

Patent Failure

By Stephen Redd, English 101

In the spring of 2012, the Samsung Galaxy S3 became one of the most eagerly anticipated smartphone releases ever. Expected to ship in the summer of 2012, continuous parades of T.V. commercials, geek bloggers, and YouTube reviewers foretold the imminent arrival of the new king of smartphones. Three months and 18 million units later, the Samsung Galaxy S3 bested Apple's iPhone 4s to become the most popular smartphone on the planet ("Strategy Analytics"). Prior to release though, Apple Computer Corporation attempted to garrote Samsung and ensure the premature death of the prophesized king of smartphones.

Apple's attack against Samsung alleged that the phone's software was composed of ideas to which Apple held the patents. The problem for many is that these patents, and the inventions claimed thereby, seem outright absurd. As an example, Apple patent 305 claims that Apple invented the idea of rounding the corners of square icons (Smith).

Apple is far from the only company to litigate seemingly inane software patents. In 1999, Amazon.com won the infamous one-click case, preventing rival Barnesandnobel.com from selling books through their web site via a single click payment method (Hansell). In 1994, E-Data Corp. attempted to claim ownership over the entire World Wide Web through a patent granted six years prior to the invention of the Web ("Everlasting Software" 1455). Thousands of similar cases over the years feature one company bludgeoning another with incremental, trivial, or obvious software patent claims.

Few argue that the existing software patent system is a healthy one, but deep controversies over how to improve the situation remain. One camp argues for the elimination of software patents on the premise that they do more harm than good, while another argues for reforms to restore sanity and continue to encourage healthy software innovation. Until the matter is settled, it is likely that costly and abusive patent infringement cases will continue to plague the software industry.



Figure 1 Apple phone/Wikimedia Commons

Patents have a long history and are a complex topic comprised of law, judicial precedent, and international treaty. According to the *Oxford English Dictionary Online*, a patent is "a license from a government conferring for a set period the sole right to make, use, or sell some process or invention; a right conferred in this way." Put concisely, a patent is a government-approved monopoly.

Article I of the U.S. Constitution obliges the government to grant patents by proclaiming they "promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries" (Sec. 8, Cl. 8). The Patent Act of 1790 placed the initial burden of patent oversight onto the shoulders of the Secretary of War, Secretary of State, and the Attorney General. By 1836, the task had grown so onerous as to prompt the establishment of an official Office of Patents. Since then, numerous

additional restructurings culminated in today's U.S. Patent and Trademark Office, and the rules by which it operates ("A Brief History").

The issue of what should be patentable, and for how long, has always been thorny; however, the justification for patents is rather simple. Invention is time consuming and risky. An inventor may labor for years only to fail, or a rival inventor may beat him to market. Patents encourage inventors to brave such risks by guaranteeing a limited monopoly over successful inventions, during which time inventors can recoup their costs and earn a profit (Bessen and Maskin 2). Since inventions are of immense public value, patents also ensure public disclosure, thus preventing advanced knowledge from following jealous inventors to their graves (Nieh 307-308).

Traditional patents operate under complex rules, but patent applications must fully describe the intended functionality, as well as the means by which the functionality is achieved. For physical devices, this amounts to a detailed description of the idea, as well as the physical characteristics of the device implementing the idea ("Everlasting Software" 1457). The Patent Act of 1952 codified the modern requirements of a valid patent under U.S. Law. It established a list of the specific categories of invention considered patentable and formalized the requirements that the invention be novel, non-obvious, and useful (35 USC 101, 2007).

The 1952 Patent Act does not explicitly name software as patentable. In 1981, the Supreme Court ruled on the case of *Diamond v. Diehr*, concluding that software does fall under the statutory label of a "process." This early case centered on software as a mere component in a larger invention with physical results, the molding of rubber. While the case made software patentable, it did so only in a narrow set of circumstances. Software patents as we know them today only became possible in 1998 when the Federal Circuit Court ruled in the case of *State Street Bank & Trust Co. v. Signature Fin. Group*. This case set the modern precedent under which software became fully patentable

without the need for specific physical characteristics or outcomes (Nieh 303).

Software patents have always been a contentious subject, raising many concerns across the industry. Is software an invention deserving of patent protections, or does the nature of software render it ineligible? Are patents a useful means of incentivizing software innovation? Does the industry benefit from patents, or has patent litigation resulted in more damage than good? These issues are growing in importance every year, but definitive answers remain elusive.

Opponents of software patents have long contended that the nature of software makes a poor subject for patent protection. Traditional patents include a "functional claim" describing what result to be achieved, as well as a "method claim" detailing how the invention achieves that result ("Everlasting Software" 1457-1460). In reviewing patent claims, the Federal Circuit decides validity based on the claims made in the filing. Concentrating on the functional claim of the invention's achievement comprises the "written description requirement" doctrine, where a focus on the method claim is "enablement doctrine" (Merges 1648-1652).

For patent disputes, both functional and method claims play a role. With physical devices, enablement doctrine has been dominant; the result being that infringement cases are decided by the merits of the method claim. This is important because the method claim of physical patents specifies numerous physical characteristics and behaviors. A subsequent invention, having a similar functional claim to the first, can avoid infringing by achieving the same result in a different way; both having the same functional claim but differing in the method by which the function is accomplished. This allows new, and hopefully better, versions of prior inventions to develop, without infringing prior patents ("Everlasting Software" 1457-1458).

With software, incremental innovation without infringing prior patents becomes troublesome. Software has no physical

characteristics, and so the functional claim is essentially the entire claim. Any method claim in a software patent tends to be highly generic and vague. This has resulted in the Federal Circuit largely abandoning enablement doctrine in favor of written description requirement when deciding infringement cases for software patents. Since software patents are claimed by functionality alone, they encompass a much wider range of invention than is typical for physical patents. Software will always infringe if it achieves similar functionality regardless of how that functionality may be physically implemented (“Everlasting Software” 1464-1466). The functional basis for software patents also results in a broader scope of claims than for physical patents. This often allows software patents to apply to seemingly unrelated software inventions, even ones the original inventor could not have anticipated (“Everlasting Software” 1459-1460).

Proponents of software patents acknowledge the unusual breadth of software patents and the abuse that sometimes follows, but assert that the broad scope is simply a by-product of necessarily generic and vague method claims. Simple reforms could narrow the scope by either returning to a balanced version of enablement doctrine, or through modification to the application process that narrow the scope of functional claims (Merges 1654-1657).

Opponents of software patents hypothesize that overly broad claims are the unavoidable result of an inherent incompatibility between software and the patent system. Software is, by nature, incrementally constructed. Each software project builds on earlier software efforts. For this reason, opponents claim that patent protections are untenable when applied to

software. Patents also require that the functional claim be novel, and non-obvious. If all software is an incremental advance over previous developments, then it follows that all software is an obvious invention, lacks novelty, and is therefore not patentable (Nieh 319-320).



Figure 2 Samsung Phone/Wikimedia Commons

As interesting as the patentability arguments may be, the 1998 State Street decision by the Federal Circuit temporarily settled the matter by proclaiming that software is patentable under current law. The discussion over the last decade has shifted from whether software *can* be patented, to whether it *should* be. Today’s critics ask whether software patents do spur innovation, encourage new entrants to the market, and protect R&D investments (Merges 1628-1629).

As with general patentability, the arguments around the usefulness of software patents also rest largely on an examination of the nature of software. In an M.I.T. working paper, James Bessen and Eric Maskin explain that software

innovation is not just incremental, but also complimentary in nature:

This is because these are industries in which innovation is both *sequential* and *complementary*. By “sequential,” we mean that each successive invention builds on the preceding one—in the way that Windows built on DOS. And by “complementary,” we mean that each potential innovator takes a somewhat different research line and thereby enhances the overall probability that a particular goal is reached within a given time. Undoubtedly, the many different approaches taken to voice recognition software, for example, hastened the availability of commercially viable packages. (2-3)

The sequential and complementary view of software development argues that patent protections interfere with the natural process of innovation, which occurs through imitation. Imitation among software developers encourages parallel innovation along different paths, which in turn accelerates the pace of innovation. As a result, they argue, patents limit the necessary free flow of ideas and prevent complimentary imitation, the result of which is lost opportunities and a slower pace of advance (Bessen and Maskin 3-4).

Proponents of software patents argue that there is no evidence that patents have significantly slowed the pace of invention, nor have they prevented the efficient sharing of ideas. Software patents are often licensed between competing companies, allowing non-infringing complimentary development to continue (Bessen and Maskin 3). Others claim that technological advances and financial investment have been the primary source of software innovation and industry growth, while patents have not played a significant role (Merges 1641-1642).

Opinions as to the financial effects of software patents are another hotly contested aspect of the software patent dispute. Publishing in the *Texas Law Review*, Robert Merges provides statistics that show healthy growth for software R&D and venture funding over the years. In 1986, software development claimed only 1% of the R&D budgets at major software companies. By the year 2000, software spending rose to 10% of total R&D expenses (1635). Merges's data also shows that venture investment in software startups has flourished. In 1995, 210 startups received 553 million in venture capital. By 2005, 238 startups claimed 1.1 billion in funding (1639-1640).

Software patent abolitionists argue that the growth of the software industry and of software R&D spending occurred despite patents, not because of them. When looking at R&D spending within companies, instead of across the industry, the trend during the 1990s showed that companies were diverting funds into the creation of large patent portfolios at the expense of R&D

budgets (Neih 310). The licensing of patents has not served to promote the responsible sharing of ideas, but has instead become a necessary means of defend against patent litigation, or the threat of such. This cold war strategy drains money from software development and diverts it into unproductive legal maneuverings. A 2003 report by the U.S. Federal Trade Commission remarks on this trend toward increased patent spending:

Moreover, as more and more patents issue on incremental inventions, firms seek more and more patents to have enough bargaining chips to obtain access to others' overlapping patents. One panelist asserted that the time and money his software company spends on creating and filing these so-called defensive patents, which "have no . . . innovative value in and of themselves," could have been better spent on developing new technologies. (United States. Federal Trade Commission 6-7)

Building defensive patent portfolios drains budgets at a controllable rate, but patent abolitionists are just as concerned about the less predictable costs and risks associated with patents that do go into litigation. In the *New York Law School Law Review*, Andrew Neih writes that, "software patents have a 4.6% likelihood of being involved in a lawsuit, the second highest among all categories of technological patents" (312). Software patent litigation is also increasing, up from 5% of the total number of patent suits filed in 1984 to 26% of all patent suits filed in 2002. The high probability of litigation forces less spending on software development, and discourages venture investment in new software projects (Neih 314-315).

Patent supporters claim that litigation is an expected, and proper, result of patent protections granted to software inventions. The purpose of a patent is to provide legal protection; therefore, it is reasonable to expect that litigation will occur whenever a patentees needs to defend their intellectual property. With software patents becoming an accepted norm during the late 90s, it is reasonable that the number of patent claims has risen along with the number of filings. Patent supporters acknowledge the many high-

profile cases of patent abuse, but caution that patents work well in more cases, though the successes do not enjoy the same level of publicity. The evidence does not show that the litigiousness of some software companies offsets the positive incentives and protections that software patents provide (“Everlasting Software” 1474-1475).

The issues around software patents have come to the public’s notice more frequently in recent years. Recent infringement cases against Google make a prime example. Google, having entered the software market relatively recently, has a smaller patent portfolio than most of its established competitors. Google has become a popular target of patent suits from established software giants such as Apple, Oracle, and Microsoft. Oracle sued for patents related to Oracle’s Java programming language (Brodkin). Microsoft has been licensing patents to phone manufacturers who use Google’s Android operating system and filing suits against any who refuse to license. Microsoft now earns more profit from the sales of Android-based smartphones than Google, despite the fact that Microsoft contributed nothing to the development of the Android operating system (McDougall). Apple has also been suing Google along with almost every manufacturer of smartphones using Google’s software (Barrett).

The pressure on Google from patent suits grew so great, that in August 2011 Google shelled out \$12.5 billion to acquire Motorola. The entire motivation for the deal was to acquire Motorola’s extensive patent portfolio in hopes of defending itself against competitors who prefer to compete in court, rather than the market (Thomas). In the meantime, Apple continues to win injunctions that have prevented the sales of several smartphones made by HTC and Samsung, while Microsoft continues to extort license fees from most smartphone manufacturers. Google finally won its lengthy case against Oracle in the spring of 2012, but the court victory came at enormous expense (Mullin).

Google’s woes are just the tip of a very large and ugly iceberg. Similar patent disputes have erupted all over the industry, especially in the emerging tablet and smartphone markets where innovation is critical and competition is fierce.

Patent advocates and abolitionists continue to disagree on solutions, but recent legal events have renewed the hopes of both camps. The 2010 ruling of the Supreme Court in the case of *Bilski v. Kappos* has bolstered the anti-patent movement’s viewpoint. In this complex case, the court called into question the validity of business process patents, which closely mirror the legal theory behind software patents. While the case was too narrow as to invalidate software patents outright, the ruling did express the Supreme Court’s doubts as to the validity of functional process patents. The *Bilski* case fails to settle the matter of software patents, but it does imply that, in future cases, the court may declare software non-patentable (Neih 323-326).

Those who prefer a reformed software patent system had some of their desires granted through passage of the America Invests Act of 2011. This legislation enacted numerous reforms to the current patent system, several aimed at curtailing the rising abuses seen in the software industry. The new law seems to have fallen short in many critical ways, but it does make progress toward real patent reform. Two of the most important changes are that prior-user rights can be a viable defense in infringement cases, and non-practicing entities (NPEs), which hold patents but do not sell goods or services based on the patents, are required to file separate suites rather than targeting multiple companies in a single case. Reformers are not fully appeased by the final bill, but the enactment of such legislation shows that Congress is at least aware of the need for significant reform (“Patently Inadequate” 18-20).

Unless Congress significantly reforms software patent law or the Supreme Court abolishes them entirely, the patent war will likely continue to accelerate. Currently, the most visible disputes have been between industry titans with enough resources to fight protracted patent battles and amass the necessary defensive

portfolios to survive. Left unchecked though, the patent war will spread to smaller companies and compromise innovation for more popular consumer products. Small shops will find themselves nuked out of existence by established players wielding their mighty arsenals of patents, while real software innovation stagnates in favor of lawyers' salaries and spiraling litigation costs. Software patents, as they stand today, only incentivize inventive ways to file overly broad patent claims.

Works Cited

- "A Brief History of the Patent Law of the United States." *Ladas & Parry, LLC*. Ladas & Parry, LLC., 17 July 2009. Web. 11 Nov. 2012.
- Barrett, Paul M. "Apple vs. Samsung: The Longer View." *Businessweek*. Bloomberg L.P., 30 Aug 2012. Web. 17 Nov 2012.
- Bessen, James, and Eric Maskin. "Sequential Innovation, Patents, and Imitation." *Research on Innovation*. Research on Innovation, Nov. 1999. Web. 9 Nov. 2012.
- Brodkin, Jon. "Oracle's IP War against Google Finally Going to Trial: What's at Stake." *Ars Technica*. Ars Technica, 15 Apr 2012. Web. 15 Nov 2012.
- "Everlasting Software." *Harvard Law Review* 125.6 (2012): 1454-1475. *Academic Search Premier*. Web. 3 Nov. 2012.
- Hansell, Saul. "Amazon Wins Court Ruling to Protect Patent on Order System." *The New York Times*. The New York Times Company, 3 Dec. 1999. Web. 27 Oct. 2012.
- McDougall, Paul. "Microsoft Inks More Android Licensing Deals." *InformationWeek*. UBM LLC., 8 Nov 2012. Web. 10 Nov 2012.
- Merges, Robert P. "Software and Patent Scope: A Report from the Middle Innings." *Texas Law Review* 85.7 (2007): 1627-1676. *Academic Search Premier*. Web. 4 Nov. 2012.
- Mullin, Joe. "Oracle v. Google: no patent infringement found." *Ars Technica*. Ars Technica, 23 May 2012. Web. 15 Nov 2012.
- Nieh, Andrew. "Software Wars: The Patent Menace." *New York Law School Law Review* 55.1 (2010): 295-330. *Academic Search Premier*. Web. 4 Nov. 2012.
- "patent." *Oxford English Dictionary Online*. Oxford University Press, Sept. 2012. Web. 13 November 2012.
- "Patently Inadequate." *Communications of the ACM* 55.7 (2012): 18-20. *Academic Search Premier*. Web. 4 Nov. 2012.
- Smith, Kevin. "Apple versus Samsung." *Business Insider*. Business Insider, Inc., 28 Aug. 2012. Web. 4 Nov. 2012.
- "Strategy Analytics: Samsung Galaxy S3 Becomes World's Best-Selling Smartphone Model in Q3 2012." *MarketWatch*. Business Wire, 8 Nov. 2012. Web. 12 Nov. 2012.
- Thomas, Owen. "Why Google Really Bought Motorola." *Business Insider*. Business Insider, Inc., 5 Nov. 2012. Web. 16 Nov 2012.
- United States. Federal Trade Commission. *To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy*. Washington: GPO, Oct. 2003. Print.
- 35 USC Sec. 101. 2007. Print.
- U.S. Constitution*. Art. I, Sec. 8, cl. 8. Print.