Business models in FLOSS-based companies1

Carlo Daffara1

1 Conecta srl, research division

cdaffara@conecta.it

Abstract. The sustainability of Free/Libre/Open Source Software (FLOSS) has

recently become an important aspect of the overall open source phenomenon,

and as such has received a significant research attention. Since 1999, several

articles addressed the role of licenses, communities and business models in

FLOSS-based companies, but mostly from a theoretical point of view. In this

article, we present the result of a detailed analysis of 80 FLOSS-based

companies and business models, using publicly available data on service offers,

mailing list archives, online articles and when available public financial data.

The result is a simplified and effective classification that can be used as a guide

to companies willing to offer services in the same area, and an analysis on how

the community process is influenced by the choice of licensing and business

model.

1 Introduction

Business models are abstract, conceptual models that represents the business and

money earning logic of a company in a structured way [1,2]. The traditional business

models for software are challenged by the redistributability of open source software,

and the fact that the user/customer does have several additional rights not usually

a

Business models in FLOSS-based companies1

Carlo Daffara1

1 Conecta srl, research division

cdaffara@conecta.it

Abstract. The sustainability of Free/Libre/Open Source Software (FLOSS) has

recently become an important aspect of the overall open source phenomenon,

and as such has received a significant research attention. Since 1999, several

articles addressed the role of licenses, communities and business models in

FLOSS-based companies, but mostly from a theoretical point of view. In this

article, we present the result of a detailed analysis of 80 FLOSS-based

companies and business models, using publicly available data on service offers,

mailing list archives, online articles and when available public financial data.

The result is a simplified and effective classification that can be used as a guide

to companies willing to offer services in the same area, and an analysis on how

the community process is influenced by the choice of licensing and business

model.

1 Introduction

Business models are abstract, conceptual models that represents the business and

money earning logic of a company in a structured way [1,2]. The traditional business

models for software are challenged by the redistributability of open source software,

and the fact that the user/customer does have several additional rights not usually

a

**HISTORY OF LINUX**

The history of Linux began in 1991 with the commencement of a personal project by [Finnish](https://en.wikipedia.org/wiki/Finland) student [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds) to create a new free operating system kernel. Since then, the resulting [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) has been marked by constant growth throughout its history. Since the initial release of its [source code](https://en.wikipedia.org/wiki/Source_code) in 1991, it has grown from a small number of [C](https://en.wikipedia.org/wiki/C_Programming_Language) files under a license prohibiting commercial distribution to the 4.15 version in 2018 with more than 23.3 million lines of source code without comments under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) v2

In 1983, [Richard Stallman](https://en.wikipedia.org/wiki/Richard_Stallman) started the [GNU project](https://en.wikipedia.org/wiki/GNU_project) with the goal of creating a free UNIX-like operating system. As part of this work, he wrote the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL). By the early 1990s, there was almost enough available software to create a full operating system. However, the GNU kernel, called [Hurd](https://en.wikipedia.org/wiki/GNU_Hurd), failed to attract enough development effort, leaving GNU incomplete

In 1985, [Intel](https://en.wikipedia.org/wiki/Intel) released the [80386](https://en.wikipedia.org/wiki/80386), the first [x86 microprocessor](https://en.wikipedia.org/wiki/X86) with a [32-bit](https://en.wikipedia.org/wiki/32-bit) [instruction set](https://en.wikipedia.org/wiki/Instruction_set) and a [memory management unit](https://en.wikipedia.org/wiki/Memory_management_unit) with [paging](https://en.wikipedia.org/wiki/Paging).

In 1986, Maurice J. Bach, of AT&T Bell Labs, published [The Design of the UNIX Operating System](https://en.wikipedia.org/wiki/List_of_important_publications_in_computer_science#Operating_systems). This definitive description principally covered the [System V Release 2](https://en.wikipedia.org/wiki/System_V#SVR2) kernel, with some new features from [Release 3](https://en.wikipedia.org/wiki/System_V#SVR3) and BSD.

In 1987, [MINIX](https://en.wikipedia.org/wiki/MINIX), a Unix-like system intended for academic use, was released by [Andrew S. Tanenbaum](https://en.wikipedia.org/wiki/Andrew_S._Tanenbaum) to exemplify the principles conveyed in his [textbook](https://en.wikipedia.org/wiki/Textbook), [Operating Systems: Design and Implementation](https://en.wikipedia.org/wiki/Operating_Systems:_Design_and_Implementation). While source code for the system was available, modification and redistribution were restricted. In addition, MINIX's [16-bit](https://en.wikipedia.org/wiki/16-bit) design was not well adapted to the 32-bit features of the increasingly cheap and popular Intel 386 architecture for personal computers. In the early nineties a commercial UNIX operating system for Intel 386 PCs was too expensive for private users.

These factors and the lack of a widely adopted, [free](https://en.wikipedia.org/wiki/Free_software) kernel provided the impetus for Torvalds' starting his project. He has stated that if either the GNU Hurd or [386BSD](https://en.wikipedia.org/wiki/386BSD) kernels had been available at the time, he likely would not have written his own.

In 1991, while studying [computer science](https://en.wikipedia.org/wiki/Computer_science) at [University of Helsinki](https://en.wikipedia.org/wiki/University_of_Helsinki), Linus Torvalds began a project that later became the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel). He wrote the program specifically for the hardware he was using and independent of an operating system because he wanted to use the functions of his new PC with an 80386 processor. Development was done on MINIX using the [GNU C Compiler](https://en.wikipedia.org/wiki/GNU_Compiler_Collection).

**THE LINUX PHILOSOPHY**

Linux has a philosophy that underlies the entire operating system and which strongly affects the way in which administrators — and advanced users — interact with it. This philosophy is very similar to that of its Unix parenthood. It is called “The Linux Way.”

When Unix was being developed in the late 1960’s and early 1970’s, the developers were intent upon building an operating system that was significantly different from the operating systems that preceded. The philosophy of Unix was markedly different from that of other operating systems. And the Linux philosophy is quite naturally derived directly from the Unix philosophy.

The creators of the Unix operating system started with a radical concept: They assumed that the user of their operating system would be computer literate from the start. The entire Unix philosophy revolves around the idea that the user knows what he is doing.”

“Unix was not designed to stop its users from doing stupid things, as that would also stop them from doing clever things.”

Linux treats everyone equally and allows everyone the maximum amount of power. That is egalitarian. Other operating systems are elitist and exclusive because they withhold or hide their power behind an inflexible Graphical User Interface that allows one to do only what the developers think we should be allowed to do.

In contrast,  the philosophy of Windows is very similar to that of Digital Equipment Corporation’s (DEC) VMS operating system. This amounted to “shield the users from everything that might get them into trouble.

**LINUX COMMUNITY**

The Linux kernel, at over 6 million lines of code and well over 1000 active contributors, is one of the largest and most active free software projects in existence. Since its humble beginning in 1991, this kernel has evolved into a best-of-breed operating system component which runs on pocket-sized digital music players, desktop PCs, the largest supercomputers in existence, and all types of systems in between. It is a robust, efficient, and scalable solution for almost any situation.

With the growth of Linux has come an increase in the number of developers (and companies) wishing to participate in its development. Hardware vendors want to ensure that Linux supports their products well, making those products attractive to Linux users. Embedded systems vendors, who use Linux as a component in an integrated product, want Linux to be as capable and well-suited to the task at hand as possible. Distributors and other software vendors who base their products on Linux have a clear interest in the capabilities, performance, and reliability of the Linux kernel. And end users, too, will often wish to change Linux to make it better suit their needs.

One of the most compelling features of Linux is that it is accessible to these developers; anybody with the requisite skills can improve Linux and influence the direction of its development. Proprietary products cannot offer this kind of openness, which is a characteristic of the free software process. But, if anything, the kernel is even more open than most other free software projects. A typical three-month kernel development cycle can involve over 1000 developers working for more than 100 different companies (or for no company at all).

Working with the kernel development community is not especially hard. But, that notwithstanding, many potential contributors have experienced difficulties when trying to do kernel work. The kernel community has evolved its own distinct ways of operating which allow it to function smoothly (and produce a high-quality product) in an environment where thousands of lines of code are being changed every day. So it is not surprising that Linux kernel development process differs greatly from proprietary development methods.

The kernel's development process may come across as strange and intimidating to new developers, but there are good reasons and solid experience behind it. A developer who does not understand the kernel community's ways (or, worse, who tries to flout or circumvent them) will have a frustrating experience in store. The development community, while being helpful to those who are trying to learn, has little time for those who will not listen or who do not care about the development process.

It is hoped that those who read this document will be able to avoid that frustrating experience. There is a lot of material here, but the effort involved in reading it will be repaid in short order. The development community is always in need of developers who will help to make the kernel better.

**LINUX DISTRIBUTIONS**

A Linux distribution (often abbreviated as distro) is an [operating system](https://en.wikipedia.org/wiki/Operating_system) made from a software collection, which is based upon the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and, often, a [package management system](https://en.wikipedia.org/wiki/Package_management_system). Linux users usually obtain their operating system by downloading one of the Linux distributions, which are available for a wide variety of systems ranging from [embedded devices](https://en.wikipedia.org/wiki/Embedded_device) (for example, [OpenWrt](https://en.wikipedia.org/wiki/OpenWrt)) and [personal computers](https://en.wikipedia.org/wiki/Personal_computer) (for example, [Linux Mint](https://en.wikipedia.org/wiki/Linux_Mint)) to powerful [supercomputers](https://en.wikipedia.org/wiki/Supercomputer) (for example, [Rocks Cluster Distribution](https://en.wikipedia.org/wiki/Rocks_Cluster_Distribution)).

A typical Linux distribution comprises a Linux kernel, [GNU](https://en.wikipedia.org/wiki/GNU) tools and libraries, additional software, documentation, a [window system](https://en.wikipedia.org/wiki/Window_system) (the most common being the [X Window System](https://en.wikipedia.org/wiki/X_Window_System)), a [window manager](https://en.wikipedia.org/wiki/Window_manager), and a [desktop environment](https://en.wikipedia.org/wiki/Desktop_environment). Most of the included software is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) made available both as compiled binaries and in [source code](https://en.wikipedia.org/wiki/Source_code) form, allowing modifications to the original software. Usually, Linux distributions optionally include some [proprietary software](https://en.wikipedia.org/wiki/Proprietary_software) that may not be available in source code form, such as [binary blobs](https://en.wikipedia.org/wiki/Binary_blob) required for some [device drivers](https://en.wikipedia.org/wiki/Device_driver).

Widely used distributions

* [**Debian**](https://en.wikipedia.org/wiki/Debian), a non-commercial distribution and one of the earliest, maintained by a volunteer developer community with a strong commitment to free software principles and democratic project management
  + [Knoppix](https://en.wikipedia.org/wiki/Knoppix), the first [Live CD](https://en.wikipedia.org/wiki/Live_CD) distribution to run completely from [removable media](https://en.wikipedia.org/wiki/Removable_media) without installation to a hard disk, derived from Debian
  + [Linux Mint Debian Edition](https://en.wikipedia.org/wiki/Linux_Mint_Debian_Edition) (LMDE) uses [Debian packages](https://en.wikipedia.org/wiki/Debian#Package_management) directly (rather than Ubuntu's)
  + [**Ubuntu**](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)), a desktop and server distribution derived from Debian, maintained by British company [Canonical Ltd.](https://en.wikipedia.org/wiki/Canonical_Ltd.)
    - [Kubuntu](https://en.wikipedia.org/wiki/Kubuntu), the [KDE](https://en.wikipedia.org/wiki/KDE) version of Ubuntu
    - [Linux Mint](https://en.wikipedia.org/wiki/Linux_Mint), a distribution based on and compatible with Ubuntu. Supports multiple desktop environments, among others GNOME Shell [fork](https://en.wikipedia.org/wiki/Fork_(software_development)) [Cinnamon](https://en.wikipedia.org/wiki/Cinnamon_(user_interface)) and GNOME 2 fork [MATE](https://en.wikipedia.org/wiki/MATE_(desktop_environment)).
    - [Trisquel](https://en.wikipedia.org/wiki/Trisquel), an Ubuntu-based distribution based on Linux-libre kernel composed entirely of free software
    - [Elementary OS](https://en.wikipedia.org/wiki/Elementary_OS), an Ubuntu-based distribution with strong focus on the visual experience without sacrificing performance.
* [**Fedora**](https://en.wikipedia.org/wiki/Fedora_(operating_system)), a community distribution sponsored by American company [Red Hat](https://en.wikipedia.org/wiki/Red_Hat) and the successor to the company's previous offering, [Red Hat Linux](https://en.wikipedia.org/wiki/Red_Hat_Linux). It aims to be a technology testbed for Red Hat's commercial Linux offering, where new [open source software](https://en.wikipedia.org/wiki/Open_source_software) is prototyped, developed, and tested in a communal setting before maturing into Red Hat Enterprise Linux.
  + [Red Hat Enterprise Linux](https://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux) (RHEL), a derivative of Fedora, maintained and commercially supported by Red Hat. It seeks to provide tested, secure, and stable Linux server and workstation support to businesses.
    - [CentOS](https://en.wikipedia.org/wiki/CentOS), a distribution derived from the same sources used by [Red Hat](https://en.wikipedia.org/wiki/Red_Hat), maintained by a dedicated volunteer community of developers with both 100% Red Hat-compatible versions and an upgraded version that is not always 100% upstream compatible.
    - [Oracle Linux](https://en.wikipedia.org/wiki/Oracle_Linux), which is a derivative of Red Hat Enterprise Linux, maintained and commercially supported by [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation)
    - [Scientific Linux](https://en.wikipedia.org/wiki/Scientific_Linux), a distribution derived from the same sources used by Red Hat, maintained by [Fermilab](https://en.wikipedia.org/wiki/Fermilab)

[Mandriva Linux](https://en.wikipedia.org/wiki/Mandriva_Linux) was a Red Hat derivative popular in several European countries and Brazil, backed by the French company of the same name. After the company went bankrupt, it was superseded by [OpenMandriva Lx](https://en.wikipedia.org/wiki/OpenMandriva_Lx), although a number of derivatives now have a larger user base.

[Mageia](https://en.wikipedia.org/wiki/Mageia), a community fork of Mandriva Linux created in 2010

[PCLinuxOS](https://en.wikipedia.org/wiki/PCLinuxOS), a derivative of Mandriva, which grew from a group of packages into a community-spawned desktop distribution

[**ROSA** Linux](https://en.wikipedia.org/wiki/ROSA_Linux), another former derivative of Mandriva, now developed independently

[**openSUSE**](https://en.wikipedia.org/wiki/OpenSUSE), a community distribution mainly sponsored by German company [SUSE](https://en.wikipedia.org/wiki/SUSE).

[SUSE Linux Enterprise](https://en.wikipedia.org/wiki/SUSE_Linux), derived from openSUSE, maintained and commercially supported by SUSE

[Arch Linux](https://en.wikipedia.org/wiki/Arch_Linux), a [rolling release](https://en.wikipedia.org/wiki/Rolling_release) distribution targeted at experienced Linux users and maintained by a volunteer community, offers official binary packages and a wide range of unofficial user-submitted source packages. Packages are usually defined by a single [PKGBUILD](https://en.wikipedia.org/wiki/PKGBUILD) text file.

[Manjaro Linux](https://en.wikipedia.org/wiki/Manjaro_Linux), a derivative of Arch Linux that includes a graphical installer and other ease-of-use features for less experienced Linux users. Rolling release packages from Arch repositories are held for further testing to achieve increased stability, and packages identified as addressing security issues of critical or high severity are "fast-tracked" to the stable branch.

[Gentoo](https://en.wikipedia.org/wiki/Gentoo_Linux), a distribution targeted at [power users](https://en.wikipedia.org/wiki/Power_user), known for its [FreeBSD Ports](https://en.wikipedia.org/wiki/FreeBSD_Ports)-like automated system for compiling applications from source code

[Chrome OS](https://en.wikipedia.org/wiki/Chrome_OS), Google's commercial operating system (using Gentoo and its Portage) that primarily runs [web applications](https://en.wikipedia.org/wiki/Web_application)

[Chromium OS](https://en.wikipedia.org/wiki/Chromium_OS), the fully open-source version of Chrome OS

[Slackware](https://en.wikipedia.org/wiki/Slackware), created in 1993, one of the first Linux distributions and among the earliest still maintained, committed to remain highly [Unix-like](https://en.wikipedia.org/wiki/Unix-like) and easily modifiable by [end users](https://en.wikipedia.org/wiki/End_user).