
Jon M. Garon
Exclusivity in the New Economy: Software Strategies

Jon M. Garon*

This article is part of a series of book excerpts from The Entrepreneur's Intellectual Property & Business Handbook, which provides the business, strategy, and legal reference guide for start-ups and small businesses.

1. **Building Friends in Software Gaming.**

   Software companies are built on copyright exclusivity. Copyright allows the author of the software to have exclusive rights over the expression created, which in the case of code means the precise programming language. However, the ideas or objectives of the software are not protected. Microsoft owns the copyright to Word, but it does not own the idea of word processing.

   Copyright may create a strong barrier against copying, but it provides only modest protection from legitimate competition. The challenge for software companies is to create works that are sufficiently unique. Only the most unique software can separate itself from other copyrighted works.

   Computer games provide the most obvious form of protection. Electronic Arts, the owner of The Sims, Madden Football, and many other titles, has found that the best way to protect copyright and build relevance is to add more and more exclusivity to the mix. Electronic Arts has extended copyrights, trademarks, and publicity rights to separate its titles from those of its competitors. For example, it brands its sports games with popular athletes and coaches in the titles of the games. “Madden Football” builds a stronger presence than “Pro Football” could provide. The publicity rights also extend beyond the packaging into the games themselves, with the names, images, and voices of the celebrities serving as elements of the games.

   Electronic Arts also uses team building in its other products. It regularly licenses signature music for its games from well-known popular bands, and it had a three-game deal with leading motion picture producer and director Steven Spielberg to design three new titles. In its sports division, it regularly licenses exclusive rights with professional sports leagues. By tying its software to the trademarks, publicity rights, and copyrights of other leaders in their respective fields, Electronic Arts has propelled itself to the top of the market.

2. **Bundling to Overcome Compatibility Learning Curve Barriers.**

   Consumers have somewhat odd behavior when it comes to computers. They regularly buy expensive new machines because the incessant software upgrades demand more productivity than their hardware can deliver. At the same time, they are a bit reluctant to spend money on new software and generally rely on whatever software comes bundled with their machines.

   For the software company fortunate enough to sign a bundling arrangement with a hardware manufacturer, the pre-loaded software on shipped computers provides a tremendous boost towards dominating that product segment. The fees paid by computer manufacturers to software companies

---

* Jon M. Garon, Dean and Professor of Law, Nova Southeastern University Shepard Broad College of Law; J.D. Columbia University School of Law 1988. Adapted from The Entrepreneur’s Intellectual Property & Business Handbook (reprinted with permission). Dean Garon is admitted in New Hampshire, California and Minnesota and of counsel with Gallagher, Callahan & Gartrell, PC, Concord NH.
tend to be low as a result, since the computer manufacturer also knows that the delivery of the
software is an extremely effective method to develop market share. Despite potentially low payments, this
strategy has helped Microsoft and others gain and sustain market share.

A second bundling strategy is used by smaller software companies and those writing for a niche
market. Software products are bundled into suites that share a common user interface. Microsoft
pioneered this strategy as well, and many others have followed. Bundling provides three key benefits
to consumers, and as a result, improves sales. First, a bundled product is likely to be, or at least appear
to be, a more cost-effective purchase for the consumer than a stand-alone product and a more efficient
delivery method for the manufacturer. Although the development costs for each component of the
bundled package may remain the same or actually increase, the costs associated with packaging,
marketing, and distribution are the same whether there is one product or ten in the package. Reducing
these distribution and packaging costs may be enough to make the bundled product more cost
effective.

Second, consumers are appropriately concerned with software compatibility. A bundled product
is presumably well integrated, so that all compatibility problems have been eliminated. There is also a
presumption that the larger products have greater quality control, so that the purchaser will have less
technical difficulty. The corollary of this presumption is that software compatibility problems will
destroy the marketability of a software product and should be eliminated at all costs.

Third, consumers are frustrated by the need to continually learn new software navigation, new
keystrokes, and different interfaces. The consumer will likely stay with the tools offered in the bundle,
even if they really only like one of the bundled products, because of the shared interface. Although
Microsoft Office provides the best example, Roxio has created a suite of tools for CD and DVD
creation, and Adobe’s Creative Suite combines Photoshop, InDesign, Illustrator, GoLive, Acrobat,
and many other products. Although Adobe has not fully integrated its user interfaces across its various
products, each product is, by itself, an extremely complex bundle of tools and programs, providing
the best example of the suite model.
3. **The Open Source Dilemma.**

Perhaps the best suite on the market today is the Google Chrome suite of tools for Internet browsing and e-mail. Google’s domination of search and its suite of free products have led to Chrome’s having 67% of all browsing, compared to Apple Safari’s 5.5% and Microsoft’s 11% split between Internet Explorer and Edge. Firefox is comparable in usage to Microsoft. Google’s dominance and free distribution have also led to success for products like G-mail and Google Drive.

Firefox is an open source product, as was the initial version of Chrome. Open source software remains protected by copyright. The copyright license allows copying without charge and adaptation or alteration to the software as long as the changes are then subject to the terms of the original open source software license. Copyright provides the legal protection necessary to stop companies from using open source software to produce proprietary products. Open source differs from public domain products, because once in the public domain, anyone can change the content. This includes changing the content in such a manner that it becomes exclusively owned by the person making the changes.

Open source software has an inherent limitation in its commercialization. A legitimate open source company cannot make revenue from the adaptation of the software. There is no exclusivity for the author of open source software. In the case of Chrome, for example, Google provided the public a free version of the software but incorporates proprietary updates into the version it provides.

At the operating system level, Linux is perhaps the best known open source software, but most users need much more assistance than is available from the free version of Linux. To meet this need, Red Hat created a strong service model to provide enterprise solutions for Linux adopters.

Red Hat has built a subscription model around the free product, presumably selling not the software but the services associated with the support for that software. If companies hope to use the subscription support model based on open source software, they must provide relevant service for a product that they do not exclusively control or provide. Process innovation and the elimination of stressors create opportunities for exclusivity through servicing, developing trade secrets in know-how, building brand reputation and trademarks, and potentially, promoting publicity rights. However, copyright and patent exclusivity are foreclosed by the nature of open source software licenses.

4. **Software Patents.**

In addition to the inherent ability to copyright software, some innovations rise to the level of patentability. Mathematical algorithms are not protected by patent, but there are many processes that are patentable. Patents have been issued to Unisys for computer graphics which overlapped with CompuServe’s GIF format, for encryption, computer style sheets, and for many more seemingly mundane features.

As an industry, the bulk of academic and empirical literature tends to reject the efficacy of software patents. Research suggests that the competitive need for innovation and service is not further promoted by the potential for patent protection. Having acknowledged this, the availability of patents to differentiate an entrepreneur’s business and the risk that another company or enterprise has patented an innovation under development are both important considerations which must be taken into account.

Finally, there may be the truly novel extension of software that is so far beyond the state of the industry that patents are appropriate. Dan Bricklin and Bob Frankston invented VisiCalc as the first
spreadsheet, before software patenting was available.\(^1\) That product was later sold to Lotus to create Lotus 1-2-3, a product that revolutionized PC sales and fueled the PC revolution. The authors of this revolution did not receive nearly the financial reward earned by less creative business owners. If there is a defense for software patents, the ability to have rewarded this caliber of innovation may be its best support.