

# A CONCISE INTRODUCTION

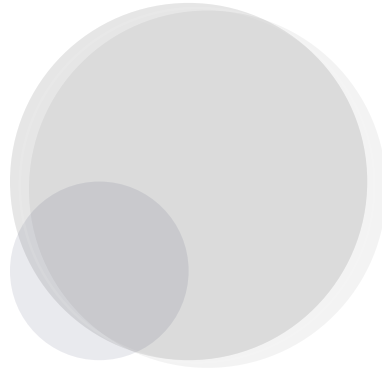
## FREE AND OPEN SOURCE SOFTWARE

by Andrew Updegrave

---

2019  
AUGUST





# A CONCISE INTRODUCTION TO FREE AND OPEN SOURCE SOFTWARE

2019  
AUGUST

ABOUT



[OpenForum Europe \(OFE\)](#) is a not-for-profit, independent European based think tank which focuses on openness within the IT sector. We draw our support not only from some of the most influential global industry players, but most importantly from across European SMEs and consumer organisations and the open community. OFE also hosts a global network of [OpenForum Academy Fellows](#), each contributing significant innovative thought leadership on core topics. Views expressed by OFE do not necessarily reflect those held by all its supporters.



**THIS WORK IS LICENSED UNDER A CREATIVE COMMONS  
ATTRIBUTION-SHAREALIKE 4.0 INTERNATIONAL LICENSE.**

# A CONCISE INTRODUCTION

## FREE AND OPEN SOURCE SOFTWARE

by Andrew Updegrave<sup>1</sup>

### INTRODUCTION

In the early days of information technology (IT), computers were delivered with operating systems and basic application software already installed, without additional cost, and in editable (source code) form. But as software emerged as a stand-alone product, the independent software vendors (ISVs) that were launched to take advantage of this commercial opportunity no longer delivered source code, in order to prevent competitors from gaining access to their trade secrets. The practice also had the (intended) result that computer users became dependent on their ISVs for support and upgrades.

Due to the increasingly substantial investments computer users made in application software, they also became “locked in” to their hardware, because of the high cost of abandoning, or reconfiguring, their existing application software to run on the proprietary operating system of a new vendor. In response, a movement in support of distributing human-readable source code as well as the legal right to modify, share and distribute that code, together with the usual, machine-readable object code, emerged in the mid-1980s. The early proponents of such “free software” regarded the right to share source code as an essential freedom and created licenses – notably, the GNU General

---

<sup>1</sup> Andrew Updegrave is a founding partner of Gesmer Updegrave LLP, a Boston-based technology law firm. He has a broad range of experience in representing both mature and emerging high technology companies of all types in all aspects of their legal affairs. Since 1988, he has also represented and helped structure more than 180 worldwide standard setting, open source, research and development, promotional and advocacy consortia, including some of the largest standard setting and open source organizations in the world. He spends a significant part of his time giving strategic advice to clients of the firm.

Public Licenses – that required vendors to give back their own innovations to the project community. Those who espoused this view are usually referred to as being part of the “free software movement.” A later faction focused only on the practical advantages of freely sharable code, which they called “open source software” (OSS), leading to adherents of that group becoming known as the “open source movement.”<sup>2</sup>

Concurrently, the Internet enabled a highly distributed model of software development to become possible, based upon voluntary code contributions and globally collaborative efforts. The combined force of these developments resulted in the rapid proliferation of millions of both free software and OSS development projects that have created many “best of breed” operating system and application software products. Today, virtually all proprietary software includes open source software, and an increasingly large percentage of crucial software platforms and programs are entirely open source.

While terms like free software and open source software may sound innocuous, when properly understood they imply elements of political philosophy, revolutionary zeal, technical development methodologies, traditional as well as radical legal theories, and cold, hard business pragmatism. Needless to say, such a rich stew of attributes is likely to present something of a challenge to anyone interested in gaining a quick understanding of exactly what this phenomenon is all about.

The reasons for investing the time to gain a better understanding of FOSS are several. From a socio-political point of view, the FOSS movement is part of a broader, socio-political initiative, energized in part by the ability of the Internet to enable the sharing of information and the active collaboration of people on a global basis. In the case of the free software movement, that movement questions the utility and fairness of many traditional copyright and patent-based legal restrictions, and seeks to liberate

---

<sup>2</sup> In this article, I use the word FOSS to mean (a) software delivered in both machine-readable object code and human-readable source code, together with (b) the rights to modify, copy and distribute that under any license that complies with the “free software” OR the “open source” definitions that are discussed further below. When necessary, I use “free software” to refer to software that complies with the free software definition created by Richard Stallman and “OSS” to refer to any other software made available under a license approved by the Open Source Initiative as an “open source license.”

software for the benefit of all.<sup>3</sup> Unlike proponents of OSS, who primarily wish to permit open source software to be freely available without traditional proprietary constraints, free software advocates support a set of ethical rules intended not only to foster free access, but also to inspire — and in some cases require — those that benefit from such access to contribute their own modifications and additions back to the community of developers as well.

From an economic point of view, the OSS development model has reordered the business realities of software development in multiple ways: for a software vendor or user, the per-business costs of development of a given piece of software can be radically reduced by participating in a development project in which many others contribute their efforts as well; for an end user, access to the source code of an OSS product grants independence from a proprietary vendor, since the end user can adapt the code, or put development work out for competitive bidding; for commercial intermediaries, efforts can be directed towards developing value added services on top of core code that is available for free and maintained by a community of developers; and for policy makers, OSS offers opportunities to level the playing field for domestic vendors while lowering costs of procuring public IT systems. From a marketplace perspective, the OSS model presents a disruptive force that offers opportunities for both existing as well as new businesses to attack the dominance of entrenched market participants whose advantages rest on proprietary development and sales models.

Today, FOSS has become so pervasive that effective IT procurement and management requires a working knowledge of what FOSS is all about. Active participants in the development and use of FOSS products additionally

---

<sup>3</sup> Richard Stallman laid out the foundation for the concept of free software in 1981 in the GNU Manifesto, Stallman codified the definition of free software in 1986 in what he refers to as “the four essential freedoms.” They are:

A program is free software if the program’s users have the four essential freedoms:

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help others (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

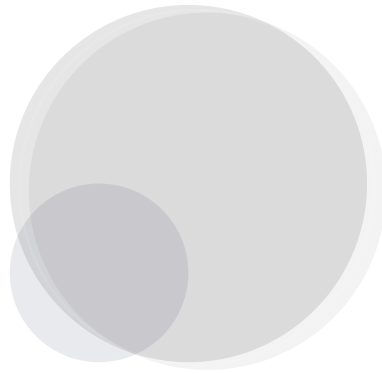
Importantly, Stallman’s use of the word “free” in the definition is not meant to have economic significance, although free software is usually available without cost. Rather, “free,” as Stallman explains, is meant to be “free as in speech, not as in beer.”

need to know how FOSS can be expected to evolve in the future, and how the legalities of FOSS apply to anyone that participates in the development of FOSS, uses a FOSS product, or embeds a FOSS code in their own products for resale.

In this article, I will provide an overview of the history of FOSS and its champion, the major philosophical differences that differentiate free software from other open source software, the multiple licenses under which FOSS is made available, and the principal non-profit institutions that support and promote FOSS. I will conclude with a brief bibliography of primary FOSS sources for those that wish to learn more than this necessarily superficial review can hope to provide regarding such a rich and complex topic.

---

*The next chapter of **A Concise Introduction - Free and Open Source Software** will explore what open source software is (and what it isn't) and its value proposition.*



# A CONCISE INTRODUCTION

## FREE AND OPEN SOURCE SOFTWARE

by Andrew Updegrave

---

2019  
AUGUST

