



## Open Source

- I. What is It?
  - a. Source Freely Available
  - b. If it doesn't work, anyone can change it
  - c. Eric Raymond (big champion)
    - i. Cathedral and Bazaar
    - ii. Bazaar – wide range of approaches; get huge collection of “stuff” you can use for a program (take what you need)
  - d. Well-Known Open Source Software
    - i. Utilities
      1. Linux – most famous
      2. emacs, gcc
      3. Apache – affects people the most, though they probably don't know it
      4. fetchmail
    - ii. Enterprise Software: MySQL, DFS filesystem, Websphere Studio
    - iii. Applications: Star Office, gimp
    - iv. Older games get re-released as open source once all the code's been rewritten
- II. How it Works
  - a. Release Early, Release Often
  - b. Early release gets people interested but shows that it still needs work.
  - c. If it's too polished on the initial release nobody will bother working on it.
  - d. Users make changes; submit them. Release with changes often.
  - e. Contributors
    - i. Up to 1000s, usually on a few (10 to 15).
    - ii. Almost exclusively part-time / volunteer
- III. Maintenance
  - a. Bug Fixes. Requested Enhancements. Developer Originated
  - b. Software may mutate drastically from the original intent as people suggest changes
- IV. Don'ts
  - a. Forking Projects. Splits developer base in half. Generates confusion about versions
    - i. Good when you really need a specialized version.
    - ii. Sometimes merged back in later; sometimes stays a fork.
  - b. Distributing Rogue Versions
    - i. Make sure anyone running it knows it's different.
    - ii. Once you start releasing a Rogue version you're really forking the project
  - c. *Never* remove credit!
  - d. Won't be taken to court over these things but there's major social pressure
- V. Organization
  - a. Usually have a maintainer coordinating changes
  - b. Initially the maintainer is the original developer, who can then appoint a successor
  - c. Projects can get orphaned. If nobody's doing anything on a project, adopt it!
  - d. Good for many reasons to work on open source projects as a student. Build a reputation!
  - e. Committee
    - i. Big projects (Linux, Apache, Perl)
    - ii. Hierarchical (like Linux, with Torvalds at the top, then specialists below)
    - iii. Committee with Specialists (Apache)
    - iv. Rotating Committee Leaders (Perl, possibly)
- VI. Overhead
  - a. Brooks' Problem: Overhead increases  $O(N^2)$  by the number of developers
  - b. Debugging / testing distributes pretty well. *Someone* will find the problem and announce it. Then *someone* will solve it.
  - c. It's inefficient in that many people are doing the same work, but very productive!
- VII. Licensing
  - a. GNU: Once open source, Always open source. Can't close it off.

- b. Berkeley: Can turn into closed source later
  - c. Sun Community: *Cannot* fork. Takes scrutiny, strangely. It's really bad to fork a project, but apparently it's even worse to legally forbid it.
- VIII. Why Bother?
- a. Seems really *weird* to have people working for free.
  - b. One answer: it's fun for developers!
  - c. Another: want to gain favor with who's in control
  - d. Raymond: *Time* is in excess right now (developer time). "Gift Culture" (vs Control vs. Exchange). Basic economics.
  - e. Owners
    - i. Get attention! Mozilla was the first big project to go from commercial to open source.
    - ii. Get free labor!
    - iii. Get confidence from the customer
      - 1. Big customers might get closed source software in escrow in case the vendor dies (so the customer can take over the project or give it to another developer).
      - 2. With open source, customer knows the source is always available
      - 3. Also see boost in reliability
- IX. Issues
- a. What programs work?
    - i. No trade secrets.
    - ii. Needs low or no sale value
  - b. Microsoft's Halloween Document
    - i. Open source can't innovate
    - ii. True in many ways – open source trails in many areas
  - c. What will Get Developed?
    - i. Need programmer interest
    - ii. Targeted at novices and kids – issues there