

Managing Quality in an Open Source Project

By Jeremy Allison

Samba Team



email: jeremy@netcom.com

Trust Your Users

- Open Source doesn't mean better programmers, it means better users.
 - User bug reports are the most valuable quality feedback you have. Cherish them.
 - Samba uses Jitterbug - another Open Source tool - to collect and reply to user bugs.
 - Jitterbug is now used by Linus and the GNOME developers.
- Open Source empowers end users to help you maintain high quality.
 - Samba patch quality varies enormously, but even incorrect patches often reveal the actual problem.

Maintain Strict Revision Control

- Necessary for any complex software project.
 - Samba learned this lesson fairly early on.
- Revision control allows easy duplication of user bugs.
 - Samba has more problems with variations in Microsoft clients - "DLL hell" where every new Microsoft application is an OS upgrade.
- Samba developers use Open Source software
 - CVS for revision control.
 - ssh for secure remote source code access.

Keep Control of Master Source Code Write Access

- Only "core" contributors have code write access.
 - Even though Samba has a "bazaar" style development model, a core team is needed to review code changes.
- Patch contributions go into the master source code via one or more members of the core team.
- Defer patch decisions to the Team Lead (this can lead to bruised egos).
- When regular patch contributors appear, invite them into the core team.
- Give feedback to patch contributors.

Peer Review ALL Submissions

- As Samba is a daemon running as "root", security is paramount. All code contributions must pass a security review.
 - Unilateral decision was taken to ban the use of certain 'C' language functions that lead to buffer overflow.
- Appoint subject area experts, but the Team Lead must be able to override their decisions.
 - Team Lead is "benevolent dictator".
- Promote "egoless" programming.
 - Don't get hung up on formatting style.
 - Everyone writes buggy code.
 - Finding bugs is a group effort with your users.

Manage the Core Team

- People management is harder than writing code.
 - Email discussions suck up hours of time.
 - People get easily offended when a decision goes against them.
- Samba uses the IETF mantra of "rough consensus and working code".
 - If contributors disagree, let them fight it out in implementation, may the best win !
- Allow people to fall into their natural roles.
 - Samba has a release/QA manager, a design specialist, an advanced projects coder, etc.

Keep the Documentation Current

- Users who want to contribute but cannot code are often the most useful here.
- Man pages should always match the code for a current release, Samba is good at this (after a shaky start).
- Provide a project roadmap so users are aware of your future plans. Users need to know the software is being actively developed and has a future.
- If the project is very successful, commercial authors will help document the software.

Don't Blow it if You Go Commercial

- Use a license that achieves the authors aims
 - Samba uses the GPL, this has served us well.
- Many users will want to pay for the software.
 - Samba Team has jumped through hoops to provide zero dollar invoices to companies who are prohibited from using "freeware".
- Samba provides a list of commercial support consultants with the software.
 - This is not an endorsement from the Samba Team, but allows users and consultants to get in touch.
- Keep code ownership with the community of authors and users.

References

- Samba Web site:
 - <http://samba.anu.edu.au/samba/samba.html>
- Samba survey - please register:
 - <http://samba.anu.edu.au/samba/survey/>
- Samba consultants list for commercial support:
 - <ftp://samba.anu.edu.au/pub/samba/Support.txt>
- Samba book available from Computer Literacy:
 - <http://www.clbooks.com>