Summer 2012

A Generation of Software Patents

James Bessen
Boston University School of Law

ARTICLE

A GENERATION OF SOFTWARE PATENTS

JAMES BESSEN[*]

Author information:
jbessen@gmail.com
202 High Head Rd.
Harpswell, ME 04079
207-725-4237

ABSTRACT

This study examines changes in the patenting behavior of the software industry since the 1990s. It finds that most software firms still do not patent, most software patents are obtained by a few large firms in the software industry or in other industries, and the risk of litigation from software patents continues to increase dramatically. Given these findings, it is hard to conclude that software patents have provided a net social benefit in the software industry.

[*] Lecturer, Boston University School of Law, and Fellow, Berkman Center for Internet and Society at Harvard University.

I. INTRODUCTION

In 1994, the Court of Appeals for the Federal Circuit decided in *In re Alappat* that an invention that had a novel software algorithm combined with a trivial physical step was eligible for patent protection.[1] This ruling opened the way for a largescale increase in patenting of software.[2] Alappat and his fellow inventors were granted patent 5,440,676, the patent at issue in the appeal, in 1995.[3] That patent expired in 2008. In other words, we have now experienced a full generation of software patents.

The *Alappat* decision was controversial, not least because the software industry had been highly innovative without patent protection.[4] In fact, there had long been industry opposition to patenting software. Since the 1960s, computer companies opposed patents on software, first, in their input to a report by a presidential commission in 1966[5] and then in amici briefs to the Supreme Court in *Gottschalk v. Benson* in 1972.[6] Major software firms opposed software patents through the mid-1990s.[7] Perhaps more surprising, software developers themselves have mostly been opposed to patents on software. Surveys of software developers in 1992 and 1996 reported that most were opposed to patents.[8]

[1] *In re* Alappat, 33 F.3d 1526, 1543-45 (Fed. Cir. 1994).

[2] James Bessen & Robert M. Hunt, *The Software Patent Experiment*, FED. RESERVE BANK OF PHILA. BUS. REV., 3d Quarter 2004, at 24, *available at* http://www.phil.frb.org/files/br/brq304rh.pdf.

[3] U.S Patent No. 5,440,6876 (filed Jan. 28, 1988).

[4] *See, e.g.*, Peter J. Ayers, *Interpreting In re Alappat with an Eye Toward Prosecution,* 76 J. PAT. & TRADEMARK OFF. SOC'Y 741, 741 (1994) ("Both the electronics industry and the patent bar were anxiously awaiting [the Alappat] decision because one of the issues involved addressed the patentability of inventions that can be implemented in either hardware or software.").

[5] *See* Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025, 1028 n.3 (1990).

[6] *Id.* at 1143 n.458. Significantly, computer companies later changed their views on software patenting. *Id.* at 1143 n.458. For an overview of the early legal changes regarding the patentability of software, see *id.*

[7] *See* USPTO, PUBLIC HEARINGS ON USE OF THE PATENT SYSTEM TO PROTECT SOFTWARE-RELATED INVENTIONS, TRANSCRIPT OF PROCEEDINGS (1994) *available at* http://www.uspto.gov/web/offices/com/hearings/index.html.

[8] Effy Oz, *Acceptable protection of software intellectual property: A Survey of Software Developers and Lawyers,* 34 INFO. & MGMT. 161, 161-173 (1998); Pamela Samuelson et al., *Developments on the Intellectual Property Front*, 35 Communications of the ACM 33

Given this controversy, economists began looking at the use of software patents in the software industry during the 1990s. Two results stand out from this initial literature. First, relatively few software firms chose to acquire patents.[9] Second, software patents were much more likely than other patents to be involved in litigation.[10]

These two findings suggest that the extension of patent eligibility for software might not have been socially beneficial, at least not for the software industry. The low patenting rates suggest that patents might not have provided significant benefits to software firms, although software patents are, in fact, used heavily in other industries.[11] And the high litigation rates might imply high social costs that would outweigh these meager benefits. In addition, the litigation might also create disincentives for investing in innovation.

However, this intuition about social benefit is not conclusive. Some people have attributed problems with software patents to the newness of these patents.[12] This argument might be called the "adaptation hypothesis."[13] Possibly, the inexperience of patent examiners, the unfamiliarity of software firms with the patent process, and the difficulty of legal interpretation in court cases[14] for this new subject matter might explain the initial findings. It is possible that critical conditions have changed, that these problems were only temporary. For example, the patent office has now hired many examiners with

(1992).

[9] *See* Bessen & Hunt, *supra* note 2, at 25 ("In the second half of the 1990s, firms in the software industry received . . . at most 7 percent of all software patents"); James Bessen & Robert M. Hunt, *An Empirical Look at Software Patents*, 16 J. ECON. & MGMT. STRATEGY 157, 171 (2007) [hereinafter Hunt]; Ronald J. Mann, *Do Patents Facilitate Financing in the Software Industry?*, 83 Tex. L. Rev. 961, 965 (2005); Iain M. Cockburn & Megan J. MacGarvie, *Entry and Patenting in the Software Industry*, Table 3 (NBER Working Paper No. 12563, 2006), *available at* http://people.bu.edu/cockburn/cockburn_macgarvie_entry_and_patenting_in_software.pdf.

[10] JAMES BESSEN & MICHAEL J. MEURER, PATENT FAILURE: HOW JUDGES, BUREAUCRATS, AND LAWYERS PUT INNOVATORS AT RISK, 18 (2008).

[11] *See* Hunt, *supra* note 9, at 171.

[12] *See* Martin Campbell-Kelly, *Not All Bad: An Historical Perspective on Software Patents*, 11 MICH. TELECOMM. & TECH. L. REV. 191, 193-95 (2005).

[13] *See id*. at 195 (The initial problems with software patents stems from the newness and unfamiliarity of the patents, but over time the patent office and software industry will learn to adapt.).

[14] *See* Iain M. Cockburn & Megan J. MacGarvie, *Patents, Thickets, and the Financing of Early-Stage Firms: Evidence From the Software Industry*, 18 J. ECON. & MGMT. STRATEGY 729, 751-52 (2009) (Discussing the evolution of software protection in the courts).

computer backgrounds.[15] Also, aside from the adaptation hypothesis, court decisions in cases such as *KSR International Co. v Teleflex Inc*.[16] and *Bilski v Kappos*[17] substantively changed the law affecting software patents.

This report updates the picture of software patents by reviewing the literature and by updating some of the empirical analysis to ask:

1. whether software firms now appear to receive greater benefit from software patents as shown by their patenting behavior, and,
2. whether the risk of litigation from software patents has mitigated, and, if so, why.

The answers to these questions should provide some guidance about whether the great software patent experiment launched by the courts has been socially beneficial.

The focus of this report is the software industry rather than the broader range of industries that use software patents, such as the electronics, computer and communications industries. I focus on the software industry for several reasons. First, this has long been an innovative industry with a highly talented pool of programmers (about one third of all programmers work in this industry).[18] If patents provide strong benefits, this should show up clearly in this industry. On the other hand, much of the use of software patents in other industries might not reflect significant innovation. Perhaps many software patents in other industries combine software with old technologies, as in *Diamond v. Diehr*, reflecting rather routine innovation. Moreover, some research suggests that these other industries increased their patenting for strategic reasons rather then because of increased innovative activity.[19] The

[15] Stuart J.H. Graham & David C. Mowery, *Intellectual Property Protection in the U.S. Software Industry*, Berkeley 7-9, *available at* http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.87.6776 (The USPTO initiated the Business Methods Patent Initiative in early 2000, and one of the stated goals was to hire over 500 patent examiners specializing in software, computer, and business method patents.).

[16] 550 U.S. 398 (2007) (This decision made granting a patent more difficult because the patent examiner no longer needs to show a teaching, suggestion, or motivation to combine the prior art in order to overcome non-obviousness. This affects software patents because many combine old technologies with software in ways that would now be found to be obvious.).

[17] 130 S. Ct. 3218 (2010) (The Supreme Court rejected the Federal Circuit's Machine or Transformation test as the sole test in determining whether a process is patentable and thus eliminated a bright-line rule on the patentability of software inventions.).

[18] Bessen & Hunt, *supra* note 2, at Table.

[19] *See e.g*. Bronwyn H. Hall & Rosemarie Ham Ziedonis, *The Patent Paradox Revisited:*

heavy patenting of software in these industries might be evidence of efforts to inexpensively bulk up firm patent portfolios — software patents, unlike most other technologies, require little evidence of a working invention other than a high level description of the idea.[20] Finally, the software industry has notoriously low barriers to entry in contrast to many of these other industries where large complementary assets provide substantial barriers to competition.[21] This means that if patents provide substantial protection for innovations, this should show up most clearly in the software industry.

II. LITERATURE ON THE POSITIVE AND NEGATIVE INCENTIVES OF SOFTWARE PATENTS

Patent rights can potentially provide a variety of social benefits. Patents can provide incentives to invest in innovation, including investment in R&D.[22] Patents can also facilitate contracting for trading in technology[23] and for financing startup firms.[24] Finally, the disclosure of technical knowledge in the

An Empirical Study of Patenting in the US Semiconductor Industry, *1979-1995*, 32 RAND J. ECON. 101, 109-10 (2001) (Firms began amassing large patent portfolios in order to defend against lawsuits.).

[20] Dan L. Burk & Mark A. Lemley, *Is Patent Law Technology-Specific?,* 17 BERKELEY TECH. L.J. 1155, 1156 ("The Federal Circuit has essentially excused software inventions from compliance with the enablement and best mode requirements."); DAN L. BURK & MARK A. LEMLEY, THE PATENT CRISIS AND HOW THE COURTS CAN SOLVE IT 120 (2009) (The Federal Circuit has found "high-level functional descriptions to satisfy both enablement and best mode doctines.")**.** *See also* Northern Telecom, Inc. v. Datapoint Corp., 908 F.2d 931, 946 (1990); *and* Fonar Corp. v. General Elec. Co., 107 F.3d 1543, 1549 (Fed. Cir. 1997).

[21] *See* Seldon J. Childers, *Don't Stop the Music: No Strict Products Liability For Embedded Software*, U. FLA. J.L. & PUB. POL'Y 125, 175 n.205 (2008) ("Barriers to entry are an economic concept representing the cost for a new competitor to enter the market. Since all a programmer needs is access to a computer in order to enter the market, the barriers to entry are extremely low.").

[22] Bessen & Hunt, *supra* note 2, at 28 (By granting strong property rights with a patent, the government incentivizes firms to innovate by allowing firms to exclude others from practicing the invention and thereby allowing firms to realize more returns from their invention.).

[23] Firms will be more willing to buy and sell technology when they are confident in the ownership rights to the technology.

[24] *See* Mark A. Lemley, *Rational Ignorance at the Patent Office*, 95 NW. U. L. REV. 1495, 1505-6 (2001) ("Venture capitalists use client patents (or more likely, patent applications) as evidence that the company is well managed, is at a certain stage in development, and has defined and carved out a market niche.").

patent specification can help spread knowledge.

However, none of these benefits can be realized unless firms choose to obtain patents and, as noted above, initial studies found that software firms generally obtained few software patents (or any other), both in absolute numbers and also relative to firms in other industries.[25] Thus the patenting rate of software firms provides a first order measure of the possible social benefit. Bessen and Hunt found that software publishing and software services industries combined accounted for only 7% of software patents during the 1990s, despite employing 33% of the computer programmers.[26] Most software patents were acquired by other industries. Controlling for firm characteristics, Bessen and Hunt found that comparable firms in the machinery and computer industry acquired four times as many software patents as software firms and firms in the electronics industry acquired nearly ten times as many.[27] As noted above, firms in these industries might patent for strategic reasons rather than to protect innovations.[28]

Bessen and Hunt also found that *startup* software firms had even lower patent propensity than established software firms during the 1990s.[29] Similarly, Mann and Sager found in a survey of 877 venture capital backed software startups that only 24% had applied for any patents within four years of receiving funding.[30] Cockburn and MacGarvie conducted a fine-grained analysis of 27 software markets for 1994 - 2004 and found that in most segments, 80-95% of the firms had no patents related to that segment.[31] Only 20% of the startup software firms in Cockburn and MacGarvie's sample ever filed for a patent during the sample period.[32]

More recently, Grahamand colleagues conducted a survey of startups founded since 1998.[33] Of all software startups, only 24% held patents at the

[25] *Supra* note 9.

[26] Bessen & Hunt, *supra* note 2, at Table 1; *see also* Hunt, *supra* note 8, at 171. These figures exclude IBM, which switched from being primarily a hardware vendor to a software vendor during this time.

[27] Bessen & Hunt, *supra* note 2, at Table 1.

[28] *See supra* note 19.

[29] Bessen & Hunt, *supra* note 2, at 27.

[30] Ronald. J. Mann & Thomas W. Sager, *Patents, Venture Capital, and Software Start-ups*, 36 RES. POL'Y 193, 197 (2007).

[31] Cockburn and MacGarvie, *supra* note 9, at 29.

[32] Cockburn & MacGarvie, *supra* note 12, at 769.

[33] Stuart J.H. Graham et al., *High Technology Entrepreneurs and the Patent System: Results of the 2008 Berkeley Patent Survey,* 24 BERKELEY TECH. L.J. 1255, 1270-71 (2009).

time of the survey (in 2008).[34] Survey respondents at software startups also reported that patents did not provide important incentives to them.[35] However, 67% of software startups backed by venture capital held patents, marking a sharp change from the earlier study by Mann and Sager.[36]

This result is notable since some researchers have posited that patents facilitate financing because they provide a "signal" of firm quality to potential investors.[37] However, the empirical support for this hypothesis for software startups has been difficult to disentangle from other explanatory factors. Mann conducts a regression analysis and finds that "patenting practices have at best a minuscule ability to predict the success of a venture-backed software startup".[38] Cockburn and MacGarvie find a positive correlation between patent applications (but not patent grants!) and the probability of financing, IPO and acquisition.[39] However, they note that this cannot be taken as evidence in support of the signaling hypothesis because firms with better technology might be more willing to patent and also might be more likely to be funded.[40] Moreover, they note that this correlation might simply reflect a reverse causality: "it may be that investors require early-stage firms to file patent applications as a condition of receiving funds, or that applications are observed disproportionately by firms that get funding and are more able to support the substantial costs of patent prosecution."[41] I will discuss the signaling hypothesis further below in the context of new data on patenting by software IPOs.

The private value of patents provides another metric with which to measure the social value of software patents. To the extent that the private value

[34] *Id.* at Table 1, 1277.

[35] *See id.* at 1283-87.

[36] *See id.* at Table 1, 1277.

[37] *See* Clarissa Long, *Patent Signals*, 69 U. CHI. L. REV. 625, 637 (2002) ("By acting as a signal, possession of intellectual property may reduce the cost of communicating private information to the market regarding the financial prospects of the firm. "); Mann, *supra* note 8, at 993 ("[F]irms that obtain patents tend to be more careful in their engineering work and have a better understanding of what is special about their products than competitors that do not have patents."); David H. Hsu & Rosemarie H. Ziedonis, *Patents as Quality Signals for Entrepreneurial Signals*, ACAD. MGMT. ANN. MEETING. PROC., 1-6 (2008).

[38] Mann, *supra* note 8, at 981; *see also* James Bessen, *A Comment on 'Do Patents Facilitate Financing in the Software Industry?'* 6 (BOS. U. SCH. OF L, Working Paper No. 06-13, 2006), *available at* http://ssrn.com/abstract=912521.

[39] Cockburn and MacGarvie, *supra* note 12, at 767.

[40] *Id.* at 769.

[41] *Id.*

represents socially desirable incentives to invest and to trade technology, then it also represents a major component of the social value of these patents. Researchers have used different methods to estimate private patent value. Bessen and Meurer conducted an analysis of the value of software patents based on the payment of patent maintenance fees.[42] They found that software patents were substantially less valuable than other patents.[43] Hall and MacGarvie find no significant correlation between firm market value and patents for software firms in the period after legal changes eased the patenting of software.[44]

Other researchers have attempted to measure the incentive effect of patents by measuring the effect of software patents on firm R&D. Bessen and Hunt found that firms that increased their software patents relative to other patents decreased their R&D spending.[45] Lerner and Zhu, on the other hand, looked at firms that they judged to be in market segments where user interface features were important.[46] This sample was chosen on the theory that legal decisions limiting copyright protection for user interfaces would have spurred these firms to rely more on software patents.[47] They found that this group of firms did increase its patenting relative to other software firms; these firms also increased their R&D spending by a small amount.[48] While both of these studies are consistent with the idea that software patents affected R&D spending (in one case negatively, in the other, positively), neither can completely exclude other explanations for the observed changes.[49]

Patents can also provide disincentives for investing in innovation and they can impose social costs. Some researchers have studied the role of "patent

[42] For a discussion on the methodology of valuing patents based on patent maintenance fees, see BESSEN & MEURER, *supra* note 9, at 99-104.

[43] *Id.* at 143-44.

[44] Bronwyn H. Hall & Megan MacGarvie, *The Private Value of Software Patents*, 39 RES. POL'Y. 994, 1002-03 (2010).

[45] Bessen and Hunt, *supra* note 2, at 27-28.

[46] Josh Lerner & Feng Zhu, *What is the impact of software patent shifts? Evidence from Lotus v. Borland,* 25 INT'L J. IND. ORG. 511, 517-19 (2007).

[47] *Id.* at 517. Copyright protection of user interfaces did not preclude simultaneous patent protection, so presumably these firms would have found patents too costly to use prior to these court decisions.

[48] *Id.* at 528.

[49] Moreover, the Lerner-Zhu study has the difficulty that user interface features are rarely a substantial part of software products and user interface patents rarely comprise a large share of the patents obtained by software companies.

thickets".[50] When firms acquire large numbers of patents, they can restrict entry into an industry and they can use these patents to extract rents from other firms beyond the rents needed to encourage innovation. To the extent that startup firms are particularly important for innovation, patent thickets can reduce innovation by reducing entry. To the extent that patent thickets encourage firms to engage in socially unproductive business stealing activities, they can impose social costs. The empirical research, however, shows that it is difficult to disentangle the positive and negative incentives associated with patent thickets. For example, Cockburn and MacGarvie find that patent thickets do, indeed, restrict firm entry, but they find that a prospective entrant can reduce entry barriers by obtaining patents itself.[51]

Another disincentive arises from litigation risk.[52] Prospective innovators consider the risk of subsequent litigation when deciding whether to invest in research and development.[53] Ideally, innovators will conduct a patent clearance search and license any necessary patents. However, this might not be feasible because search costs might be too great, especially in fields like software where there are large numbers of patents.[54] Also, the boundaries of patents might not be predictable, so it might be difficult or impossible to determine what patents a prospective technology might infringe.[55] In these cases, the innovating firm faces a risk of inadvertent infringement, that is, risk of a future lawsuit that cannot be feasibly avoided up front. Prospective innovators consider this risk when making investment decisions. Since the cost of such lawsuits diminishes the profits from an innovation, this risk counts as a disincentive for investing in innovation. Note that relatively little patent infringement litigation appears to involve direct copying, so most patent litigation appears to be inadvertent infringement.[56]

[50] *See* Cockburn & MacGarvie, *supra* note 12*;* Mark A. Schankerman & Michael D. Noel, *Strategic Patenting and Software Innovation* CEPR Discussion Paper No. 5701 (2006), *available at* http://ssrn.com/abstract=922111.

[51] Cockburn & MacGarvie, *supra* note 12, at 768.

[52] *See* Bessen & Meurer, *supra* note 9, at 120-46.

[53] *Id*. at 130.

[54] *Id*. at 213 ("One software executive estimates that checking clearance costs about $5,000 per patent" but that there may be thousands of patents to be checked.).

[55] Oftentimes, the scope of the patent's claims does not clearly delineate the rights the inventor has over the invention. *See id*. at 46-72 (chapter discussing the boundary problems with patents).

[56] Christopher A. Cotropia & Mark A. Lemley, *Copying in Patent Law,* 87 N.C. L. Rev. 1421, 1457-58 (2009); Bessen & Meurer, *supra* note 9, at 126.

Bessen and Meurer find substantial risk associated with patent litigation generally.[57] Bessen and Meurer additionally compare the positive incentives patents provide public firms to invest in innovation (based on estimates of patent value) with the annual risk of litigation to these firms.[58] For firms in the chemical and pharmaceutical industries, the positive incentives substantially outweigh the disincentives from litigation through 1999.[59] However, for firms in other industries, the litigation risk substantially exceeded the positive incentives that patents provided by 1999, thanks to the dramatic rise in patent litigation beginning during the mid-1990s.[60]

Software patents play an important role in this litigation, accounting for nearly a quarter of the lawsuits by the end of the 1990s.[61] Software patents played this enhanced role because these patents are much more likely to be litigated than other patents. Bessen and Meurer find that software patents are nearly five times as likely to be in a lawsuit than are chemical patents; business method patents are nearly fourteen times as likely to be litigated.[62] Lerner finds that financial patents are from 27 to 39 times more likely to be litigated than are other patents.[63] Bessen and Meurer explore the reasons for these high litigation rates, attributing it largely to the unpredictable boundaries of these patents.[64] For example, software patent decisions in district court are much more likely than other patents to be appealed over issues of claim construction.[65] While software firms had to defend against somewhat fewer lawsuits during the 1990s than firms in manufacturing industries, the rate of their lawsuits was increasing more rapidly.[66]

To summarize the literature, in the 1990s, the number of software patents granted grew rapidly, but these were acquired primarily by firms outside the

[57] *See* James Bessen & Michael J. Meurer, *The Private Costs of Patent Litigation*, (BOS. U. SCH. OF L., Working Paper No. 07-08, 2007), *available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=983736.

[58] BESSEN & MEURER, *supra* note 9, at 130-46.

[59] *Id.* at 139.

[60] *Id.* at 140.

[61] *Id.* at 143.

[62] *Id.* at Table 9.1, 189.

[63] Josh Lerner, *The Litigation of Financial Innovations*, 53 J. L. & ECON 807, 808 (2010).

[64] BESSEN & MEURER, *supra* note 9, at 18.

[65] *Id.* at 153.

[66] *See* James Bessen & Michael J. Meurer, *What's Wrong with the Patent System? Fuzzy Boundaries and the Patent Tax*, 12 FIRST MONDAY, 6 (2007), http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1867/1750.

software industry and perhaps for reasons other than to protect innovations. Relatively few software firms obtained patents in the 1990s and so, it seems that most software firms did not benefit from software patents. More recently, the majority of venture-backed startups do seem to have obtained patents. The reasons for this, however, are not entirely clear and so it is hard to know whether these firms realized substantial positive incentives for investing in innovation from patents. On the other hand, software patents are distinctly implicated in the tripling of patent litigation since the early 1990s. This litigation implies that software patents imposed significant disincentives for investment in R&D for most industries including software.

It is hard to conclude from the above findings that software patents significantly increased R&D incentives in the software industry. However, this poor performance might have arisen from problems of "growing pains" with this new subject matter. I next look at some more recent statistics to assess whether software firms today realize greater benefits and reduced disincentives from software patents.

III. DATA

A. *Data sources*

This analysis uses patent data and litigation data. The patent data come from the Patent Data Project of the National Bureau of Economic Research.[67] This project covers patents granted from 1976 through 2006.[68] An extensive effort was put into matching patent assignees to Compustat data for publicly listed firms, matching first by algorithm and then by manual inspection.[69] This procedure also incorporated merger and acquisition data so that ownership of a patent could be tracked as the original assignee was acquired or spun off.[70] Tests performed on subsamples verify a high quality of match with relatively small numbers of false positives and false negatives.[71]

I used the USPTO's technology classes to identify software patents.[72] I

[67] PATENT DATA PROJECT, http://sites.google.com/site/patentdataproject (last visited June 8, 2011). *See also* JAMES BESSEN, NBER PDP PROJECT USER DOCUMENTATION: MATCHING PATENT DATA TO COMPUSTAT FIRMS (2009), *available at* http://www.nber.org/~jbessen/matchdoc.pdf.

[68] PATENT DATA PROJECT, *supra* note 63; BESSEN, *supra* note 63.

[69] BESSEN, *supra* note *63*.

[70] *Id*.

[71] Tests were performed by a group I participated in at NBER.

[72] *See* USPTO BULK DOWNLOADS: PATENT CLASSIFICATION INFORMATION, GOOGLE

obtained primary and secondary class data for each patent from the U.S. Patent Grant Master Classification File and the U.S. Patent Application Master Classification File from the USPTO.[73]

The litigation data come from Derwent's Litalert database for lawsuits filed through 2009.[74] For each lawsuit only the record of the initial filing was kept to avoid double counting (subsequent records referred to subsequent actions in the case). All of the patents listed in each case were counted.

Derwent does not capture all lawsuit filings.[75] To correct for undercounting, I compared the total counts of lawsuit filings in the Derwent data to the totals reported by the Administrative Office of the U.S. Courts. On average, Derwent reported only 69% of the total lawsuits for the fiscal years 1984 through 2008.[76] To correct for undercounting in several of the figures below, I divided the tabulated number of suits by 0.69.[77] Finally, a significant number of patent lawsuits have been filed in recent years concerning false marking, where products listed patent protection for patents that had expired.[78] However, these lawsuits only accounted for 1 % of all patent lawsuits in 2009, so I made no correction for these lawsuits.[79]

B. What is a software patent?

In order to count software patents, it is necessary to identify them. Conceptually, the goal is to select patents that use a logic algorithm for processing data that is implemented via stored instructions residing on a disk

PATENTS BETA http://www.google.com/googlebooks/uspto-patents-class.html (last visited June 8, 2011).

[73] *Id.*

[74] DERWENT LITALERT, *available at* http://www.westlaw.com.

[75] *Id.* (Derwent data includes "records for patent and trademark litigation lawsuits filed in ninety-four U.S. District courts that have been reported to the Commissioner of the United States Patent and Trademark Office (USPTO). Also included are records for thousands of lawsuits filed since the early 1970's that have never been published in the Official Gazette").

[76] James Bessen & Michael J. Meurer, "The Patent Litigation Explosion," BU School of Law Working Paper #05-18 (2005) at 11; updated figures through 2008 are available from the author.

[77] It is possible that Derwent misses fewer lawsuits in the most recent years. If so, then the figures for the number of lawsuits and probability of lawsuits might be slightly overstated for these years. Using a year-by-year adjustment for undercounting does not change the substantive conclusions below. *Id.*

[78] FALSE PATENT MARKING, http://www.falsemarking.net/ (last visted June 8, 2011).

[79] *Id.*

or other storage medium or in read-only memory. Additionally, at least some novel aspect of the invention should reside in the software. Since 1981 in *Diamond v Diehr*, patent protection has not been controversial for inventions that use software in the implementation of an otherwise patentable product or process.[80]

In practice, researchers have identified software patents by two main methods: using keyword searches and/or using patent office technology classes. John Allison individually read a number of software patents for several studies; however, that approach is not suitable for the entire sample of software patents.[81] Moreover, the patents that Allison identified by this method largely overlap those identified by more automated methods.[82]

Initially, there was some controversy over which methods were best.[83] However, further study revealed that for the kinds of analysis done here, similar qualitative results are obtained using different selection methods.[84]

For this study, I used a simple selection based on USPTO technology classes that are titled data processing (classes 700-707 and 715-717) and several other classes that are reliant on software and in which software companies obtain patents (341, coded data generation or conversion; 345, computer graphics processing; 370, multiplex communication; 375, digital communications; 380, cryptography; 381, audio signal processing; 382, image analysis; 726, information security; and 902, electronic funds transfer). I used the patent classification as of December 28, 2010 (the USPTO regularly reclassifies

[80] *See* Diamond v. Diehr, 450 U.S. 175 (1981) ("[W]hen a claim containing a mathematical formula implements or applies that formula in a structure or process which, when considered as a whole, is performing a function which the patent laws were designed to protect (e.g., transforming or reducing an article to a different state or thing), then the claim satisfies the requirements of § 101.").

[81] In both studies, John Allison and his co-authors collected a random sample of 1000 utility patents for their studies. *See* John R. Allison & Mark A. Lemley, *Who's Patenting What? An Empirical Exploration of Patent Prosecution*, 53 VAND. L. REV. 2099, 2108 (2000); JOHN R. ALLISON & EMERSON H. TILLER, *Internet Business Method Patents*, *in* PATENTS IN THE KNOWLEDGE-BASED ECONOMY 259, 261 (Wesley M. Cohen & Stephen A. Merrill eds., 2003).

[82] *See* Hunt, *supra* note 8, at 163-4.

[83] *See* Robert W. Hahn & Scott Wallsten, *A Review of Bessen and Hunt's Analysis of Software Patents* (2003), *available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=467484. *See also* http://en.wikipedia.org/wiki/Bessen/Hunt_technique (discussing the Bessen/Hunt technique in identifying software patents).

[84] *See* Hunt, *supra* note 8, at 165. *See also* Hall, *supra* note 40, at 999.

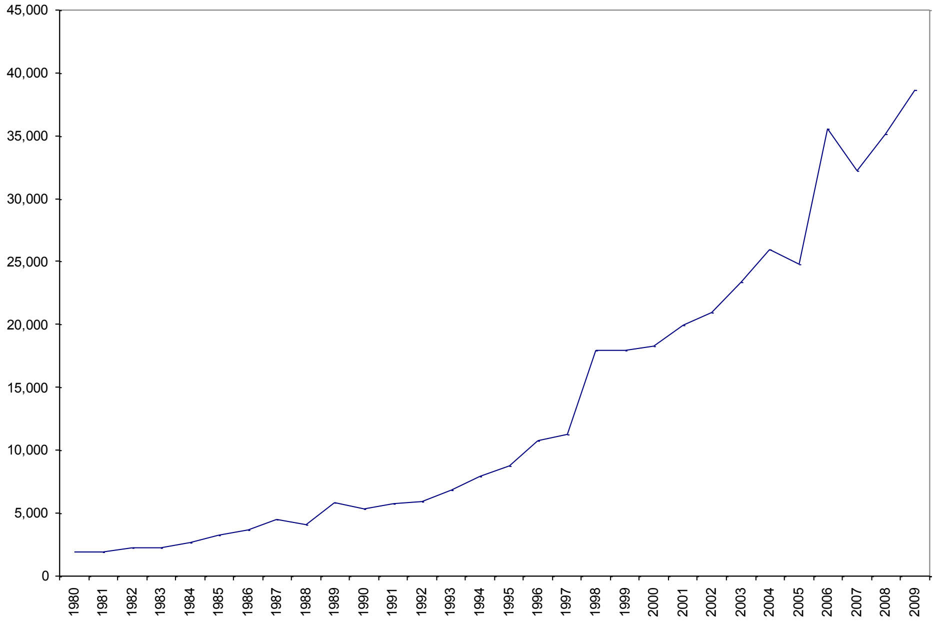
patents).

IV. SOFTWARE PATENTS OVER THE LAST DECADE

A. *Overview*

Figure 1 shows that the number of software patents granted per year has continued to increase dramatically. This growth has been faster than the growth in total patent grants, so that software patents account for a growing share of total patent grants; they make up about one quarter of all patent grants today.[85] Figure 2 shows software patent grants and total patent grants on a logarithmic scale. Both series closely follow a linear trend, suggesting that the exponential growth rates of both series have remained roughly constant, but software patent grants grow at a substantially faster rate.

Figure 1. Annual grants of US software patents.



[85] The USPTO reports 167,349 utility patent grants for 2009. USPTO, US Patent Statistics Chart, Calendar Years 1963-2011, http://www.uspto.gov/web/offices/ac/ido/oeip/taf/us_stat.htm. The data in Figure 1 show 38,654 software patents granted in 2009.

Note: Grants to patent classes for data processing, classes 700-707 and 715-717, 341, coded data generation or conversion, 345, computer graphics processing, 370, multiplex communication, 375, digital communications, 380, cryptography, 381, audio signal processing, 382, image analysis, 726, information security, and 902, electronic funds transfer.

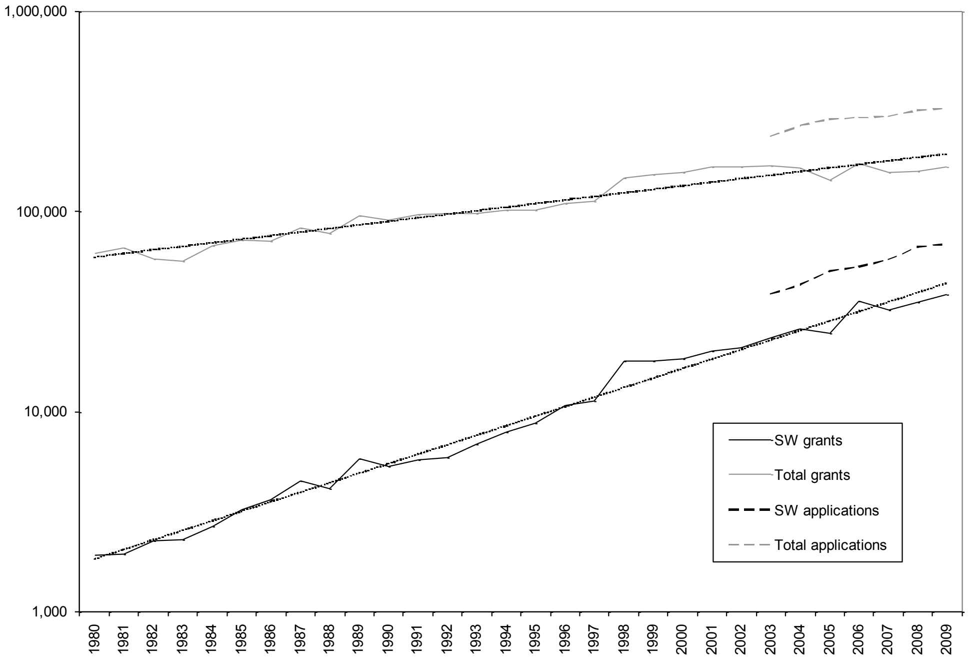
Because these trends are roughly constant, it seems unlikely that software patenting behavior has changed much in response to changes in the patent examination process or changes in the law from recent court decisions. However, the patent grant rate is a product of the rate of patent applications and the rate of patent allowances, after some delay (and that delay has grown in recent years).[86] It is possible that patent application trends and patent examination trends might offset each other. For example, perhaps fewer software patent applications have been approved, but the number of software patent applications has increased. Figure 2 also shows the number of published patent applications, for both total patents and software patents for recent years.[87] There does not appear to be any dramatic divergence between the series for patent grants and patent applications for software patents. Moreover, total patent applications have grown only slightly faster than total patent grants, which has resulted in growing delay between application and grant. But little here suggests any dramatic change in behavior of either patent applicants or patent examiners especially in regard to software patents.[88]

[86] Dennis Crouch & Jason Rantanen, *Unreasonable Patent Applicant Delay and the USPTO Backlog*, PATENTLY-O BLOG (Jul. 9, 2010), http://www.patentlyo.com/patent/2010/07/unreasonable-patent-applicant-delay-and-the-uspto-backlog.html.

[87] Only about 71% of patent applications are published, but that share has been roughly constant since 2003. Software patents have a longer pendency (delay between application and grant) than many other technologies, but pendency for other technologies appears to be growing slightly faster in recent years.

[88] Some members of the patent bar have claimed that court decisions such as that in *KSR v Teleflex* have so weakened the value of patent protection that patent applications have dropped dramatically. In fact, any such changes appear to be little more than temporary blips in these series.

Figure 2. Grants and published applications of software patents and all patents (logarithmic scale)



Note: software patents identified as in previous figure. Applications are all published applications. Published applications account for 71% of total applications from 2003 through 2009 with minor variation.

B. *Do software firms now get more patents?*

Table 1 shows the share of firms listed on US stock exchanges that have any patents. The share of firms with patents in the narrowly defined “Prepackaged software” industry (SIC 7372) increased from 24 % in 1996 to 33 % in 2006.[89] In the broader industry classification of “Computer Programming, Data Processing, And Other Computer Related Services” (SIC 737), which includes service and Internet companies, the share of firms with patents increased from

[89] The industry classification refers to the firm’s primary line of business as determined by Compustat.

20 % to 27 %. Although these increases are significant, it is still true that most software firms do not patent at all.

Table 1. Share of publicly listed software firms with any patents

	1996 Share	Total firms	2006 Share	Total firms
Pre-packaged software (SIC 7372)	24.2%	495	33.2%	358
Startup firms only	19.8%	369	12.3%	57
Computer services and software (SIC 737)	20.1%	922	26.8%	730
Startup firms only	16.8%	647	13.8%	138
<u>Addendum:</u>				
Venture-backed software startups (Mann & Sager 2005)	24%			
All software startups (Graham et al. 2009)			24%	
Venture-backed software startups (Graham et al. 2009)			67%	

Public listed startups are firms that have been publicly listed on US exchanges for less than 5 years. Source: NBER Patent Data Project.

This seems to be even more true for startup firms, here defined as firms that have been publicly listed for less than 5 years. The share of startup firms in the prepackaged software industry with patents declined from 20 % in 1996 to 12 % in 2006. In the broader software industry (SIC 737), the share declined from 17 % to 14 %. Thus even fewer startup firms in 2006 had obtained any patents, and the share that had appears to have declined. These shares reported for publicly listed startups are slightly smaller than the shares of early stage software startups with patents reported in the previous literature, with the exception of the Berkeley survey result for venture capital backed software startups, which had a much higher level of patenting. I will discuss this difference below.

The large number of software patent grants and the small share of software firms obtaining patents imply that software firms account for relatively little of the activity in software patenting. This intuition is verified in Table 2, which shows that the broad software industry (SIC 737) accounted for only 11 % of software patent grants to public firms in 1996 and 17 % in 2006. The prepackaged software industry accounts for 2.8 % and 9.8 % in those years respectively. Thus, the software industry still accounts for a small portion of software patent grants, although that portion has increased over the last decade. Most software patents still go to non-software firms.

Table 2. Industry share of software patents granted to publicly listed firms

	1996	2006
Prepackaged software industry (SIC 7372)	2.8%	9.8%
Top 10 firms	2.1%	8.0%
Computer services and software (SIC 737)*	11.4%	17.2%
Addendum:		
Top 10 firms' share of software patents granted to prepackaged software industry	75%	81%
Number of public firms in prepackaged software	495	358

*Excluding IBM.

Moreover, the increase in the share of software patents granted to software firms is largely accounted for by the activity of a small number of large software firms. Table 3 lists the patents granted to the top 10 recipients in the prepackaged software industry for each year. These firms increased their patenting by an order of magnitude, and this accounts for most of the increase in patents going to the software industry as shown in Table 2. These few large firms account for most of the software patents granted to the software industry (75 % and 81 % in 1996 and 2006 respectively).

Table 3. Number of patents granted to the top 10 publicly listed (in the US) recipients in the software industry (SIC 7372)

1996		2006	
97	Microsoft Corp	1461	Microsoft Corp
20	Borland Software	178	Oracle Corp
13	Intergraph Corp	116	Cadence Design
12	Adobe Systems Inc	76	Digimarc Corp
11	Wang Labs Inc	74	National Instruments
9	National Instruments	59	Adobe Systems Inc
9	Cadence Design	56	SAP AG
8	Oracle Corp	44	Synopsys Inc
6	3dO Co	37	Autodesk Inc
6	Sybase Inc	37	BEA Systems Inc

Note: the table lists all patents granted to these firms; the totals for those classified as software patents is slightly smaller.

To summarize, most software firms still do not patent, although the percentage has increased. And most software patents go to firms outside the software industry, despite the industry's substantial role in software innovation. While the share of patents going to the software industry has increased, that increase is largely the result of patenting by a few large firms.

C. *Are software patents important for financing startups?*

As noted above, one important change over the last decade is that the majority of venture-back software startups now do appear to get some patents[90], while the majority did not during the 1990s.[91] But does this represent a change in the benefits that these firms are receiving from patents or does it, instead, represent a change in the behavior of venture capitalists?

Some people argue that patents provide a real benefit to startups because they signal venture capitalists about the quality of the startup's technology.[92] The argument goes that startups have asymmetric information about the quality of their technology that they cannot credibly communicate to investors.[93] A

[90] *See* Graham, *supra* note 29, at 1276-7.

[91] *See* Mann, *supra* note 8, at 964.

[92] *See* Long, *supra* note 33, at 637.

[93] *Id*.

patent is supposed to provide a signal of technology quality or of the quality of firm management.[94] Patents are thus supposed to provide an important benefit to startup firms by facilitating investment.

But this sort of signaling differs from the usual discussion of signaling in the economics literature.[95] In economic theory, signaling requires that agents holding private information engage in a costly activity. Only high quality types will be willing to make a substantial investment, so investors can conclude that firms that do invest are of higher quality. If low quality types can afford to make the investment, then they can send the signal as well. However, filing a patent application is not particularly costly, especially for a software patent — software patents might not require significant development, often requiring little other than a high level description of an idea. Without significant cost, the patent cannot serve to reveal private information by separating high quality firms from low quality ones.

Nor does the patent examination serve to screen out low quality technologies. This is so for two reasons. First, the patent examiner typically has access to much less information about the technology than does a venture capitalist. For example, in the author's experience, venture capitalists routinely look at product source code as part of their due diligence while source code is not usually provided in a patent application. Second, the patent examination is concerned with other issues than the commercial quality of the technology.

In any case, the data in Table 2 shed some light on whether patents facilitate financing. This is because public investors also face asymmetric information about the quality of the technology. Financial analysts typically have far less access to private company information about the technology than venture capitalists have. This means that if patents signal high quality technology to venture capitalists, this signal should be even more important to stock market investors. However, Table 2 shows that most public software startups do not obtain patents. This means that patents cannot provide a signal of quality to them.

And this suggests that the change in patenting over the last decade for venture-backed software startups might have more to do with the changing behavior of venture capitalists rather than changing benefits for software startups. For example, venture capitalists might be interested in the salvage value that patents provide in the frequent cases where the startups fail. Because

[94] *Id.*

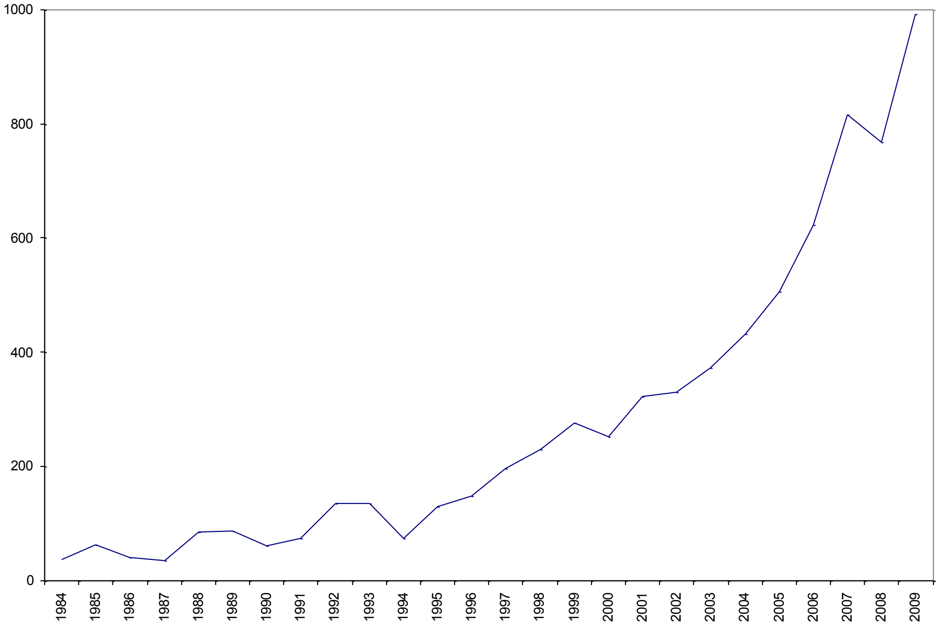
[95] Thanks to Mike Meurer for this argument. The economics of signaling began with Michael Spence. *See* Michael Spence, *Job Market Signaling* 87 QUARTERLY JOURNAL OF ECONOMICS 355 (1973).

a more robust market for selling patents has developed over the last decade, venture capitalists can now recoup some of their investment by selling the patents that a failed firm acquired.[96] In this case, some portion of the benefits that venture capitalists receive might flow back to the startup firm in the form of reduced financing costs. However, given that the mean value of patents sold at auction seems to be little more than patent prosecution costs[97], these benefits would seem to be small.

D. *Has the litigation risk from software patents abated?*

Figure 3 shows the number of patent lawsuits (infringement and declaratory actions) involving software patents that are filed each year, corrected for undercounting of the Derwent data. Clearly, the number of software patent lawsuits has continued to grow rapidly, meaning that the risk of litigation from software patents has necessarily increased.

Figure 3. Number of patent lawsuit filings involving software patents

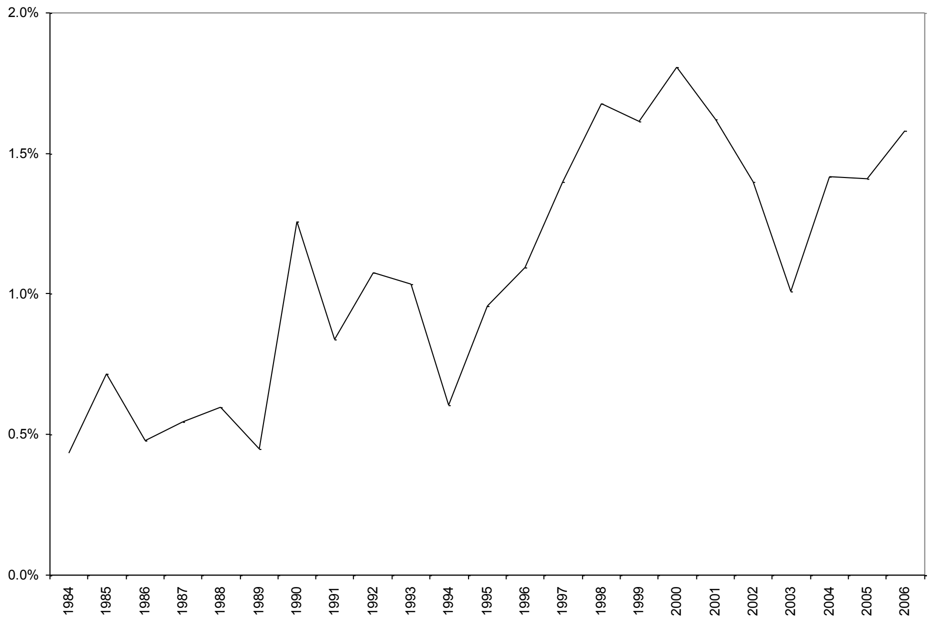


[97] Bessen & Meurer, *supra* note 9, at 140, 181 ("On April 6, 2006, Ocean Tomo, an intellectual property merchant banc held the first live patent auction . . . The mean value sold at the auction was only $29,000" while the legal cost of patent prosecution calculated by Mark Lemley was $20,000 per application.)

Note: this chart is based on Derwent Litalert data corrected for undercounting as described in the text.

Of course, the number of software patent lawsuits is a product of the number of software patents times the probability a software patent will be in a lawsuit. The number of software patents has been increasing rapidly, but how has the probability of a lawsuit changed for software patents? Figure 4 looks at the probability that a newly issued software patent will be in a lawsuit within four years of issue. At the very least, the probability that a software patent is in a lawsuit has stopped increasing and it might very well have begun a decreasing trend. This series provides some preliminary indications of a positive trend, although this is a noisy time series, and it only looks at the first four years of the patent life. Moreover, the court decisions affecting software patents beginning in 2007 could explain at least part of this reversal. Nevertheless, given the rapid growth in software patent grants, the aggregate litigation risk from software patents continues to grow rapidly and a much more dramatic change would be required to reverse that trend.

Figure 4. Probability that a software patent will be in a lawsuit within four years of issue



Note: Using Derwent Litalert data for patent lawsuits corrected for undercounting as described in the text.

Finally, note that despite the decline in the probability of litigation, this probability is still well above the levels of the late 1980s and early 1990s.[98]

V. CONCLUSION

Has the patent system adapted to software patents so as to overcome initial problems of too little benefit for the software industry and too much litigation? The evidence makes it hard to conclude that these problems have been resolved. While more software firms now obtain patents, most still do not, hence most software firms do not directly benefit from software patents. Patenting in the software industry is largely the activity of a few large firms. These firms realize benefits from patents, but the incentives that patents

[98] *Id*. at 153 (The probability of software patent in a suit in the late 80s and early 90s was 4.6%).

provide them might well be limited because these firms likely have other ways of earning returns from their innovations, such as network effects and complementary services. Moreover, anecdotal evidence suggests that some of these firms patent for defensive reasons, rather than to realize rents on their innovations: Adobe, Oracle and others announced that patents were not necessary in order to promote innovation at USPTO hearings in 1994, yet they now patent heavily.[99]

On the other hand, the number of lawsuits involving software patents has more than tripled since 1999. This represents a substantial increase in litigation risk and hence a disincentive to invest in innovation. The silver lining is that the probability that a software patent is in a lawsuit has stopped increasing and might have begun a declining trend. This occurred perhaps in response to a new attitude in the courts and several Supreme Court decisions that have reined in some of the worst excesses related to software patents.

This analysis only concerns the software industry. It is possible, of course, that software patents might be highly beneficial to the various hardware industries that obtain large numbers of software patents. Clearly software patents are *privately* beneficial in these industries – that is why firms acquire so many of them. However, this does not mean that there are corresponding social benefits. For example, this patenting might be aimed at building large strategic portfolios that facilitate business stealing (e.g., allowing portfolio holders to collect royalties from new, innovative firms) without increasing the level of innovation. Only further study can tell.

Nevertheless, if software patents were socially beneficial, this should show up in the evidence from the software industry. In this regard, it is notable that after more than a decade of experience, this economic experiment played out in a highly innovative industry still lacks clear evidence of net benefit.

[99] *See* Table 3.