

FLOSSing Your Bits

A student's guide to starting an open source project

Andy Price

SUCS Talks — 2007.05.15

Jargon

- Free Software
- OSS (Open Source Software)
- FLOSS (Free, Libre, Open Source Software)
- We'll use “open source” here as a generic term

Perspective

- Partly from own experience
- Partly from observations
- The rest is common sense
- Student-orientated
- Minimal and idealistic

Setting The Scene

- So you want to start an open source project?
- You're new to this open source thingamabob
- You have an idea for a project
- You have coding skill
- You have some spare time

Choose A Good Name

Examples of good names:

- Inkscape - Snappy, suggests function
- Firefox - Creates memorable imagery
- Sendmail - Does exactly what it says on the tin

Examples of bad names:

- Gaim - AOL trademark craziness
- Wmii - Pronunciation ambiguity
- Gimp - As seen in Pulp Fiction?

Choose A License

- GPL
- BSD
- Mozilla
- Apache
- Creative Commons
- WTFPL (Do What The F*** You Want To Public License)
- ... etc...

http://en.wikipedia.org/wiki/Category:Open_source_licenses
(44 listed)

Organise Your Code

- Set up a version control system - svn, cvs, bzd, etc.
- Provide methods of branching and tagging
- Import code (if applicable) and add license
- Write README file and other docs
- Consider version scheme
- Consider build/distribution methods - make, autotools, distutils, etc.

Create A Face

- Simple web page outlining your program's functionality
- Maybe a spot of artwork - logos, mascots
- Link to static download location - don't confuse users
- Highlight current state of project
- List big future plans

Caution

- Projects often take a long time to attract attention
- Be patient
- Be transparent and open (document changes etc.)
- Don't overdo it at the start
- Don't judge your project's importance
- Don't be code shy, bugs happen
- Get on with it and have fun

Things Users Appreciate

- Openness - Emit plenty of information
- Appreciation - Make users feel wanted (if they are)
- Predictability - Plenty of notice for big changes
- Consideration - Make decisions with users in mind
- Backward compatibility - Don't break on upgrade
- User docs!
- Character - Don't come over as "beige" ;)

Things Developers Appreciate

- (See: Things Users Appreciate)
- Decisiveness - Stick to your guns
- Pragmatism - Solve problems intelligently and efficiently
- Enthusiasm - It's contagious
- Direction - Disseminate your plans for the project
- Readable code and developer docs!

Releases

- Release often, regularity aids predictability
- Tag releases for future reference
- Distribute code in “standard” format (tar -czf or tar -cjf etc.)
- Remember: Each new release is a call for users to break your program and report bugs

Routine Maintenance

- Update code for new language versions
- Keep copyright, README, webpage, contact details etc. up-to-date
- Keep an eye on library releases and API changes
- Research the field and look for inspiration

Reaching A Wider Audience

- Linux distros - Submit packages for inclusion
- Raise your own profile by contributing to other projects, for example
- Blog regularly about the project
- Keep the improvements and releases coming

Managing Community Feedback

- Use a bug tracker - trac, bugzilla, launchpad, mantis, etc.
- Mailing list - Get your users interacting
- Accept criticism
- Review patches
- Offer tasks/challenges

At The End Of The Day

- The code speaks for itself.

Any Questions?

Any questions/comments?