What is open source?

Dept. of Electrical Engineering and Computer Science
Academic Year 2011-2012
What is open source?

• It is not about software.
• It is the process by which software is created.
• Software is the artifact of applying the process.
In order to understand open source, we need answers to:

- Who are the people who write open source code?
- What do these people do, exactly?
- How do they collaborate with each other?
- How do they resolve disagreements and deal with conflict?
The Software “Problem”

- This is a hard problem because:
  - An auto engineer can envision with great accuracy how and in what conditions a car will be driven.
  - A software engineer cannot envision so freely.

_A piece of software can be used in radically different ways, hardening any assumptions to be made during the design process._
The Software “Problem”

- The essence of software design, like the writing of poetry, is a creative process.
The Software “Problem”

- Place a young eccentric in an isolated basement room with a computer and lots of coffee, and let write software until the point of exhaustion?
- How about an OS that contains millions of lines of code?
The Software “Problem”

- We need **division of labor**. How do we divide?
The Software “Problem”

- First, the problem solution needs to be conceptually architected.
- A small architecture team needs to consist of resonant minds:

  This is essential for “Conceptual integrity”.
The Software “Problem”

- Architecture by Master Architect.
  - Produces a master plan.
  - Dictates division of labor.
- Implementation. Sub-architect, Engineers.
  - Produces a sub-master plan.
  - Develops each independent component.
The Software “Problem”

• Brooks’s Law:

Adding more manpower to a software project that is behind schedule will put the project even far behind.
The Software “Problem”

- Brooks says that as the number of programmers increase, the possibility of failure scales quadratically.

  because the number of communication paths increases at this rate.

- We know the software problem is a rather difficult one, how does open source can help?
Open source process

- The key element of the process is volunteering.
  - voluntary participation.
  - voluntary selection of tasks.
- anyone can join an open source project.
- anyone can leave at any time.
- each person is free to choose what he wishes to work on or to contribute.
Open source process

- There is a core code base. And it is available freely.
- Anyone can obtain and modify.
- From this point on, licensing makes a difference.
  - BSD license: minimally constraining, e.g., creating a proprietary product.
  - GPL license: more constraining, e.g., you cannot create a proprietary product and hide the source code.
Open source process

• Coding practices
  • In a typical BSD style process, a small and a core team of developers write most of the code. Source code is freely available for anyone.
  • In a Linux style process, there is no distinction. There is an army of developers, and they can work on any piece and submit changes. However such changes are evaluated through a methodical process.
  • In a Linux style process, the norm is to be extremely vocal, and be not afraid of what others might say.
Open source process

- What happens when your patch gets rejected?
  - You can take the source code, add your patch, and branch out, create a new fork.
  - Everyone has this right.
Open source process

• The fundamental question is:

• why would someone contribute time and effort to write code that they do not copyright and for which they will not be directly compensated for?

• this is not a matter of having the right people on board, or the number of people on board. it is the organizational nature of the underlying process itself that drives the success.
Open source process

• Who participates in the process?
  • a high school student writing patches, you who is willing to contribute to an existing open source community, ..., so who?

• SourceForge: As of Sept 2003, ~67K projects, ~600K active users.

• Counter.li.org counts the number of active Linux users: guesstimate is ~18M as of May 2003.
Open source process

• Who participates in the process?
  • Studies have shown that Linux contributors are geographically dispersed, and notably international, and large in number.
  • Linux version 1.0 had 78 individual developers, and 12 countries.
Figure 1  Linux code authors listed in first credits file (1994), concentration by country.
Open source process

Figure 4 Linux software map (LSM) entries, by top-level domain of email address.
Open source process

• 10% of the developers are credited with about 72% of the code.
• Top 10 individuals are responsible for 20% of the total code.
• Contributions to Linux, Gnome etc. conform to this pattern.
Open source process

• What do they do?
  • **Make it interesting and make sure it happens.**
  • Open Source developers look for opportunities to create new and exciting functions or do hard things in an elegant way.
  • The number of volunteers is huge, and they have different interests and expertise.
Open source process

• What do they do?
  • Volunteers create cool stuff, but it is cooler if others say so, it is even cooler.
  • Project leaders need to create opportunities for growth, intellectual satisfaction, and the likes.
Open source process

- What do they do?
  - Scratch an itch.
Open source process

• What do they do?

  • **Minimize how many times you have to reinvent the wheel.**

  • A good programmer is “lazy like a fox”.

  • It is so hard and time consuming to write good code, the lazy fox is always searching for efficiencies.
Open source process

• What do they do?
  • Minimize how many times you have to reinvent the wheel.
  • Code moves freely without any boundaries.
  • There is no vendor lock-in problem.
Open source process

- What do they do?
  - Solve problems through parallel work processes whenever possible.
- Traditional approach is Engineering. The architect sketches the course to take towards a solution.
- Open source relies on Evolution where many people are trying to come up with a solution while lacking a central authority dictating which way to go.
Open source process

- What do they do?
  - Leverage the law of large numbers.
  - Traditional software: Field testing is tricky. Buggy code can be tested by users on beta versions. But buggy product may hurt the brand.
  - Open source software is always in beta stage. A bug is not a problem, but rather it is a challenge to take on.
Open source process

• What do they do?

• **Document what you do.**

• In a traditional environment, documentation has less importance. Knowledge transfers through other channels.

• However, in open source, documentation is king!

• Documentation in open source is a means of transferring what the author knows across time as well as space.
Open source process

- What do they do?
- Document what you do.
- Writing code that is easier to document. That should be the philosophy.
Open source process

• What do they do?

• Release early and release often.
Open source process

• What do they do?

• **Talk a lot.**

• Open source developers love to talk about what it is they are doing and why.

• In 1992, Linus Torvalds wrote to Andrew Tanenbaum: “Linux still beats the pants off Minix in almost all areas.. your job is being a professor and a researcher, that is one hell of a good excuse for some of the brain-damages of Minix.”