CS 208

Software Engineering

Open Source Software Development

Abhishek Srivastava

Indian Institute of Technology Indore
Email: asrivastava@iiti.ac.in

Open Source Software

php
MySQL
@pen Webmail
The Apache Software Foundation
http://www.apache.org/
FCK the text editor for Internet editor
fedora
phpMyAdmin

"Generically, *open source* refers to a program in which the source code is available to the general public for use and/or modification from its original design *usually* free of charge. Open source code is typically created as a collaborative effort in which programmers improve upon the code and share the changes within the community."

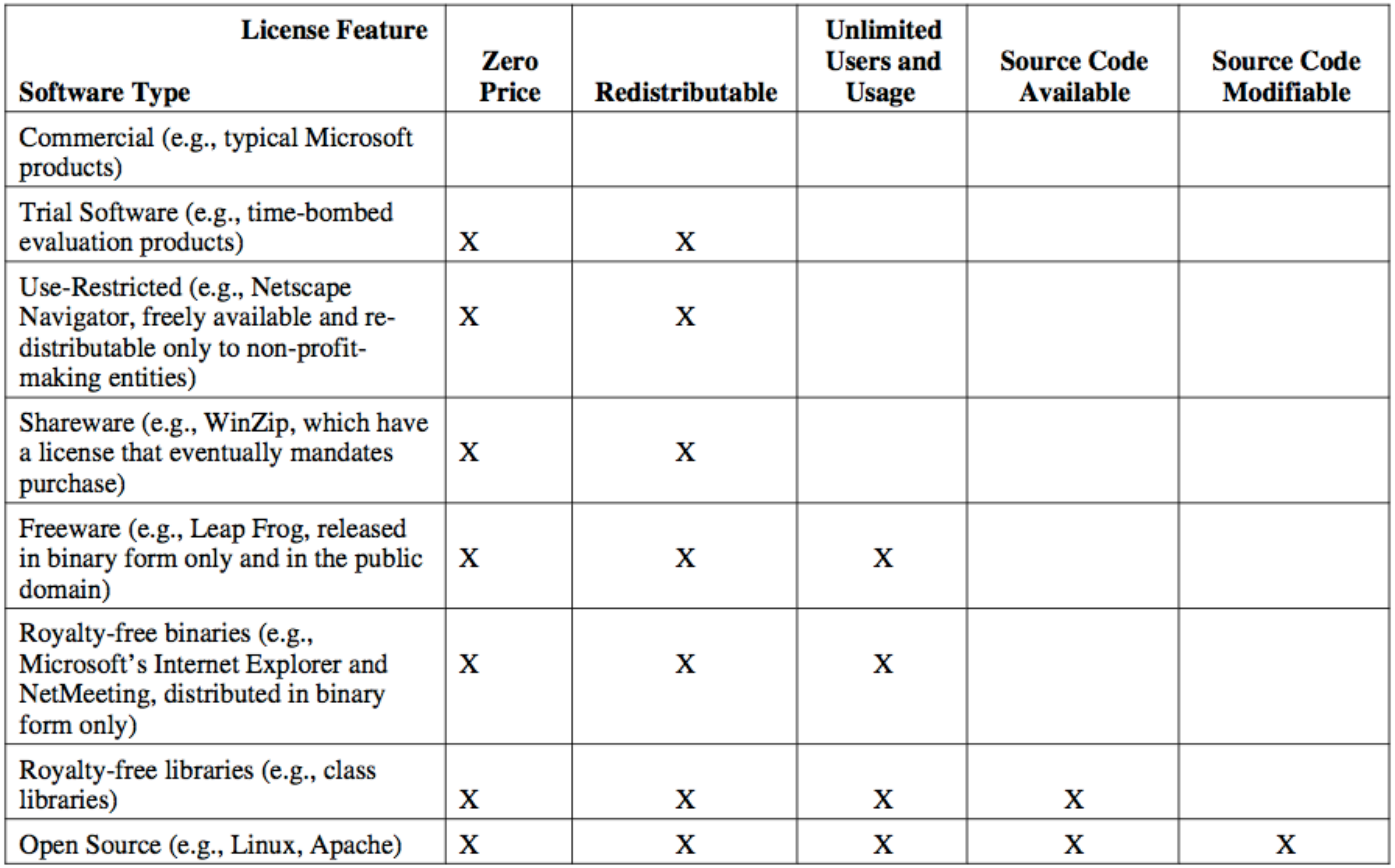
- Webopedia

What is Open Source Software?

- The source code must be available to user
- The software must be redistributable
- The software must be modifiable and creation of derivative works must be permitted
- The license must not discriminate against any user, group of users, or field of endeavour
- The license must apply to all parties to whom the software is distributed

- Feller and Fitzgerald

Software Licensing Taxonomy



License Feature / Software Type	Zero Price	Redistributable	Unlimited Users and Usage	Source Code Available	Source Code Modifiable
Commercial (e.g., typical Microsoft products)					
Trial Software (e.g., time-bombed evaluation products)	X	X			
Use-Restricted (e.g., Netscape Navigator, freely available and re-distributable only to non-profit-making entities)	X	X			
Shareware (e.g., WinZip, which have a license that eventually mandates purchase)	X	X			
Freeware (e.g., Leap Frog, released in binary form only and in the public domain)	X	X	X		
Royalty-free binaries (e.g., Microsoft's Internet Explorer and NetMeeting, distributed in binary form only)	X	X	X		
Royalty-free libraries (e.g., class libraries)	X	X	X	X	
Open Source (e.g., Linux, Apache)	X	X	X	X	X

- The Halloween Documents, 1998

Open Source Development Eras

The First Era (early 1960s to early 1980s)

- development of operating systems and the Internet in academic settings
- common to share code between programmers of different organizations (code sharing was informal)
- code sharing was accelerated with the diffusion of 'Usenet'
- informal code sharing became an issue when AT&T started enforcing intellectual property rights related to Unix

- Lerner and Tirole

Open Source Development Eras

The Second Era (early 1980s to early 1990s)

- to deal with the AT&T litigation, efforts began towards formalization of code sharing
- a formal license procedure called the ‘General Public License’ which obliged developers to make code freely available in exchange of being allowed to modify it
- the big restriction in the General Public License was that any code that intermingled with the cooperative code also had to be licensed on the same terms

- Lerner and Tirole

Open Source Development Eras

The Third Era (early 1990s -)

- the Internet led to a dramatic rise in the open source activity
- cooperation between commercial organizations and the open source community became common
- a number of less restrictive licensing like the ‘Debian Software Guidelines’ came about that allowed intermingling between open source and proprietary software

- Lerner and Tirole

History of Open Source Software

1950s and 1960s	Software source code is distributed without restrictions in IBM and DEC user groups, ACM's Algorithms Section etc.
1969	Ken Thompson writes the first version of UNIX. Its source code is distributed freely throughout the seventies.
1978	Donald Knuth (Stanford) publishes TEX as free software
1979	Following A T&T' s announcement to commercialize UNIX, UC Berkeley begins with the creation of its own version of UNIX, BSD (Berkeley Software Distribution). Eric Allmann, a student at UC Berkely develops a program that routes messages between computers over ARPANET. It later evolves into Sendmail.
1983	Stallmann publishes GNU Manifesto calling for free software, and establishes Free Software Foundation.
1986	Larry Wall creates Perl (Practical Extraction and Report Language), a versatile programming language used for writing CGI (Common Gateway Interface) scripts.
1987	Developer Andrew Tanenbaum releases Minix, a version of UNIX for the PC, Mac, Amiga, and Atari ST. It comes with complete source code.

- Hars and Ou

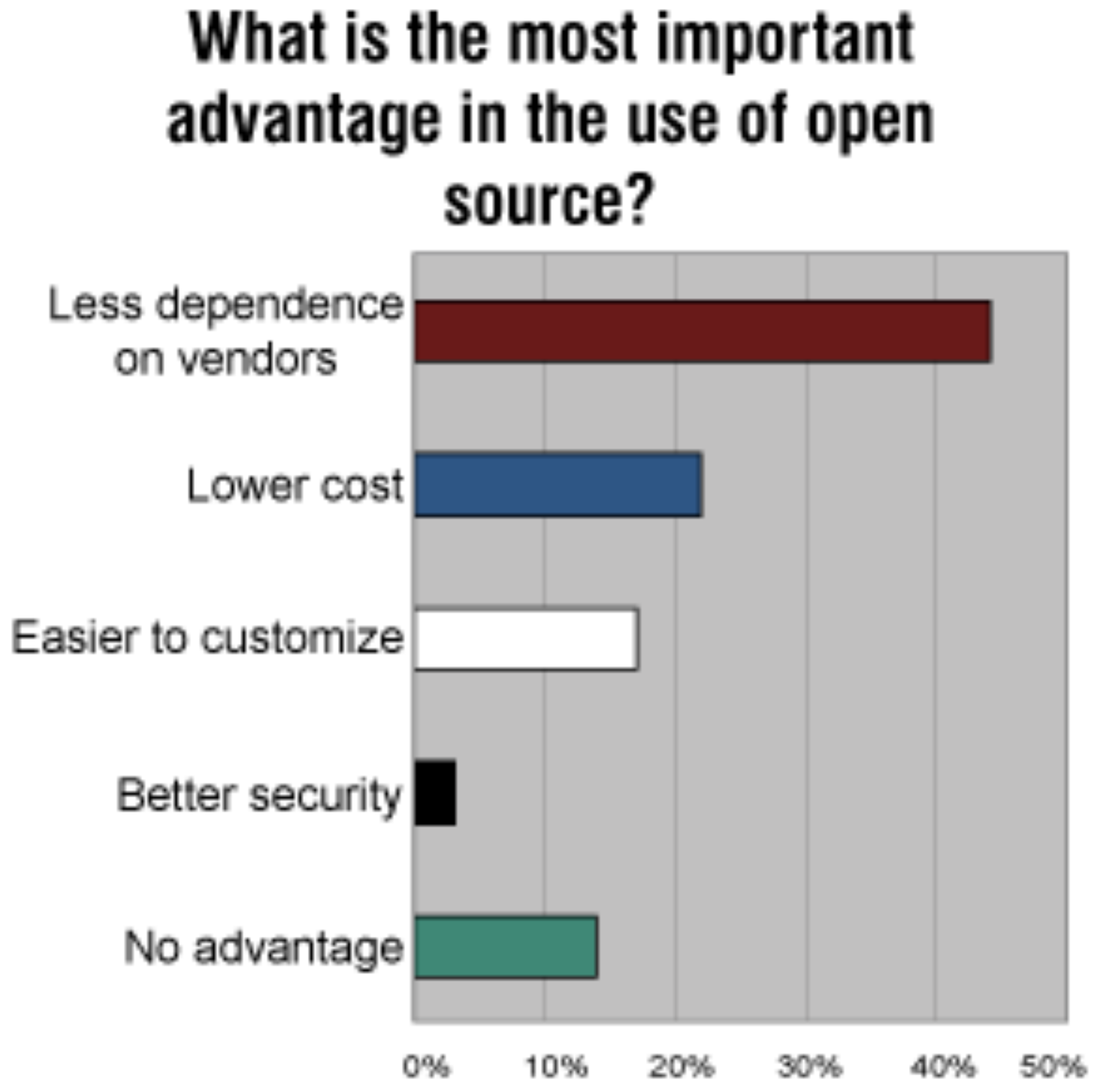
1991	Linus Torvalds publishes version 0.02 of a new UNIX variant that he calls Linux in a Minix newsgroup.
1993	FreeBSD 1.0 is released. Based on BSD Unix, FreeBSD includes networking, virtual memory, task switching, and large filenames. Ian Murdock creates a new linux distribution called Debian Linux.
1994	Marc Ewing forms Red Hat Linux. It quickly becomes the leading Linux distributor. Bryan Sparks founds Caldera with backing by former Novell CEO Ray Noorda.
1995	The Apache Group builds a new Web server, Apache, based on the National Center for Supercomputing Applications' (NCSA's) HTTPd 1.3 and a series of patch files. It has become the dominant HTTP server today.
1998	Netscape not only gives away Communicator 5.0 (Mozilla) but also releases its source code. Major software vendors, including Computer Associates, Corel, IBM, Informix, Interbase, Oracle, and Sybase, announce plans to port their products to Linux. Sun announces plans to release the source code for Java 2 to developers.
2000	More software companies such as Novell and Real release versions of their products which run on Linux.

Benefits of using open source software

- Quality
- Customizability
- Freedom
- Support Options
- Cost
- Try before you buy

- PCWorld, Nov. 2010

Benefits of using open source software



- Computer Economics, May 2005

Open Source Development Philosophy

- Users should be treated as co-developers
- Early releases
- Frequent integration
- Several versions
- High modularisation
- Dynamic decision making structure

- Gregorio Robles (based on the ‘bazaar model’)

Motivation for participation

Internal factors

- intrinsic motivation
- altruism
- community identification

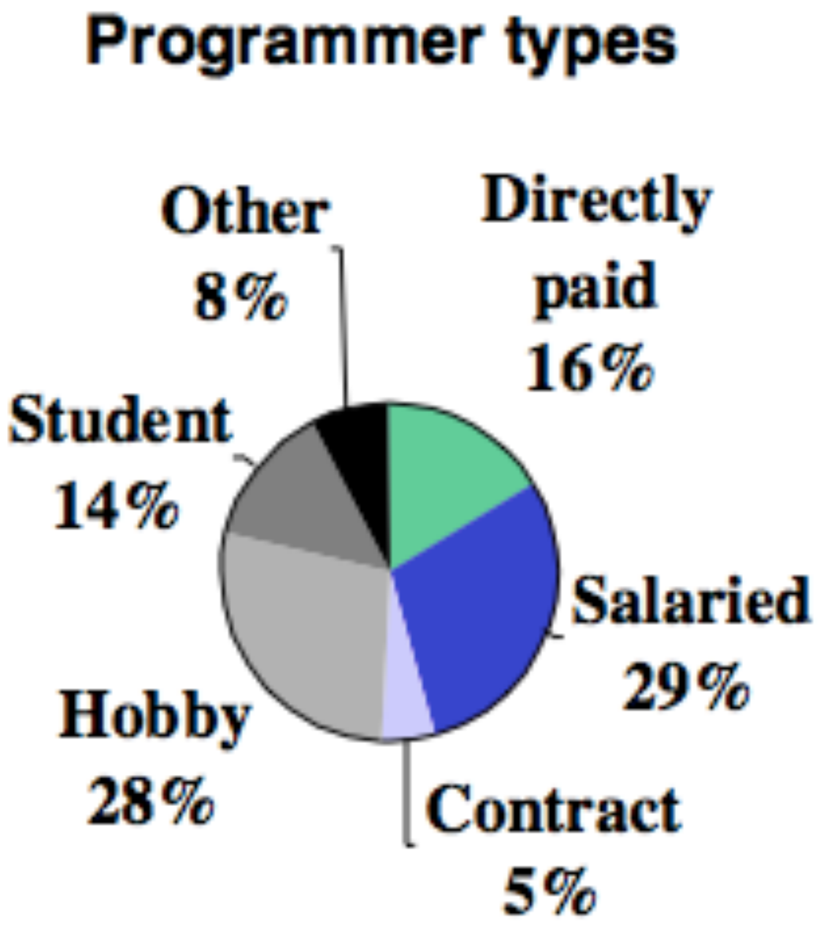
External factors

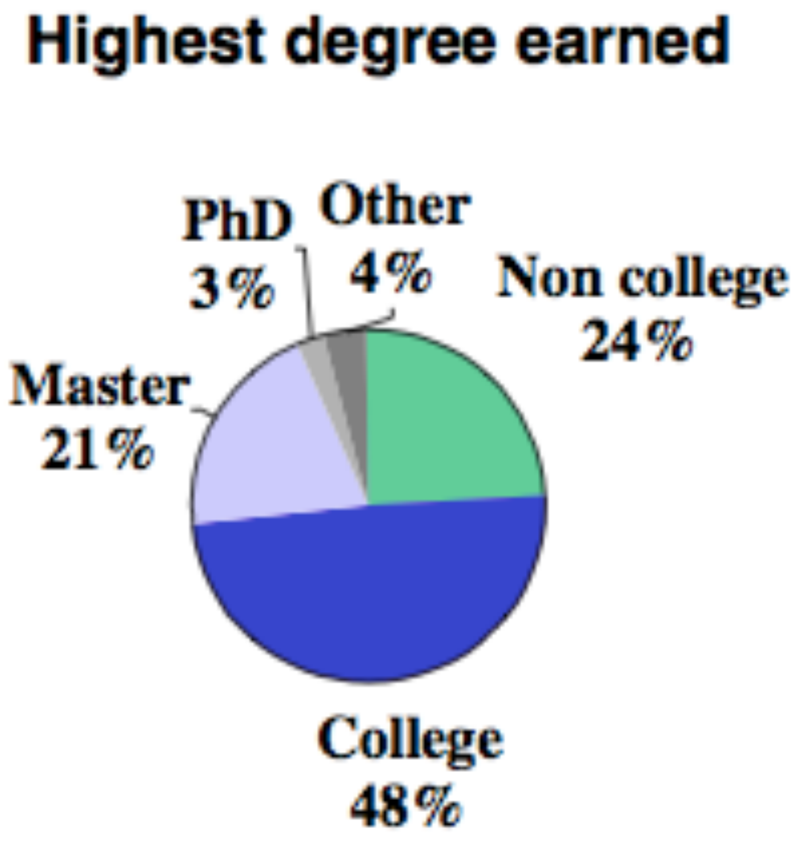
- Future rewards
  - revenue related products
  - human capital
  - self marketing
  - peer recognition
- Personal needs

- Hars and Ou

Survey: Motivation

Respondent demography

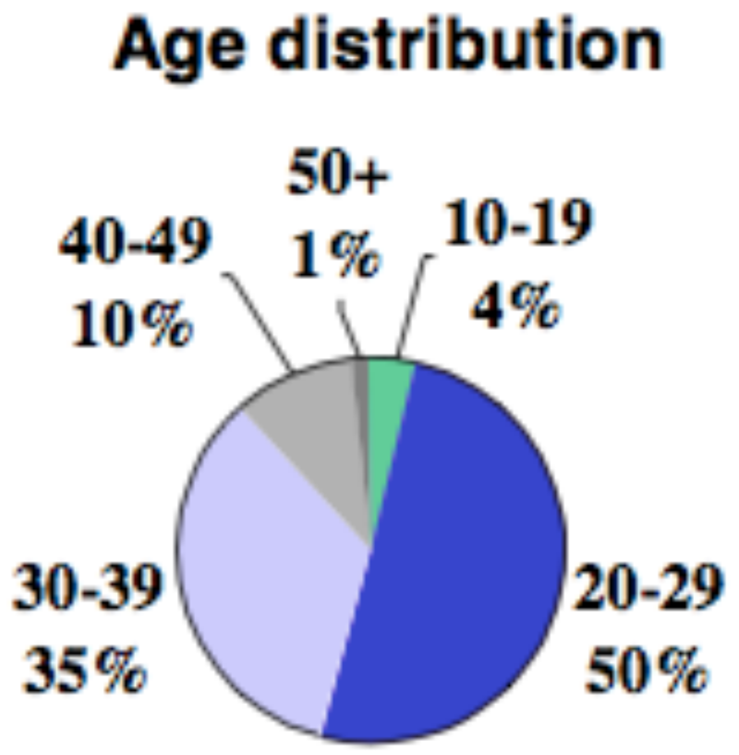


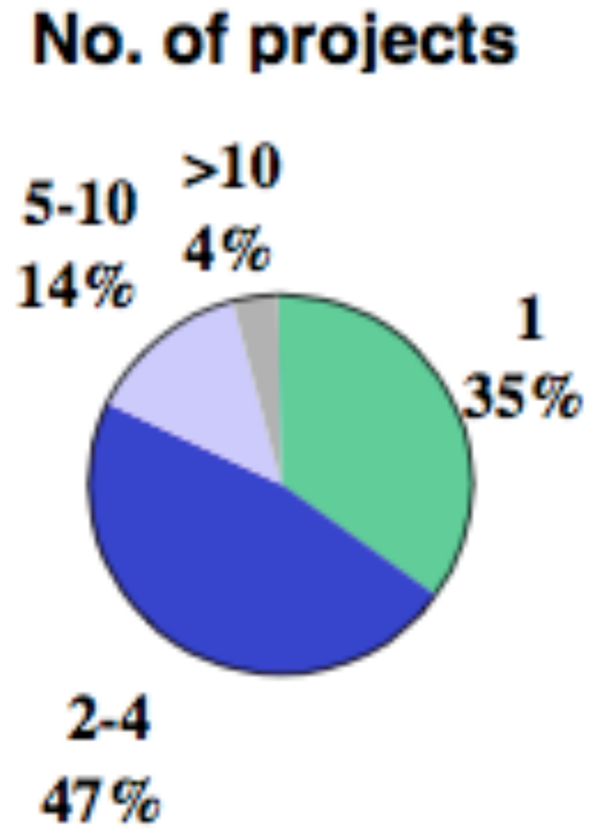


- Hars and Ou

Survey: Motivation

Respondent demography





- Hars and Ou

Myths and Fears about Open Source

- Aren't OSS simply plagiarised proprietary programs?
- Does OSS expose you to greater risk of abandonment?
- Is OSS economically viable?
- Will OSS destroy the software industry? Will software engineers starve if many programs become OSS/FS?
- Is OSS compatible with capitalism?
- Are OSS programs compatible with standards?
- Is OSS a destroyer of Intellectual Property?

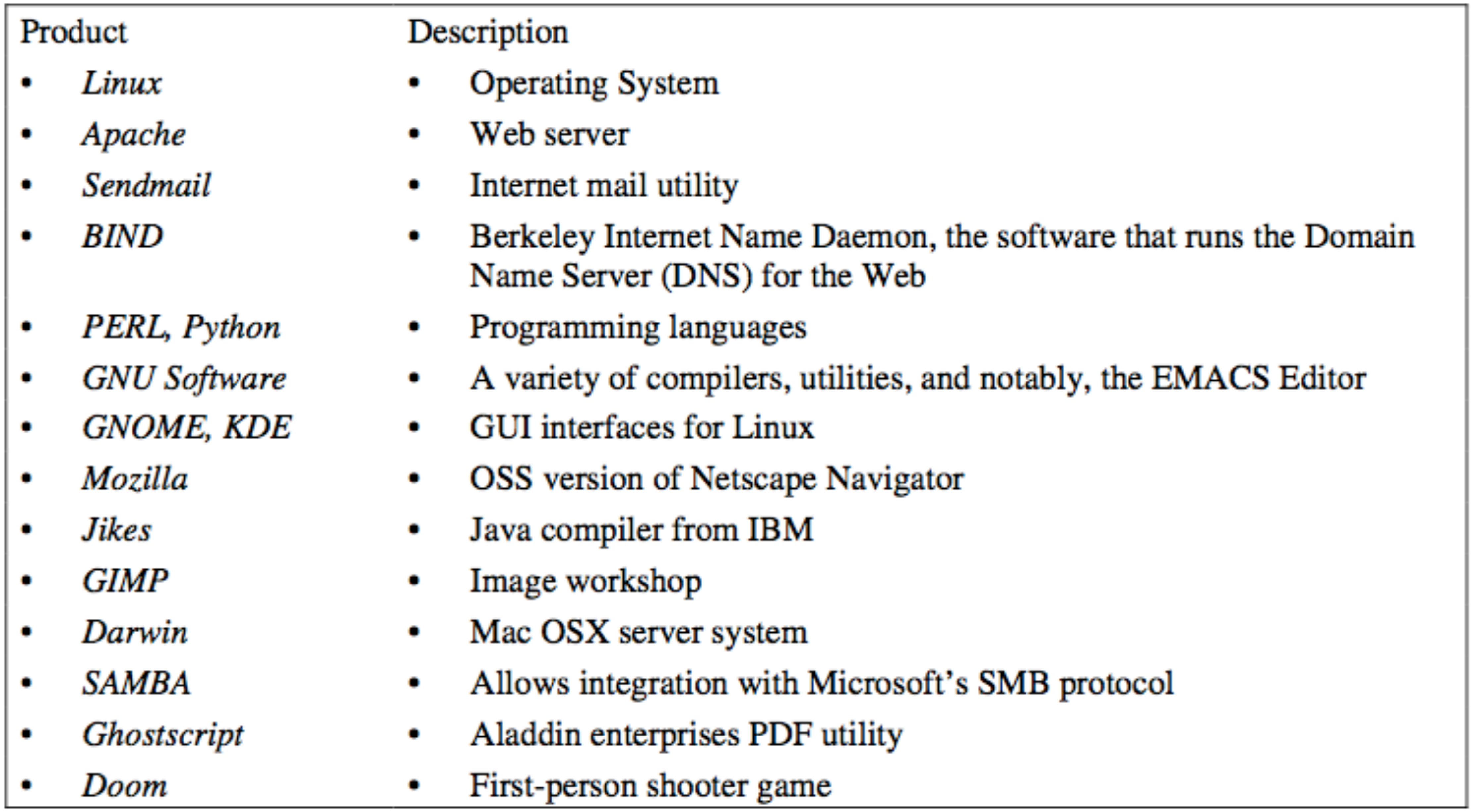
- David Wheeler

Myths and Fears about Open Source

- Is having the ability to view and change source code really important/valuable for many people?
- Is OSS just an anti-Microsoft campaign?
- Is proprietary software fundamentally better supported than OSS?

- David Wheeler

Significant open source contributions



Product	Description
• *Linux*	• Operating System
• *Apache*	• Web server
• *Sendmail*	• Internet mail utility
• *BIND*	• Berkeley Internet Name Daemon, the software that runs the Domain Name Server (DNS) for the Web
• *PERL, Python*	• Programming languages
• *GNU Software*	• A variety of compilers, utilities, and notably, the EMACS Editor
• *GNOME, KDE*	• GUI interfaces for Linux
• *Mozilla*	• OSS version of Netscape Navigator
• *Jikes*	• Java compiler from IBM
• *GIMP*	• Image workshop
• *Darwin*	• Mac OSX server system
• *SAMBA*	• Allows integration with Microsoft's SMB protocol
• *Ghostscript*	• Aladdin enterprises PDF utility
• *Doom*	• First-person shooter game

Reaction of commercial organisations to the open source movement ...

- Duplicate the open source development strategies within the company (to take advantage of effective open source practices)
- Provide a competitive and superior product to that offered openly
- Support the open source efforts by releasing some proprietary software that complements existing open source products
- Some corporations try to develop some of their products via the open source route

Limitations

- Slow development (open source development is like a democracy and commercial software development a dictatorship)
- Signal to noise ratio getting worse (number of poor quality contributions increasing)
- Conflicts between open source developers (similar to the scientific community where researchers indulge in conflicts over research questions)
- The problem of the 'Lowest Hanging Fruit' (involvement mainly in domains that are interesting to the developers)
- Fragmentation and the NIH syndrome (reinventing the wheel, e.g. the many distributions of Linux)
- Cult of personality (open source sounds democratic but is not necessarily e.g. if Linus Torvalds does not like your patch??!!)
- Microsoft
- Lack of technical support (also listed as a strength)

How does an open source project die?

- Burn out of the leader; others cannot carry on
- Inability to acquire a critical mass of users
- Loss of a leading developer
- Forking

A Case Study of Open Source Software Development: The Apache Server

- Roy Fielding *et al.*

Open source developments will have a core of developers who control the code base. This core will be no larger than 10-15 people, and will create approximately 80% or more of the new functionality.

In successful open source developments, a group larger by an order of magnitude than the core will repair defects, and a yet larger group (by another order of magnitude) will report problems.

Open source developments that have a strong core of developers but never achieve large numbers of contributors beyond that core will be able to create new functionality but will fail because of a lack of resources devoted to finding and repairing defects in the released code.

Defect density in open source releases will generally be lower than commercial code that has only been feature-tested, i.e., received a comparable level of testing.

In successful open source developments, the developers will also be users of the software.