

## **The Software License Unveiled: How Legislation by License Controls Software Access**

By: Douglas E. Phillips

New York, NY: Oxford University Press, 2009, ISBN 978-0-19-534187-4

Price \$85.00, pp. 204

Reviewed by Andrew McQuilkin

Journal of High Technology Law

Suffolk University Law School

---

Just by entering a search in Google, you are entering into a contractual agreement with the company. Computer users enter such agreements almost daily. Most of the time without understanding the terms, or even realizing they are being bound. This seems like a departure from the general contractual principles of mutual assent and acceptance of an offer. Douglas E. Phillips attempts to explain this modern phenomenon in his work, *The Software License Unveiled: How Legislation by License Controls Software Access*, which explores language and legal implications of the software licenses currently in use. Phillips argues that companies, by only licensing software instead of selling it, have effectively created private legislation regarding its use by circumventing the Copyright Act. He then goes on to explore free and open source software licensing as an alternative to the traditional proprietary system.

Originally, software was not purchased separately from hardware, as it would come pre-installed and was tied directly to the particular machine.<sup>1</sup> In 1969, IBM “unbundled” the two, allowing for software and hardware to be sold separately.<sup>2</sup> The company could not rely on trade secret law to protect its software because the

---

<sup>1</sup> DOUGLAS E. PHILLIPS, *THE SOFTWARE LICENSE UNVEILED: HOW LEGISLATION BY LICENSE CONTROLS SOFTWARE ACCESS* 10 (Oxford University Press) (2009).

<sup>2</sup> MARTIN GOETZ, *MEMOIRS OF A SOFTWARE PIONEER: PART I*, IEEE ANNALS OF THE HISTORY OF COMPUTING 43 (2002).

development would require too many people to make secrecy practical.<sup>3</sup> Instead, it turned to copyright.<sup>4</sup> However, IBM did not believe the Copyright Act was strong enough for a product that could be copied so easily.<sup>5</sup> To fix this, it would not sell its software, but instead allow customers to use it under a license. The user would accept this license by either opening the box, or clicking a button within the program. At first the courts were wary of this practice, but almost all recent cases have held the licenses enforceable.<sup>6</sup>

The proprietary software license or EULA (End User License Agreement) comes in all shapes and sizes, and its contents are up to the individual needs of each software company. Some clauses, however, are common to these agreements, such as a disclaimer of all warranties, the list of permissible uses, prohibition of reverse engineering, and that the software is licensed not sold.<sup>7</sup> While most courts have held these clauses enforceable, there are still some unresolved legal issues, including whether a person can still be bound even if he or she was not the person who opened the package, or clicked “accept.”

In stark contrast to the proprietary EULA, stands the GPL (General Public License) and its creator Richard Stallman. As a well-known programmer in the 70s and 80s, Stallman lived in a culture where software was freely traded among users.<sup>8</sup> To preserve what he viewed as the free movement of ideas, he created his own licensing scheme.<sup>9</sup>

Unlike the EULA, which restricts use of a program, GPL is a permissive license. The user can modify, use or copy any part of the program freely, but only if several

---

<sup>3</sup> PHILLIPS, *supra* note 1 at 49. WATTS S. HUMPHREY, SOFTWARE UNBUNDLING: A PERSONAL PERSPECTIVE, IEE ANNALS OF THE HISTORY OF COMPUTING 59, 60 (2002).

<sup>4</sup> HUMPHREY, *supra* note 3 at 59.

<sup>5</sup> HUMPHREY, *supra* note 3 at 59.

<sup>6</sup> PHILLIPS, *supra* note 1 at 5.

<sup>7</sup> See PHILLIPS, *supra* note 1 at 7-14, 41-42.

<sup>8</sup> PETER K. YU, INTELLECTUAL PROPERTY AND INFORMATION WEALTH 421 (2007).

<sup>9</sup> *Id.*

conditions are met. The GPL requires that all modified programs must be distributed under the GPL, and that no royalties be charged.

While the GPL may further Richard Stallman's views on how software should be distributed, software companies cannot use it since they would not be able to sell anything they create. From this limitation, comes the development of a third licensing system, open source. Open source programs have almost no limitations on their use, modification, or distribution. Usually the only condition in these licenses is that, once modified, they cannot be called by the same name. This is actually more of a trademark than a copyright protection.<sup>10</sup> Because of its freedom, this system has been widely adopted and supported by companies, who make money selling hardware and services that run open source software.

Phillips takes on each type of license in turn, examining the legal and practical implications of each. With regards to the EULA, he argues that the software companies, by licensing and not selling, are effectively writing their own version of Copyright Act to govern permissible use. The companies prefer their own agreements because it allows them to vary their prices based on the customer and get explicit permission to install copy protection programs.

Phillips argues that there is no need of these licenses, and that copyright protection should be sufficient with a few changes to the law. For example, the first-sale doctrine of copyright protection would allow a single user to sell their own copy to a business, who could then install it in all of their computers. Under the current pricing model, that business would have had to pay a much higher price to the software company. To fix this, Phillips suggests that Congress amend the Copyright Act to cover

---

<sup>10</sup> See PHILLIPS, *supra* note 1 at 148, 150.

this and other such issues unique to the distribution of software. This would make the permissible uses much clearer to the customer by creating uniformity.

While this may make things simpler, Phillips fails to make the case for how the current licensing model has caused injury. Even if the average user inadvertently violated some of the complex terms, the software company has no motivation or effective method for enforcement. For a larger business violating the license, where enforcement would be more practical, they would have access to a legal team which could explain and negotiate the terms. While there is potential for abuse in the future, there is currently little harm in keeping the present system.

To further demonstrate the difficulty of understanding the terms of the EULA, Phillips calculates the readability of the EULA by using the Flesch-Kincaid formula. This uses average sentence and word length to determine how difficult a document is to read. Unsurprisingly, the formula found that the EULA is very tough to understand. A quick glance at the terms of the average software license would show the same thing, making this entire section of the book unnecessary.

Next, Phillips attacks the argument made by some EULA supporters that software is an intangible object and therefore the license to use it is what is actually being sold. This argument is used to assert that the license is required, because otherwise software companies would just be selling ideas and those have no copyright protection. An entire chapter is devoted to demonstrating how software is in fact tangible. Phillips attacks the argument from all fronts, relying on analogy to other products, the definition of tangibility, philosophy, and how software actually works. While intangibility represents only one argument for justification of the EULA, Phillips effectively refutes it.

The second half of the book deals with free and open source software licenses and is less concerned with their legal than their practical effects. Phillips views the GPL's demand of absolute freedom as overly constricting, and resulting in limited distribution and usability. He casts a more favorable light on open source software, and demonstrates how companies like IBM can put massive resources into free software, and make money by selling related hardware and services.

In the end, *The Software License Unveiled: How Legislation By License Controls Access*, is recommended for anyone interested in the future of software licensing. Phillips does a tremendous job of identifying, and explaining the current issues of the subject, both legal and practical. While the solutions proposed to such issues are not entirely convincing, it is an excellent place to begin the discussion.