
- Salary should be directly calculated on the number of commits in order to reward performance
- A sports facility to get some exercise and stay fit. But with TV and wifi otherwise you get bored

And those are just the most important things!

CommitStrip.com

<http://www.commitstrip.com/en/2014/09/16/le-commitstrip-test/>



The Joel Test

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?

5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?