APRIL
2023
VOL. 64

Software Rights: How Patent Law Transformed Software Development in America

By Gerardo Con Diaz. New Haven: Yale University Press, 2019. Pp. 384.

In *Software Rights*, author Gerardo Con Diaz sets out to trace the legal and business history of software in the United States over a period of fifty-five years, from the mid to late twentieth century. Although the primary focus of the work is on software patents, it is also necessarily a history of copyright, the complementary and alternative mode of intellectual property protection for software. Con Diaz draws on a wealth of industry and legal archives to survey a multitude of legal and business decisions that shaped the nature and character of the software industry.

To manage this cast of thousands, Con Diaz focuses on individual vignettes, flitting between quick sketches of actors and events. The result is that the reader traverses a narrative landscape that is surprisingly flat: bit players and starring actors receive similar treatment; major and minor events are described in equal detail. The reader who is already familiar with the history of software patenting may be surprised at the relatively cursory treatment of administrative or judicial decisions of great moment—but will be equally surprised at the lengthy discussion of other decisions, some of which the reader has never before heard of, that in hindsight have proved to be entirely unimportant and forgettable.

Throughout, the focus of the volume remains squarely on the historical record of the software industry, with little time for background explanation. For example, the American patent statute extends exclusive rights to qualifying processes, machines, articles of manufacture, and compositions of matter. Much of the controversy over software patenting that Con Diaz surveys revolves around which of these categories might fit computer programs: Are they processes? Or are they machines? Could they be articles of manufacture? Con Diaz documents the long struggle to draft patent claims around the statutory subject matter categories—but without ever explicitly examining the statute on which the controversy is based. The astute reader will likely pick up on this from context, albeit with some extra effort.

Similarly, much of the action that Con Diaz follows centers on evolving strategies or attempts to procure patents for software. Patents are only obtained by a process of administrative review through the federal Patent Office. Con Diaz does not pause to explain the process of patent procurement, or "prosecution" as it is called in the vernacular of the patent bar. Neither does his historical survey explicitly delve into the organization of the U.S. Patent Office or its administrative procedures. Still, after numerous references and allusions, the astute reader cannot help but come away with a general sense of such procedures, constructed from the activity the book does describe.

However, assembling that overview of patent procedure is left to the reader; it is not set out in detail.

The author's overview ends rather abruptly in 2000 with the publication of the U.S. Court of Appeals for the Federal Circuit's patent eligibility opinion in *State Street Bank*, which the author notes is the last major software case of the twentieth century. To the reader familiar with the field, this sudden halt is rather jarring. The year 2000 is a nice round number, the edge of a new millennium, and a perennial eschatological favorite. And of course, the book must end somewhere. But *State Street Bank* is perhaps the most disruptive and controversial patent case ever decided, leading to a decade of fierce contention over the scope of patents generally and over software patents in particular. The debates generated by *State Street* led directly to the dramatic intervention of the U.S. Supreme Court to reshape software patent eligibility and to the intercession of the U.S. Congress to radically revise the American patent system.

All of this makes the year 2000 cutoff something of a software cliffhanger. Some of the later developments are mentioned in a brief coda in which the author summarizes his findings and issues a very sensible call for further research. The overall feeling the reader is left with—perhaps inadvertently—is a haunting sense of what it might have been like to experience the history of software patenting at the time it was happening, without the hindsight understanding of what proved to be important and what did not. This makes Con Diaz's volume a useful contribution to the literature on software patenting, but one that must be read alongside complementary studies; it cannot easily stand on its own.

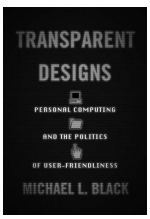
DAN L. BURK

Dan L. Burk is Chancellor's Professor and Distinguished Professor of Law at the University of California, Irvine. He is the coauthor, with Mark A. Lemley, of *The Patent Crisis and How the Courts Can Solve It* (University of Chicago Press, 2009).

Citation: Burk, Dan L. "Review of *Software Rights: How Patent Law Transformed Software Development in America* by Gerardo Con Díaz." *Technology and Culture* 64, no. 2 (2023): 618–19.

Transparent Designs: Personal Computing and the Politics of User-Friendliness

By Michael Black. Baltimore: Johns Hopkins University Press, 2022. Pp. 280.



Michael Black begins *Transparent Designs* by describing Steve Jobs's design philosophy for Apple. In 1984, Jobs explained that the company aspired "to reach the point where the operating system is totally transparent. When you use a Lisa or a Macintosh. . . . You never interact with it; you don't know about it." Black uses this anecdote (and many others like it) to illustrate the idea of transparent design and how it is hyped.