

Cathedral and the Bazaar

April 6, 2005

This lecture is based on “The Cathedral and the Bazaar” by Eric Steven Raymond. <http://sagan.earth-space.net/~esr/writings/cathedral-bazaar/>

I. Basics

The *Cathedral* is traditional software engineering: build a large product with a small number of designers, have a few releases of high quality, and keep your source code to yourself.

The *Bazaar* is very loose organic development by hundreds or thousands of programmers with very little centralized management. It is characterized by very frequent releases from many sources, all of which are imperfect but provide an incredible momentum of improvement. Linux is defining case study of the Bazaar model.

It is also useful to think it terms of human engineering versus natural (evolutionary) engineering. Cathedrals (the buildings) or skyscrapers are incredible feats of human engineering, but they are not as robust as organisms that have evolved naturally, even though the latter are individually imperfect and each generation is not far from the last.

Another useful organic analogy is the human antibody system. Linux programmers are like antibodies: individually they have limited impact and can only address certain kinds of bugs, but their sheer number and diversity provides an overall solution that is incredibly effective at finding and destroying bugs. By this analogy effectiveness depends on the number of programmers, the diversity of their interests, and the number of contacts (releases).

II. Secrets of the Bazaar

- o Release early and often (and publicly!)
- o Remind people of progress, of the project, and of their importance to it (by public recognition)
- o Maximize the number of people on the developer’s list, keep them interested/excited
- o Enable users to find and fix bugs (hence the source code)
- o Take users *very* seriously; “treat them as your most valuable resource and they will become your most valuable resource”. Ask for their feedback and take their suggestions seriously.
- o All bugs are easy for someone, just need enough people...

III. General Lessons

- o throw one away
- o conservative, simple designs are best -- complexity kills; evolution not revolution

Cathedral and the Bazaar

- o stable vs leading edge
- o smart data structure, dumb code
- o Brooks' Law: complexity/cost rises quadratically, work only linearly; The source code is the primary form of group communication: documentation matters more not less. This is why it can scale better than adding more group members.
- o Egoless programming (Weinberg): share code (not territorial), ask for improvements, leads to faster overall improvement

IV. Leadership

- o Can't lead based on authority, only vision and charm!!!
- o Shouldn't lead based on authority *even when you can*
- o Tools: compelling vision, visible signs of progress (frequent e-mail and releases), explicit recognition of contributions (rewards)
- o Transition to new leader before you get burned out or want to move on...

V. Netscape

- o Navigator is now open source
- o Question: will it be better than MSIE? Depends primarily on the number of developers and whether they work on Windows. (The Linux version is already better :)
- o long-awaited 5.0 release is finally coming....

VI. Halloween Documents

- o scary part: de-commoditizing protocols...
- o admits that Linux can be equal/better quality for servers