

UNIT – I

**The Source of Open Source:** What is Open source, Who creates open source, Who uses Open source, Where do I get open source software, When and how I use open source, Open Source History, Open Source Licenses.

**🡪Introduction to Open Sources**

Open Source is a certification mark owned by the Open Source Initiative   
(OSI). It refers to any program whose source code is made available for use or  
modification as users or other developers. Open source software (OSS) refers to  
software that is developed, tested, or improved through public collaboration and  
distributed with the idea that the must be shared with others, ensuring an open  
future collaboration. ((**OSS**) is computer software with its source code made available and licensed with a license in which the copyright holder provides the rights to study change and distribute the software to anyone and for any purpose. Open-source software is very often developed in a public, collaborative manner.)

**Definition:** Open-source software is computer software whose source code is available under a license (or arrangement such as the public domain) that permits users to study, change, and improve the software, and to redistribute it in modified or modified form. Based on two principles we can call particular software asopensourcesoftware  
**Principle1:** The software source code should be available with license and that license contain permissions they are

1) The user is able to study the code

2) The user able to change the code

3) The may able to improve the code

**Principle2:** The license should not have certain restrictions in terms of

1) Technology

2) Field

3)Hardware  
**Technology:** Here Technology means operating system, in the computer science there are many different operating systems available, here the software must support all kinds of operating systems such as windows, UNIX, Linux and Macos.  
**Field:** Now a day’s computer enter into many fields such as agriculture, medical and Biotechnology fields. Here the software must supports or works in all fields  
**Hardware:** In this context hardware means devices such as Nokia, Samsung and celkon. Here the software should work on or supports all kinds of devices.

**🡪What is the difference between free software and Open source Software?**

Free software, or Freeware, is usually freely distributed software that depend it author’s request, you can redistribute without modifying it.

Most open source software is covered by the GPL (General Public License) which means you can modify it, and redistribute your modifications to it so long as you don’t try to pass it off as the original work.

**🡪What's the difference between open source software and other types of software?**

Some software has source code that cannot be modified by anyone but the person, team, or organization who created it and maintains exclusive control over it. This kind of software is frequently called "proprietary software" or "closed source" software, because its source code is the property of its original authors, who are the only ones legally allowed to copy or modify it.

Open source software is different. Its authors make its source code available to others who would like to view that code, copy it, learn from it, alter it, or share it.

**🡪Advantages of Open Source**

Open source software can have major impact on your entire organization. There are several advantages of using open source software. The following are a list of the advantages of opting for open source software:

**1. Lesser Hardware Costs**

Since Linux and open source solutions are easily portable and compressed, it takes lesser hardware power carry out the same tasks when compared to the hardware power it takes on servers such as Solaris, windows or workstations. With these less hardware power advantage, you can even use cheaper or older hardware and still get the desired results.

**2. High quality software**

Open source software is mostly high quality software. When you use the open source software, the source code is available. Most open source software is well-designed. Open source software can also be efficiently used in coding. These reasons make open source software an ideal choice for organizations.

**3. No Vendor Lock-IN**

IT managers in organizations face different problems when dealing with vendor lock-ins. Lock of portability, expensive license fees and inability to customized software are some of the other disadvantages. Using open source software gives you more freedom and you can effectively address all these disadvantages.

**4. Integrated Management**

By using open source software, you can benefit from integrated management. Open source software use technologies. Such as, common information model (CIM) and web based enterprise management (WBEM). These high technologies enable you to integrate and combine server, application, service and workstation management. This integration would result in efficient administration.

**5. Simple license management**

When you use OSS there is no problem about license. OSS enables you to install it several times and also use it from any location. You will be free from monitoring, tracking or counting license compliance.

**6.Lower software cost**

Using OSS it helps to minimize your expenses. You can save on licensing fees and maintenance fees. The only expenses are the expenditure for documentation, media and support.

**7. Abundant support**

OSS is mostly freely available and can be easily accessed through online communities. These are also many software companies that provide free online help and also varied levels of paid support. Most organizations that create OSS solution also provide maintenance and support.

**8. Scaling and consolidating**

LINUX and OSS can be easily scaled with varied options for clustering, load balancing and open source applications such email and database. You can enable your organization to either scale up and achieve higher growth or consolidate and achieve more with less.

**🡪Disadvantages of using Open Source**

There’s a flip side to everything, and in the case of Open Source software it all boils down to the old saying of “there’s no such thing as a free lunch”. Most of the disadvantages only apply if you’re not somewhat code savvy and willing to get your hands dirty:

1. Mostly used commercial applications

2. Projects can die

3. Support issues.

4. Because there is no requirement to create a commercial product that will sell and generate money, open source software can tend to evolve more in line with developers’ wishes than the needs of the end user.

5. For the same reason, they can be less “user-friendly” and not as easy to use because less attention is paid to developing the user interface.

6. There may also be less support available for when things go wrong – open source software tends to rely on its community of users to respond to and fix problems.

7. Although the open source software itself is mostly free, there may still be some indirect costs involved, such as paying for external support.

8. Although having an open system means that there are many people identifying bugs and fixing them, it also means that malicious users can potentially view it and exploit any vulnerabilities.

**🡪Applications of Open Sources**

Some of the application listed below:

1. Accounting

2. Content Management Systems

3. CRM (Customer Relationship Management)

4. Desktop Environments/ Shell replacements

5. Email Clients

6. Encoding, Conversion & Ripping Tools

7. ERP

8. File sharing & FTP

9. Graphics-Design & Modeling Tools

10. Messengers & Communication Clients

11. Project Management

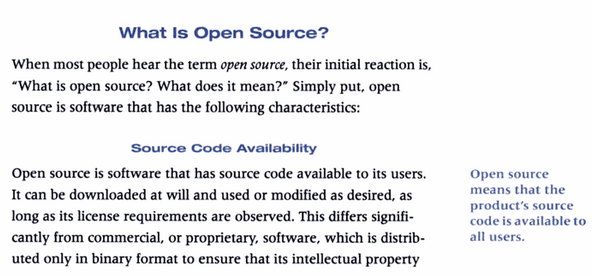
12. Reporting Tools

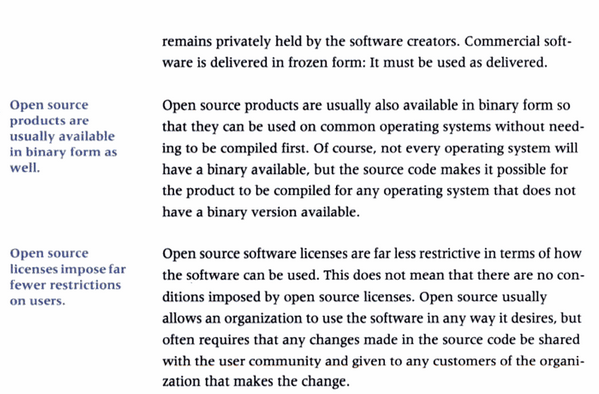
13. RSS

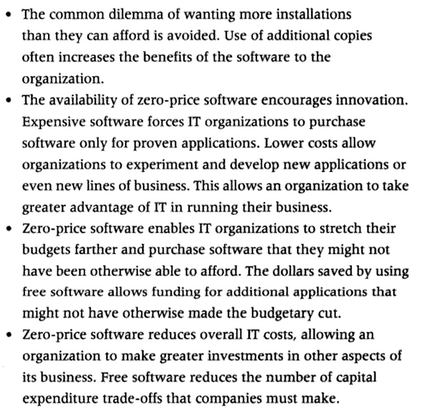
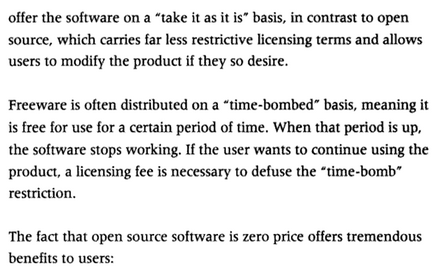
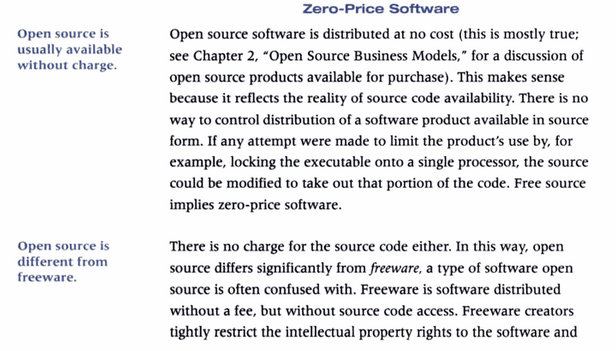
14.WebBrowsers  
**List of commercial open source applications with tools**

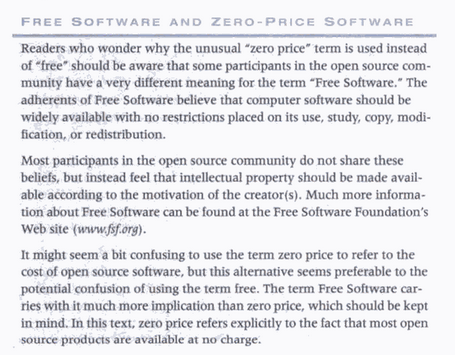
|  |  |  |
| --- | --- | --- |
| **S.NO** | **Application** | **Open Source Tools** |
| 1 | Cloud management | Abiquo |
| 2 | Ecommerce | Avactis |
| 3 | Reporting Tools | Actuate |
| 4 | Enterprise Content Management, web Content Management | Alfresco |
| 5 | Data Backup/Recovery | Bacula |
| 6 | ERP and CRM | Compiere |
| 7 | Office productivity | Lotus Symphony |
| 8 | RDBMS | Ingres Database |
| 9 | Software Development Tools for C , C++ | Sun Studio |
| 10 | Server and Client LINUX distribution | Ubuntu |

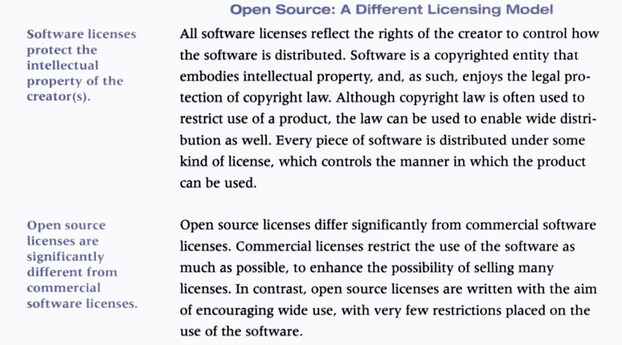
* 1. **What is Open Source**

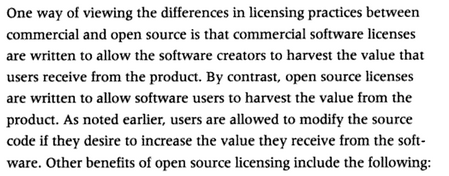
****

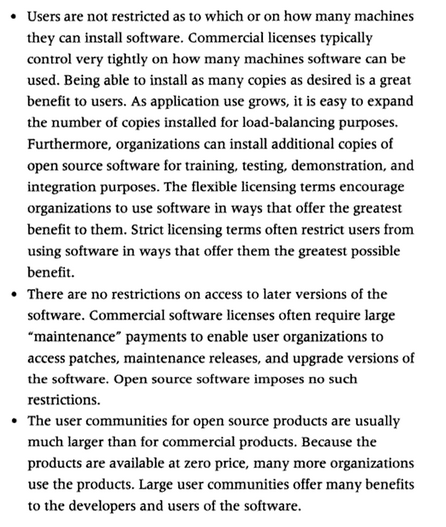
****

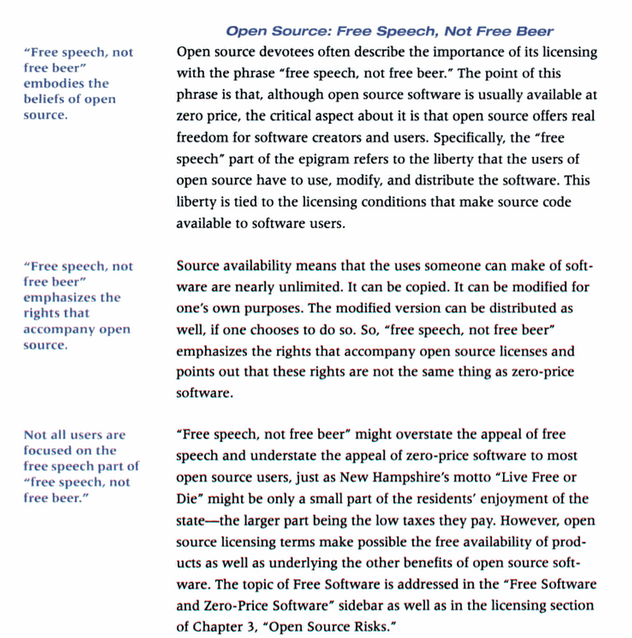
****

****

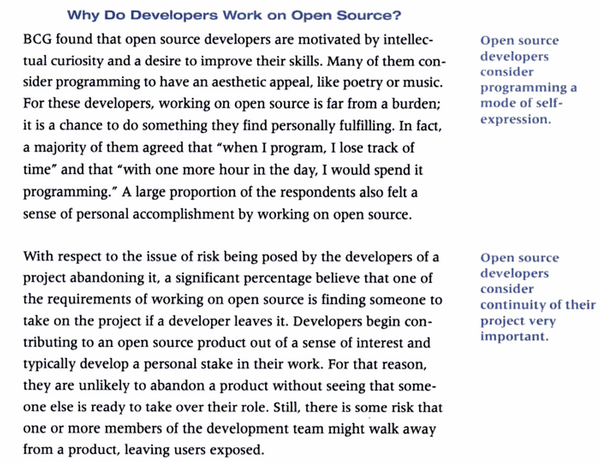
****

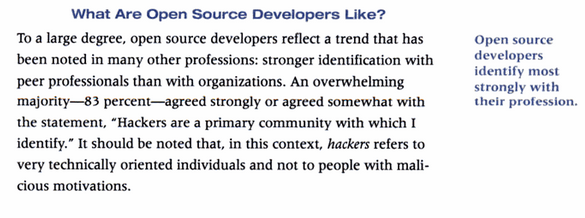
****

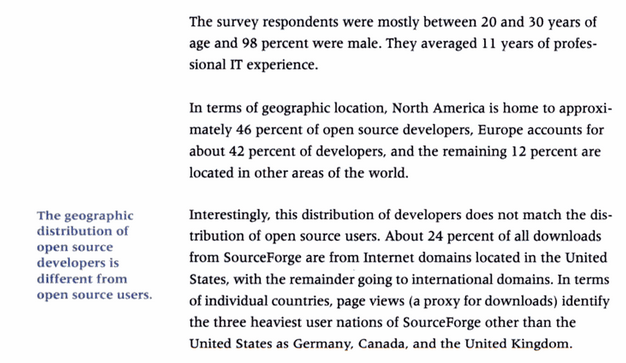
****

****

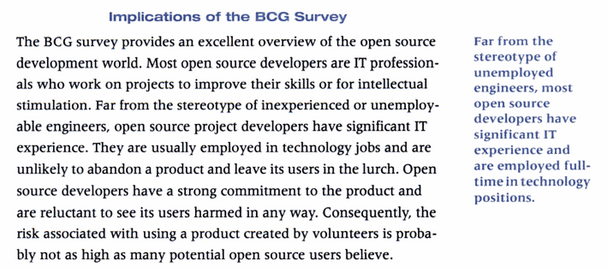
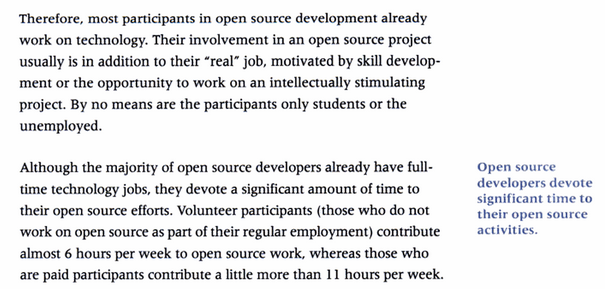
**2. Who creates open source**

****

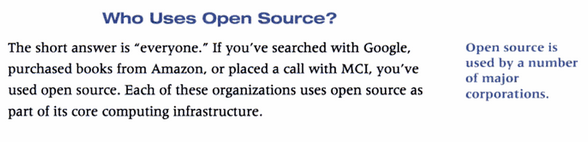
****

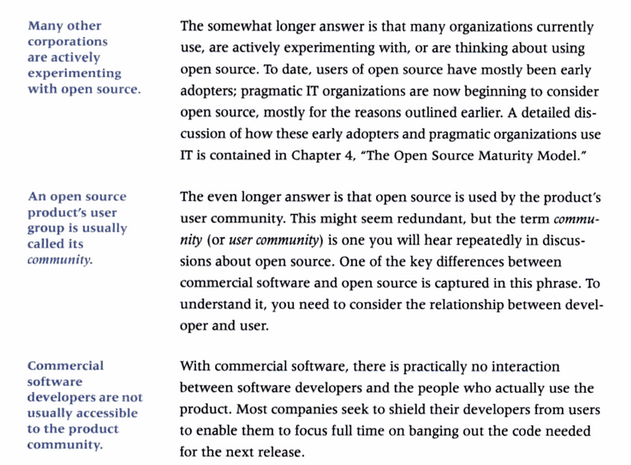
****

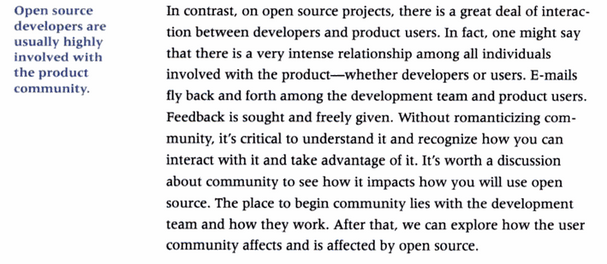
****

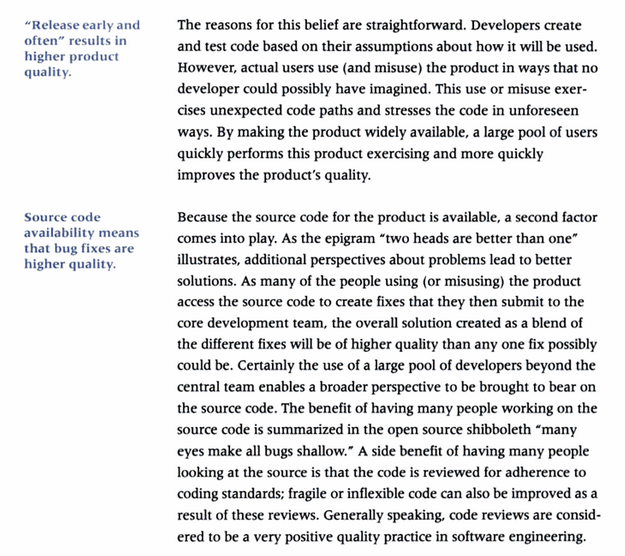
****

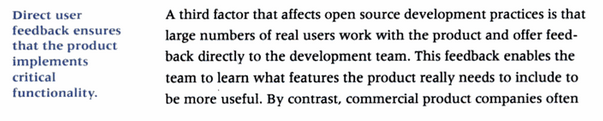
**3. Who use open source**

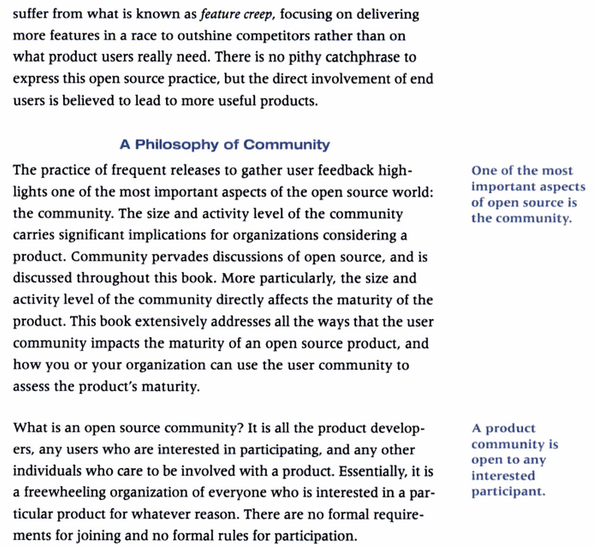
****

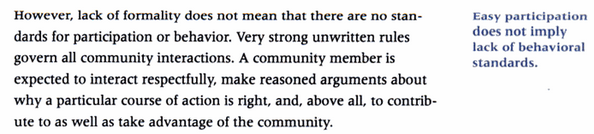
****

****

****

****

****

****

**4. Where do I get open source software?**

Open source software is available from many different places. Individual open source products might have their own web site to make the product available. There are several open source portals, which act as repositories of open source software. Many open source products are available at these portals, making them convenient for locating products via the portal’s search capability. Finally a few open source products are available for sale, typically made available by companies that have bundled the basic open source product along with some useful utilities and possibly an improved installation mechanism.

**\* The most convenient place to get a product is from one of the open source portals.**

The eight most successful open source products ever

**1. Linux**

Linux, hand in hand with GNU software as GNU/Linux, has come a long way since Linus Torvalds announced that he was creating an OS kernel based on Minix back in 1991. **These days, a majority of web servers run Linux, and with Ubuntu** (see below) it is also (finally) starting to make inroads into the desktop market, and maybe it will soon also be strong player in the mobile market with Android (which uses the Linux kernel).

#### 2. Ubuntu

http://farm4.static.flickr.com/3594/3576185198_2c6de1a0cd_o.jpg Launched in 2004, Ubuntu is by far the most popular Linux distribution today, especially on the desktop side. Considering the massive success of Ubuntu in recent years, we thought it was worth its own mention here even though we already mentioned Linux.

#### 3. BSD (Berkeley Software Distribution (various UNIX flavors))

http://farm4.static.flickr.com/3627/3576185278_17331d4814_o.jpgFreeBSD, OpenBSD and NetBSD have been well-respected server OS alternatives for a long time. Derived from Berkeley Unix in the 1990s, we chose to put them into one group here. As an interesting aside, the core for Apple’s Mac OS X is derived from FreeBSD.

#### 4. MySQL

http://farm4.static.flickr.com/3585/3575378753_3657a83c9b_o.jpgMySQL the most widely used database server in the world, used by a huge amount of websites and services (examples include Wikipedia, Facebook and, more modestly, our very own Pingdom.com…). It’s the M in the hugely popular LAMP stack (Linux, Apache, MySQL, PHP).

#### 5. Apache

http://farm4.static.flickr.com/3207/3575378487_9470cc9f8c_o.jpg**:** The Apache HTTP Server has been the most popular web server software in the world since 1996, which is also the year it got started. Apache still has a strong lead, outclassing second runner up IIS in terms of number of deployed websites(according to Netcraft, Apache is currently used by 46% of all websites, while IIS is used by 29%). In 2009 it passed a huge milestone, becoming the first web server to be used by more than 100 million websites.

#### 6.Firefox

http://farm4.static.flickr.com/3297/3576185214_18f4b3e781_o.jpg Mozilla’s crowning achievement so far, the Firefox web browser has become a mega success. Firefox 1.0 was launched in 2004 and the browser has since then taken away a huge chunk of the browser market from the previously dominant Internet Explorer, and is arguably the reason that Microsoft started to put more effort into updating IE with new versions. Although Firefox is still number two overall, it has become the dominant browser among the more “techie” crowd (this blog, for example, gets 59% of its visits from Firefox and just 18% from IE).

#### 7.WordPress

http://farm3.static.flickr.com/2465/3576185028_6c02df8eb9_o.jpg Since its launch in 2004 as a fork of the b2 blog software, WordPress has become a dominant and hugely popular blog platform. In a survey we made back in January 27% of the top 100 blogs ran on WordPress. If you also counted WordPress.com, Automattic’s hosted WordPress service, that number rose to 32%, more than any other blog software. Since then there have also been some changes, such as the nine Wired blogs in the top 100 switching from Typepad to WordPress, so that percentage is likely significantly higher now (all else unchanged, it would be 41%).

#### 8.BIND

http://farm3.static.flickr.com/2438/3576185170_e10d544c10_o.jpg (**The Berkeley Internet Name Domain Server**) is the most widely used DNS server software on the Internet. The first version of BIND goes all the way back to the early 1980s and has been the main DNS server on UNIX systems ever since. It can justly be called the world’s de facto standard DNS server.

**Top Portal Software Products**

**eXo Platform**

Open-source social-collaboration software designed for enterprises - based on standards, extensible and has an amazing design.

## Bitrix24

High-end solution designed for effective collaboration, communication, social networking, and workflow and knowledge management.

**Aura Portal**

BPM software for the creation, modeling and execution of Business Process Workflows without the need for IT programming.

**Ready Portal**

A lightweight high performance portal engine focused on ease of implementation and content publication

**Clinked**

Online client portal, project management, teamwork and collaboration platform that allows people to interact wherever they are.

**Suite Dash**

Fully branded: Client Portal, Project Management, Invoicing, File Sharing, CRM, IM, Messaging & more

**Directors Desk**

Secure web-based portal to compile board books, store documents and distribute them; as well as conduct votes, surveys and discussions.

**Online Appointment Manager**

Online appointment scheduling software for medical, dental, accounting, home repair and other professional practices.

**Billing System**

A web-enabled automated billing system to manage billing, payments and customer information. No start-up costs.

[For more refer[http://www.capterra.com/portal-software ]}](http://www.capterra.com/portal-software)]

**1.Individual Open Source Product Web Sites**

Some very well established open source products have their own web sites that act as the main distribution mechanism for the software. The web sites act as gathering points for developers and the user community to interact.

They often have forums for discussions and questions among the community. News about the product will be available as well. These web sites are the electronic equipment of an old-fashioned country store in which transactions, friendships, information swapping and gossip all take place. The sites themselves can easily found via a Google Search on the product name.

**\* Some open source products have their own web site for distribution.**

**2. Open source Portals**

Open source portals offer a centralized location for open source products. The portals host open source projects, offering a number of services that make starting and maintaining an open source project much easier.

Some of the Open source portals are listed below  
1. Apache HTTP Server [ http:// httpd.apache.org/] (Web Server)

2. Blender [http://www.blender.org] (3D graphics and animation package)

3. GNU Compiler Collection [http://www.gnu.org/software/gcc/gcc.html] (GCC, a suite of compilation tools for C, C++, etc)

4. KDE [http://www.kde.org/] (Linux desktop environment)

5. Moodle [http://www.moodle.org/] (virtual learning system)

6. Firefox [http://www.mozilla.com/en-US/firefox/] (web browser based on Mozilla)

7. MySQL [http://www.mysql.com/] (database)

8. OpenOffice.org [http://www.openoffice.org/] (office suite, including word processor, spreadsheet, and presentation software)

9. PHP [http://www.php.net/] (web development)

10. Perl [http://www.perl.org/] (programming/scripting language)

11. PostgreSQL [http://www.postgresql.org/] (database)

12. Python [http://www.python.org/] (programming/scripting language)

13. Samba [http://www.samba.org/] (file and print server)

14.Zope [http://www.zope.org/] (web application server)

\* **For more refer the URL http://oss-watch.ac.uk/resources/softwareexamples**

**https://www.neteasy.us/technology/open-source/examples-of-open-source-software**

**3. Commercial Distributions**

A few source products are available for sale. The commercial product is usually offered along with other product-oriented services, like technical support or training. Even in the companies that offer a commercial version of an open source product, however, usually the product is available at zero prices as well. The version sold is merely made available in a more convenient format (Ex: on a CD) or as part of a larger product offering that bundles services along with the software.

\***Some open source products are available on commercial distributions**

**4. The challenge of Anonymous Distribution**

One of the most interesting , yet frustrating, aspects of open source is that not only is it available at zero price, but it is available anonymously. You don’t have to identify yourself to download the product: no forms to fill in, no credit card information (unless the product is purchased), no nothing.

\***Open source products are available for anonymous download.**

**🡪5. When and how I use open source**

These are intertwined questions. The right time to use open source is when both you and the product are ready. The practices you (and the IT industry) have used over the past 40 years won’t work with open source products. A whole new method of selecting and evaluating products is required to succeed with open source.

\***Using Open source requires new working practices.**

**6. History of Open Source**

1960's and 1970's: software was largely provided by computer companies and freely shared.

● **1969:** UNIX developed at AT&T Bell Labs.

● **1969:** ARPANET created.

● **1970's:** AT&T provides CS departments with UNIX source code and encouraged modifications (could not sell due to 1974 antitrust findings).

● **1975:** Microsoft founded, first product is BASIC for MITS Altair (an early microcomputer)

* **1976:** Bill Gates accuses hobbyists of stealing his software, thus preventing “good software from being written” (of course he paid nothing for BASIC).

● **1976:** US amends copyright law, no longer requires explicit registration, etc.

● **1980:** US copyright law amended to cover software.

● **1980:** Microsoft launches UNIX-clone XENIX for 16-bit microprocessors.

● **1981:** Bill Gates makes deal to buy DOS for $50k (without mentioning pending IBM PC deal).

● **1981:** Launch of the IBM PC with MS-DOS.

● **1980's**: rise of proprietary software, companies quit sharing code and allowing modifications, and start

charging lots of money for software.

● **1980's:** IBM is #1 computer company with DEC #2 (DEC strongly associated with ARPANET but DEC anti-UNIX despite UNIX being developed on PDPs).

**● 1982:** AT&T divestiture (breakup) allows UNIX to be sold and the “UNIX wars” begin.

● **1982:** Sun Microsystems born: UNIX workstations.

● **1982:** Larry Wall creates patch utility for UNIX, enables distributed, collaborative development.

● **1983:** DARPA-funded BSD UNIX TCP/IP released.

● **Early 1980s:** ARPANET and UNIX hacker communities begin to converge on UNIX and C.

● **1984:** MIT hacker Richard Stallman starts GNU project to promote “free software.”

● **1984:** X Window project begun at MIT to develop GUI for UNIX, supported by most UNIX vendors.

● **1985:** POSIX starts to standardize UNIX.

● **Mid 1980's:** DEC Vaxes running UNIX begin to take over ARPANET/NSFNET infrastructure duties.

● 1985: NSFNET created (ARPANET successor, and start of the civilian Internet).

● 1985: Intel releases i386 chip, first 8086 CPU with flat address space that could support UNIX well.

● 1987: Larry Wall releases PERL, FOSS scripting language, for UNIX.

● 1987: first version of GNU C compiler released, and GNU development toolset largely complete.

● 1990: Berkely begins effort to remove all proprietary AT&T code from BSD UNIX.

● 1991: Finnish CS grad student Linus Torvalds announces Linux project on USENET, with goal of

producing a UNIX-like OS for Pcs (like MINIX), due to high cost of commercial UNIXes like Sun Solaris.

● 1992: AT&T sues Berkely over BSD UNIX, largely halting UNIX development at Berkely.

● 1994: AT&T/Berkely lawsuit settlement allows BSD UNIX to be released, free of AT&T code

● Mid 1990's: liberal BSD license allows companies (including Microsoft) to use BSD code in their

products, leading to Berkely sockets becoming the de facto network programming API.

● Mid 1990's: Linux with GNU tools becomes the primary UNIX-like OS on PCs.

● 1995: Red Hat Software is founded, one of the first commercial Linux distributions.

● 1996: KDE desktop project started, but relied on non-free Trolltech Qt toolkit.

● 1997: FOSS projects GTK toolkit and GNOME desktop are started over concerns about Qt.

● 1997: Eric Raymond publishes The Cathedral and the Bazaar, arguing that open source development

models produce better code, which he summarized with what he termed “Linus Law”: “with enough eyes, all bugs are shallow.”

● 1998: Trolltech re-licenses Qt under “free” license.

● 1998: Netscape decides to open source its primary product, Netscape Navigator browser.

● 2000's: Linux is increasingly widely used in corporate environments, particularly for servers.

● 2000's: Linux development is supported by numerous corporations that view it as commoditizing operating systems, reducing their reliance on Microsoft and eliminating the “Microsoft tax.”

● 2000's: open source software projects involving Internet-based collaborative programming become common, and commoditize many types of software.

● 2000's: virtually all s**upercomputers** run Linux.

● 2003: SCO sues IBM over claimed “UNIX IP” illegally transferred to Linux.

● 2007: SCO loses in court against Novell over ownership of UNIX IP, effectively ending IBM suit (plus repeatedly fails to prove UNIX is in Linux).

● 2007: Sun finally re-licenses Java under free license (but see below).

● 2007: Google releases Android OS based on Linux.

● 2010: Oracle sues Google over Java-related technology patents in Android!

● 2011: Android becomes the most widely sold OS on smart phones.

● 2011: Barnes & Noble makes Microsoft Android patent claims public (showing them to be trivial and

possibly invalid patents), and initiates claims of anti-competitive behavior against Microsoft.

**7. Open Source Licenses**

**1. Proprietary Software License:**

* Fairly standard terms
* Source code availability
  + Source code not provided - trying to figure out inner workings of software through reverse engineering or decompiling of operating mode is forbidden; OR
  + Source code provided - may or may not include permission to create modifications and enhancements

**Proprietary Software License terms – Licensees:**

* Restrictions on dissemination. Licensee and users strictly defined. Licensee has no right to share with those not defined as licensee users in license;
* Licensor indemnifies licensees against third party infringement claims;
* Often, have to sign a new license each time new licensee obtains the code.

**Proprietary Software License terms – Warranty and Support:**

* Warranties provided:
  + Defects in media and existence of viruses, Trojan Horses, backdoors, etc;
  + Can negotiate for warranties re: meet specifications in product documentation
* Maintenance and support terms included (although may be in separate document).

**2. Open Source Software License – Licensees:**

* Original software owner or developer chooses to limit the rights that he asserts over licensees
* Licensees, subject to license terms, can:
  + make and distribute copies of software;
  + build upon software to create modifications or other works.

**Open Source Software Licenses - Source Code:**

* Source code to original product always provided;
* Licensee can modify or enhance source code (create “derivative works”) or include source code with other license types (create “larger works”);
* Licensee may be required to share modifications with the world (in source and/or binary form), but not necessarily;
* Licensee may be prohibited from charging royalties for derivative and larger works, but not necessarily.

**Open Source Software License – Warranties and Support:**

* Generally, software provided “AS-IS” with no warranties, warranties excluded;
* No indemnification;
* No maintenance or support.

**Popular Open Source Licenses**

The following OSI (Open Source Initiative)-approved licenses are popular, widely used, or have strong communities:

Apache License 2.0

BSD 3-Clause "New" or "Revised" license

BSD 2-Clause "Simplified" or "FreeBSD" license

GNU General Public License (GPL)

GNU Library or "Lesser" General Public License (LGPL)

MIT license

Mozilla Public License 2.0

Common Development and Distribution License

Eclipse Public License

**[For More Visits https://opensource.org/licenses/alphabetical ]**