Washington University Law Review

Volume 85 | Issue 2

2007

The Disputed Quality of Software Patents

John R. Allison
University of Texas, Austin, Texas

Ronald J. Mann
Columbia Law School

THE DISPUTED QUALITY OF SOFTWARE PATENTS

JOHN R. ALLISON
RONALD J. MANN*

TABLE OF CONTENTS

TABLES AND FIGURES

---

* Allison is the Spence Centennial Professor in the McCombs School of Business and Professor of Intellectual Property Law at the McCombs School and College of Engineering, University of Texas at Austin. Mann is Professor of Law at Columbia Law School. We thank Abe Dunn and An-Shih Liu for statistical analysis.

ABSTRACT

We analyze the characteristics of the patents held by firms in the software industry. Unlike prior researchers, we rely on the examination of individual patents to determine which patents involve software inventions. This method of identifying the relevant patents is more laborious than the methods that previous scholars have used, but it produces a data set from which we can learn more about the role of patents in the software industry. In general, we find that patents the computer technology firms obtain on software inventions have more prior art references, claims, and forward citations than the patents that the same firms obtain on nonsoftware inventions. We also find that the patents that "pure" software firms (those producing only software) obtain on software inventions have more prior art references, claims, and forward citations than the software patents obtained by the firms that derive revenues from other product lines. Finally, we conclude that the patents of the largest firms are no better (or worse) than the patents of the smallest firms, belying the idea that large firms are plagued by challenges based on the worthless patents of their smaller competitors.

The Article closes with a brief discussion of the implications of our empirical analysis. The findings undermine the strongest criticisms about the low quality of software patents. It is simply not accurate to say that software patents as a group have remarkably low numbers of prior art references and forward citations. Thus, these findings cut against technology-based patent reforms designed to make it more difficult to obtain software patents. On the other hand, the evidence that small firms are no less capable than large firms of producing quality patents undermines concerns that higher hurdles at the early stage of the patenting process would disadvantage smaller inventors.

I. INTRODUCTION

As the use of patents has become commonplace in the software industry, concerns about their propriety have remained surprisingly strong. It is easy to understand why those who were late to develop effective patenting strategies would oppose software patents in the mid-1990s, when patenting began to spread broadly through the industry.[1] However, despite the widespread use of patenting in the modern software industry, the concerns about software patents have continued. For example, the Federal Trade Commission's (FTC) October 2003 report, *To Promote Innovation*, summarizes hearings in which "[m]any panelists and participants expressed the view that software and Internet patents are impeding innovation."[2] As technologist Hal Varian proclaims, there has been "a steady reduction in patent quality, with patents of dubious novelty being granted routinely."[3] Similarly, the National Research Council decries events that are "degrading the quality of" new patents, especially in high technology industries.[4] Large firms in particular have complained about the poor quality of the patents being asserted against them in litigation.[5] Most recently, a 2006 Brookings Institution Report authored by Doug Lichtman identifies problems in ensuring patent quality as one of the principal justifications for a proposal to abandon the legal presumption of patent validity.[6]

1. *See* John R. Allison, Abe Dunn & Ronald J. Mann, *Software Patents, Incumbents, and Entry*, 85 TEX. L. REV. 1579 (2007).

2. FEDERAL TRADE COMMISSION, TO PROMOTE INNOVATION: THE PROPER BALANCE OF COMPETITION AND PATENT LAW AND POLICY ch. 3 at 56 (Oct. 2003), *available at* http://www.ftc.gov/os/2003/10/innovationrpt.pdf.

3. Hal R. Varian, *Economic Scene; A Patent That Protects a Better Mousetrap Spurs Innovation, But What About One for a New Way to Amuse a Cat?*, N.Y. TIMES, Oct. 21, 2004, at C2, *available at* 2004 WLNR 5608092.

4. NAT'L RESEARCH COUNCIL OF THE NAT'L ACADS., A PATENT SYSTEM FOR THE 21ST CENTURY 1, 46–51 (2004). A scholarly volume that accompanies this report emphasizes concerns that "a weakening of the standards of novelty and non-obviousness [has] undermin[ed] the purpose of patents to provide an incentive to those who innovate in a genuine way." PATENTS IN THE KNOWLEDGE-BASED ECONOMY 2 (Wesley M. Cohen & Stephen A. Merrill eds., 2003).

5. *See, e.g.*, *Patent Quality Improvement: Hearing Before the Subcomm. on Courts, the Internet, and Intellectual Property of the H. Comm. on the Judiciary*, 109th Cong. 18 (2005) (statement of Richard J. Lutton, Jr., Chief Patent Counsel, Apple) ("The current patent system has given rise to too many low quality patents being issued, and a growing pattern of assertions of weak patents that threaten to damage productive companies and stifle innovation."). *But see* INTELLECTUAL PROP. OWNERS ASSOC., IPO SURVEY: CORPORATE PATENT QUALITY PERCEPTIONS IN THE U.S. (2005) (reporting a survey in which most respondents believe patent quality is unsatisfactory or poor, more believe that patent quality will continue to deteriorate than expect it to improve, and the overwhelming majority believe that litigation costs will increase in the next three years).

6. Doug Lichtman, *Aligning Patent Presumptions with the Reality of Patent Review: A Proposal*

Although a good deal of the public criticism can be dismissed as self-serving efforts to promote the interests of particular sectors of the industry,[7] two distinct variants of the criticism are substantial. The first is the idea that software patents impede innovation because they hinder entry by new firms.[8] The second is closely related to the first but reflects a distinct concern about the craft and effort reflected in the patents themselves, rather than the competitive structure of the industry in which they are deployed.[9] Indeed, the seriousness of this concern is underscored by the central role of the software industry in the Community Patent Review Initiative[10] and in such initiatives as IBM's effort to develop a "Patent Quality Index."[11]

There has been a great deal of writing on the first subject, including some sophisticated econometric analysis on the relation between patenting and innovation and competition in the software industry.[12] There has been

for Patent Reform 5 (The Brookings Inst., Hamilton Project, Discussion Paper No. 2006–10, 2006).

7. *See, e.g.*, Matthew Broersma, *Stallman: OSDL Patent Project "Worse Than Nothing*," Sept. 19, 2006, http://news.com/2102-7344_3-6117131.html (discussing the view of Free Software Foundation founder Richard Stallman that all software patents are inappropriate). For a thorough discussion of the relation between patent strategy and policy perspectives of software firms, see Allison, Dunn & Mann, *supra* note 1.

8. *See, e.g.*, PATENTS IN THE KNOWLEDGE-BASED ECONOMY, *supra* note 4, at 2.

9. *See, e.g.*, David Evans and Anne Layne-Farrar, *Software Patents and Open Source: The Battle over Intellectual Property Rights*, 9 VA. J.L. & TECH. No. 10 ¶ 25 (2004) (summarizing complaints about "trivial" software patents); Mark H. Webbink, *A New Paradigm for Intellectual Property Rights in Software*, 12 DUKE L. & TECH. REV. ¶¶ 5–8 (2005) (complaining about trivial and invalid software patents); James Bessen & Eric Maskin, *Sequential Innovation, Patents, and Imitation* 20 (Mass. Inst. of Tech. Dep't of Econ., Working Paper No. 00-01, 2000) (arguing that "software patents have been too broad and too obvious"); John Kasdan, *Obviousness and New Technologies* (Columbia Univ. Sch. of Law, Columbia Law & Economics Working Paper No. 146, 1999); *see also* John H. Barton, *Reforming the Patent System*, 287 SCIENCE 1933 (2000) (criticizing bad patents as a problem that pervades all of the high-technology sectors); Glynn S. Lunney, Jr., *E-Obviousness*, 7 MICH. TELECOMM. & TECH. L. REV. 363 (2001) (criticizing the "potential monopoly costs of unjustified patents" because of a general decline in the doctrine of obviousness).

10. *See* Beth Simone Noveck, "*Peer to Patent": Collective Intelligence, Open Review and Patent Reform*, 20 HARV. J.L. & TECH. 123 (2006).

11. *See* Jennifer LaClaire, *IBM Teams with OSDL, USPTO on Patent Quality Initiative*, E-COMMERCE TIMES, Jan. 10, 2006, http://www.ecommercetimes.com/story/48212.html. IBM's senior vice president of technology and intellectual property explained IBM's motivation as follows: "[r]aising the quality of patents will encourage continued investment in research and development by individual inventors, small businesses, corporations and academic institutions while helping to prevent over-protection that works against innovation and the public interest." *Id.*

12. For a selection of the most recent papers of interest, see MICHAEL NOEL & MARK SCHANKERMAN, STRATEGIC PATENTING AND SOFTWARE INNOVATION (2006) (analyzing whether patenting by software firms has positive or negative spillovers). Professor Mann has also individually written about this subject, providing a mix of qualitative interviews and empirical analysis of patenting practices. *See* Ronald J. Mann, *Do Patents Facilitate Financing in the Software Industry?*, 83 TEX. L. REV. 961 (2005) [hereinafter Mann, *Software Patents*]; Ronald J. Mann & Thomas W. Sager, *Patents, Venture Capital, and Software Startups*, 36 RES. POL'Y 193 (2007); Iain M. Cockburn & Megan J.

much less writing about the second subject. Given the concern that an unduly lax standard for issuing patents will have an adverse effect on economic growth,[13] as well as the importance of the software sector to our economy, questions about the quality of such a rapidly growing type of patent are important. Thus, although Stuart Graham and David Mowery's work includes a preliminary assessment of the frequency with which subsequent patentees cite the patents owned by large software firms,[14] there is much more to be done.

The problem becomes even more urgent as policymakers in all three branches seriously consider reforms to the patent system that are justified for the most part by anecdotal complaints about the system rather than actual analysis of software patents as a group. For example, the 109th Congress considered a series of bills proposing major changes to patent procedures designed to respond to a perceived problem with quality.[15] The PTO has its own *Strategic Plan 2007–2012*, which advocates a major reallocation of priorities and resources to satisfy the primary goal of

---

MacGarvie, *Entry, Exit and Patenting in the Software Industry* (Nat'l Bureau of Econ. Research, Working Paper No. 12563, 2006) (analyzing the effects of patenting rates on entry and exit into sectors of the software industry); Bronwyn H. Hall & Megan MacGarvie, *The Private Value of Software Patents* (Nat'l Bureau of Econ. Research, Working Paper No. 12195, 2006) (analyzing the effect of software patents on the value of the firms that hold them).

13. Nancy Gallini & Suzanne Scotchmer, *Intellectual Property: When Is It the Best Incentive System?* (University of California at Berkeley, Economics Working Papers E01-303, 2001); Bronwyn H. Hall & Megan MacGarvie, *Business Method Patents, Innovation, and Policy* (Nat'l Bureau of Econ. Research, Working Paper No. 9717, 2006).

14. Stuart J.H. Graham & David C. Mowery, *Software Patents: Good News or Bad News?* (Geor. Inst. of Tech., T1:GER Working Paper, 2004) [hereinafter Graham & Mowery, *Software Patents*]. For careful case studies analyzing the quality of particular software patents, see Martin Campbell-Kelly, *Not All Bad: An Historical Perspective on Software Patents*, 11 MICH. TELECOMM. & TECH. L. REV. 191, 214–15 (2005); Martin Campbell-Kelly & Patrick Valduriez, *A Technical Critique of Fifty Software Patents*, 9 MARQ. INTELL. PROP. L. REV. 241 (2005).

15. *See, e.g.*, Patent Reform Act of 2005, H.R. 2795, 109th Cong. §§ 9–10 (as proposed by H. Comm. on the Judiciary, Sept. 15, 2005) (altering postgrant procedures and easing rules for pregrant third-party submissions, both to respond to perceived quality problems); *see also* Patent Reform Act of 2006, S. 3818, 109th Cong. §§ 6–7. The concerns about quality are most evident at the hearings on those bills. *See, e.g., Amendment in the Nature of a Substitute to H.R. 2795, The Patent Act of 2005: Hearing Before the Subcomm. on the Courts, the Internet, and Intellectual Property of the H. Comm. on the Judiciary*, 109th Cong. 7 (2005) (statement of Emery Simon, Counsel, Business Software Alliance) (contending that "the IT industry, like so many others, is encountering the enormous costs of dealing with patents of questionably quality" and offering as evidence the fact that "hundreds of patent infringement suits are pending against computer software and hardware companies, costing the industry hundreds of millions of dollars each year"); *Patent Act of 2005: Hearing Before the Subcomm. on the Courts, the Internet, and Intellectual Property of the H. Comm. on the Judiciary*, 109th Cong. 50 (2005) (statements of Mr. Griswold, Chief Intellectual Property Counsel, 3M Innovative Properties Co.) ("[O]ur solution[] is to start with the major problem. The problem is patents of low quality.").

"optimiz[ing] patent quality and timeliness."[16] Finally, and most visibly, the United States Supreme Court in the last few years has granted certiorari in an unusually large number of patent cases, motivated in part by concerns that excessive numbers of low-quality patents may be stifling innovation.[17]

This paper responds to that gap in the literature by providing detailed empirical analysis of the patents held by software firms. Specifically, we examine the roughly 34,000 patents held by the firms listed in a leading industry periodical during the five-year period from 1998–2002. We started by collecting a list of the firms from *Software Magazine*'s "Software 500." Relying on questionnaires disseminated by the magazine, that list indicates the top five hundred firms in the software industry each year by revenue from software and services. Based on industry interviews, we believe that the response rate is quite high. The list appears to be widely regarded as authoritative within the industry. Martin Campbell-Kelly, for example, uses the list pervasively in his comprehensive history of the industry.[18] It is, by way of comparison, considerably more comprehensive than the "Softletter 100" list (which is limited to prepackaged software providers) that Graham and Mowery use.[19] Because

16. *See* U.S. PATENT & TRADEMARK OFFICE, STRATEGIC PLAN 2007–2012, *available at* www.uspto.gov/web/offices/com/strat2007. A central concern of the plan is to improve the quality of patent applications because of the PTO's view that "[b]etter application input contributes directly to more efficient processing and to quality, thereby benefiting both the examiner and the applicant." *Patent Quality Enhancement in the Information-Based Economy: Hearing Before the Subcomm. on Courts, the Internet, and Intellectual Property of the H. Comm. on the Judiciary*, 109th Cong. (2006) (statement of the Honorable Jon W. Dudas, Director of United States Patent and Trademark Office). For empirical evidence assessing the extent of that problem, see John L. King, *Patent Examination Procedures and Patent Quality*, *in* PATENTS IN THE KNOWLEDGE-BASED ECONOMY, *supra* note 4, at 54.

17. This concern is most evident in Justice Kennedy's concurring opinion in *eBay v. MercExchange, L.L.C.*, 126 S. Ct. 1837, 1842–43 (2006) (concurring in decision that injunctions in patent infringement cases should not be automatic and rather based on traditional principles of equity, but expressing a belief that too many ill-advised patents had been granted on software-implemented business methods). It also appears to underlie the grants in cases such as *Lab. Corp. of Am. Holdings v. Metabolite Labs., Inc.*, 546 U.S. 999 (2005), *writ dismissed as improvidently granted*, 126 S. Ct. 2921 (2006) (granting certiorari to review whether method for detecting a deficiency of cobalamin or folate by observing correlation with elevated level of total homocysteine is patentable subject matter), and in *KSR Int'l Co. v. Teleflex, Inc.*, 126 S. Ct. 2965 (2006) (granting certiorari to review the standard for combining prior art references to determine whether a patent is invalid for obviousness); *see also* Microsoft Corp. v. AT&T Corp., 127 S. Ct. 467 (2006) (granting certiorari to review whether software can be a "component" of a patented invention supplied from the United States to a foreign country and, if so, whether *copying* the software in the foreign country amounts to "combining" it with other components to make an infringing invention in violation of 35 U.S.C. § 271(f)).

18. MARTIN CAMPBELL-KELLY, FROM AIRLINE RESERVATIONS TO SONIC THE HEDGEHOG: A HISTORY OF THE SOFTWARE INDUSTRY (2003).

19. Stewart J.H. Graham & David C. Mowery, *Intellectual Property Protection in the U.S.*

of considerable turnover in the industry, the "Software 500" list includes about 1,100 firms for the five years.[20] Of importance for our project, it extends from the largest firms in the industry (IBM and Microsoft were first and second throughout the five-year study period) to much smaller firms (iCIMS, Inc. was the smallest firm in 2002, with annual revenues of only $400,000). We then collected from Delphion a complete set of all of the 34,000 patents issued between January 1, 1998, and December 31, 2002, to each of the listed firms.

Our analysis proceeds in three steps. Part II addresses the threshold question of what exactly should count as a software patent. Unlike prior researchers, we examine the individual patents to determine which patents involve software inventions. This method of identifying the relevant patents is more labor intensive than the methods on which previous scholars have relied, but it produces a data set with a far lower error rate.

Part III considers what it means to discuss patent "quality" and "value." We build on a substantial body of literature that connects the value and quality of patents to objective features of the patents: the number of claims in the patents, the number of prior art references in the patents, and the number of forward citations (citations to a patent received in later patents).[21] We compare software patents to the nonsoftware patents obtained by the same firms, not only because our data set contains large numbers of both types of patents, but also because we believe that such a comparison may provide more direct insight into questions of relative quality and value than a comparison of software patents to a sample from the general population of patents. In general, we find that the patents firms obtain on their software inventions have more claims, cite more references, and are more frequently cited as prior art by later patents ("forward citations") than the patents the same firms obtain on their

Software Industry, *in* PATENTS IN THE KNOWLEDGE-BASED ECONOMY, *supra* note 4, at 219, 232. [hereinafter Graham & Mowery, *Intellectual Property Protection*].

20. Because the purpose of our study is to focus on firms that can be fairly characterized as software firms, we excluded the eighteen firms that did not derive at least 20% of their total revenues from software in any of the five years for which we collected data. The excluded firms are Cisco, Hitachi, Intel, NEC, Raytheon, Valassis, PreVision Marketing, VCON, Adaptec, Alstom ESCA, Amdahl, Brooktrout, Infolmage, International Network Services, Kasten Chase, MessageQuest, Template Software, and TYX.

21. There may be other indicators of effort on the part of the applicant, but these characteristics have been most prominent in the existing literature, in large part because they are readily accessible in an automated way from existing databases of patent information. Other indicators that might be probative (such as the number of references added by the examiner or adequate correspondence between the specification and claims) are not as easily assessed in a replicable and automated way. On January 1, 2001, the USPTO began identifying examiner-added prior art references on the face of the patent, but 60% of the patents in our data set were issued prior to that time.

nonsoftware inventions. We explain below the relationship of patent characteristics such as number of claims, prior art references, and forward citations to the concepts of patent quality and value. We also consider the possibility noted in the existing literature that the patents obtained by pure software firms would differ in quality and value from patents obtained by less specialized firms. We ultimately conclude that our data set supports that hypothesis.

Finally, Part IV closes with a brief discussion of the implications of our empirical analysis. First, we believe that the findings undermine the strongest criticisms about the low quality of software patents. Whatever one might glean from subjective analyses of the nature of the inventions disclosed by each patent, it is simply wrong to assert, for example, that software patents, as a group, fail in any notable way to mention the relevant prior art. Second, contrary to the idea that large firms are plagued by a mass of low-quality patents obtained by smaller firms, these findings suggest that there is no substantial difference between the patents obtained by the largest and most experienced patentees and those obtained by their less well-capitalized competitors. Collectively, we argue, these findings have important ramifications for patent reform, because they cut decisively against reforms focused on a particular area of technology. At the same time, to the extent they suggest that lack of resources is not a constraint on patent quality, they undermine the concern that reforms raising the bar for patent filings will operate to the particular detriment of smaller inventors.

II. WHAT IS A SOFTWARE PATENT?

This Article employs a definition of a software patent developed by one of the authors after several years of studying tens of thousands of computer-industry patents—a definition we believe to be the only one formulated thus far that can be used to identify software patents with principled consistency. We explain that definition after discussing several previous attempts to identify data sets of software patents for research purposes.

For various reasons, identifying a data set of software patents is a difficult task—what Professors Graham and Mowery have called the "thorniest" task for a scholar in this area.[22] First, there is no universally accepted definition of what a software patent is. Second, neither the U.S. Patent and Trademark Office (USPTO) classification system nor the

22. Graham & Mowery, *Software Patents*, *supra* note 14.

International Patent Classification (IPC) system was designed for that purpose; both systems focus on functionality at a low level of abstraction and are unsuitable for defining any technology area at a conceptual level. Third, even if these systems were suitable for identifying a technology area, software is a critical element of inventions in so many disparate fields that it presents an unusual challenge for any categorical classification system.[23]

The existing literature includes three separate efforts to identify a large data set of software patents. The first was by Graham and Mowery.[24] They did not attempt to define the term "software patent," but rather first used the IPC classification system and, subsequently, the USPTO classification system in an effort to develop a class-based data set of software patents owned by prepackaged software firms.[25] Specifically, they first identified prepackaged software firms, using the "Softletter 100," an industry publication that identifies the one hundred largest prepackaged software firms.[26] Then, Graham and Mowery identified the classifications of the patents assigned to those firms. Using those classifications, they ran additional searches to compile a data set of software patents.[27] Generally, Graham and Mowery analyze those data sets to assess the relation between patenting and research and development (R&D) and find no strong evidence that strategic patenting is impeding innovation.[28] As a definitional matter, their methodology produces large concentrations of patents on software inventions. At the same time, as we discuss below, it

23. With respect to the USPTO classification system, part of the problem reflects the changes over time of the relevant U.S. patent classifications that are most commonly used for software-related inventions. For a careful discussion of that development, see GREGORY A. STOBBS, SOFTWARE PATENTS ¶ 5.10[D], at 43–50 (Supp. 2006) (discussing the replacement of the old 395 class by a new set of classes in the 700 series).

24. Graham & Mowery, *Intellectual Property Protection*, *supra* note 19.

25. *See id.* (definition based on IPCs); Graham & Mowery, *Software Patents*, *supra* note 14 (definition based on U.S. patent classes).

26. Graham & Mowery, *Intellectual Property Protection*, *supra* note 19, at 232. As the discussion below should make clear, our project uses a considerably broader conception of the software industry, which is based on *Software Magazine*'s "Software 500" and covers many sectors excluded from the prepackaged software sector classification with which Graham and Mowery work. *See, e.g.*, The 2007 Software 500, http://www.softwaremag.com/SW500/index.cfm?trk=n&id=&MODE= (listing rankings for 2006 and showing the large number of software industry sectors represented).

27. The IPC-based definition uses G06F 3/ 5/ 7/ 9/ 11/ 12/ 13/ & 15/; G06K 9/ & 15/; and H04L 9/ to develop a data set of patents held by the top fifteen firms in the 1995 "Softletter 100." Graham & Mowery, *Intellectual Property Protection*, *supra* note 19, at 231–32. The definition based on U.S. classes uses classes 345, 358, 382, 704, 707, 709–11, 713–15, and 717 to develop a data set of all patents issued in those classes from 1987–2003. Graham & Mowery, *Software Patents*, *supra* note 14, at 14.

28. Graham & Mowery, *Software Patents*, *supra* note 14, at 19–22.

omits several classes that contain large numbers of software patents—namely, those obtained by software firms that are not prepackaged software firms.

The second significant effort to identify a large set of software patents appears in a paper by Jim Bessen and Robert Hunt.[29] Eschewing a class-based definition, Bessen and Hunt instead develop a keyword-search algorithm. Their technique starts with a definition of the term "software patent" that includes, as we do,[30] patents on inventions in which the data processing algorithms are carried out by code either stored on a magnetic storage medium or embedded in chips ("firmware").[31] In applying this

29. James Bessen & Robert M. Hunt, *An Empirical Look at Software Patents* 7–9 (Fed. Reserve Bank of Philadelphia, Working Paper No. 03-17/R, 2004).

30. As Bessen and Hunt note, one of the current authors, John Allison, earlier employed a definition of software patent that excluded "firmware," including only inventions in which the code implementing the data processing algorithms are stored on a magnetic storage medium. *Id*. at 9; *see also* John R. Allison & Mark A. Lemley, *The Growing Complexity of the U.S. Patent System*, 82 B.U. L. REV. 77, 89 (2002); John R. Allison & Mark A. Lemley, *Who's Patenting What? An Empirical Exploration of Patent Prosecution*, 53 VAND. L. REV. 2099, 2110–11 (2000); John R. Allison & Emerson H. Tiller, *The Business Method Patent Myth*, 18 BERKELEY TECH. L.J. 987, 1029 (2003). The reasons for using this definition were a combination of initial doubt and compromise with a coauthor, followed by a need for consistency. Each of those articles made use of the same data set of 1,000 randomly selected patents-in-general issued between mid-1996 and mid-1998. After gaining a great deal more experience from closely reading thousands of computer-related patents, Allison became firmly convinced that the definition should include firmware. When he used the same set of 1,000 randomly selected patents in a subsequent article, he studied each patent again and reclassified them using a definition that included firmware. *See* John R. Allison, Mark A. Lemley, Kimberly A. Moore, & R. Derek Trunkey, *Valuable Patents*, 92 GEO. L.J. 435 (2004) (definition not explicitly provided in article) [hereinafter Allison et al.]. Allison used this more inclusive definition not only in that paper and in this one, but also in an extensive empirical study of more than 7,600 university-owned patents. *See* Arti K. Rai, John R. Allison, Bhaven Sampat & Colin Crossman, *University Software Ownership: Technology Transfer or Business as Usual?* (Duke Law Sch., Research Paper No. 20, 2007), *available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=996456. Thus, when we say that identifying a large set of software patents is daunting, we speak from rich experience.

31. We note another, possibly minor, difference between the basic definition that Bessen and Hunt start with and the definition we use in this study. However, the effects of those differences are difficult to gauge. The Bessen-Hunt definition of a software patent includes patents on inventions that "use" software as part of the invention, but excludes those that "use" off-the-shelf software:

> Our concept of software patent involves a logic algorithm for processing data that is implemented via stored instructions; that is, the logic is not "hard-wired." These instructions could reside on a disk or other storage medium or they could be stored in "firmware," that is, a read-only memory, as is typical of embedded software. But we want to exclude inventions that involve only off-the-shelf software—that is, the software must be at least novel in the sense of needing to be custom-coded, if not actually meeting the patent office standard for novelty.

Bessen & Hunt, *supra* note 29, at 8.

Our definition does not distinguish between custom-coded and off-the-shelf software or firmware because, even if it makes sense to do so (and we cannot be sure that it does), it is a distinction that is frequently impracticable to apply. The written descriptions occasionally provide a basis for making this distinction, but many times do not.

definition, however, Bessen and Hunt rely on automated word searches of the specifications (written descriptions) of patents in their sample. Specifically, Bessen and Hunt start by examining a random sample of patents, which they classify according to their definition and use as the basis for a keyword-search algorithm that they can then apply *en masse* to classify patents more broadly.[32] Like Graham and Mowery, Bessen and Hunt focus their statistical analysis on the relation between patenting and R&D. Their conclusion, however, is much more pessimistic: that the increase in patent propensity during the 1990s reflects a patent "arms race" that undermines incentives to innovate.[33] As a definitional matter, they report in their paper that a test of their algorithm on a random sample indicates a false positive rate of 16% and a false negative rate of 22%.[34] Given those error rates, previous scholars have criticized their method for including excessive numbers of patents that are not fairly regarded as software patents.[35] As we discuss below, their definition also omits broad classes of patents that plainly cover software inventions.[36]

32. Bessen and Hunt's keyword search query consisted of: (("software" in specification) or ("computer" AND "program" in specification)) AND (utility patent excluding reissues) ANDNOT ("chip" OR "semiconductor" OR "bus" OR "circuit" OR "circuitry" in title) ANDNOT ("antigen" OR "antigenic" OR "chromatography" in specification). *Id*. at 42.

33. *Id*. at 26–38.

34. *Id*. at 9.

35. *See* Robert W. Hahn & Scott Wallsten, A Review of Bessen and Hunt's Analysis of Software Patents (Nov. 2003) (unpublished note, on file with the AEI-Brookings Joint Ctr.); Graham & Mowery, *Software Patents*, *supra* note 14; Anne Layne-Farrar, *Defining Software Patents: A Research Field Guide* (AEI-Brookings Joint Ctr. for Regulatory Studies, Working Paper No. 05-14, 2005).

36. For a discussion and comparison of efforts to define software patents, see Layne-Farrar, *supra* note 35. Layne-Farrar compared the results of the Bessen-Hunt study, the Graham-Mowery studies, and a study by John Allison and Emerson Tiller to determine how accurately the studies identified software patents. *Id*. at 4; John R. Allison & Emerson H. Tiller, *Internet Business Method Patents*, *in* PATENTS IN THE KNOWLEDGE-BASED ECONOMY, *supra* note 4, at 259, 259–84. Layne-Farrar reports that software "experts" (associates in her firm with computer science degrees) read a random sample of patents from the data sets in each study to determine which patents in those studies that were asserted as being software were not actually software ("false positives"). She found that the Bessen-Hunt data set included a very large percentage of false positives, but that the Graham-Mowery data sets and the Allison-Tiller data set contained only about 5% false positives. Layne-Farrar, *supra* note 35, at 16, 24 tbl.2. Some caveats about the Layne-Farrar study are in order. First, the random samples used by Layne-Farrar were taken from attempted replications of the data sets, not the actual sets. *Id*. Second, there was no indication that the readers with computer science degrees actually had extensive experience in studying software patents. Third, there was no indication that these readers used any particular definition of a software patent. Fourth, the replicated data set from the Allison-Tiller study was a preliminary one of approximately 2,800 patents before Allison and Tiller further refined it to a set of 1,423 internet-related software and software-implemented business method patents. *Id*. Fifth, the Allison-Tiller data set was not an appropriate one for use in her comparison. Even if she had used the final, refined Allison-Tiller data set, that data set did not purport to be a complete or even representative set of software patents; rather, it included all patents in PTO data-processing classes 705, 707, and 709 issued through the end of 1999 that clearly addressed internet

The third notable effort, in a recent working paper by Bronwyn Hall and Megan MacGarvie,[37] combines those approaches. Hall and MacGarvie first develop their own set of patent classes, by looking to classes commonly used in patents assigned to fifteen large software firms.[38] Then, they combine the patents in those classes with the patents in the IPC-class definition developed in the first Graham-Mowery paper.[39] Finally, they apply the Bessen-Hunt textual search to the patents in the combined set of classes and exclude the patents that do not satisfy the search criteria.[40] The object of their analysis is to consider whether the arguable reduction of patentability restrictions in the 1990s increased or decreased the value of firms in the industry.[41] They generally conclude that the initial extension of patentability decreased the value of firms in the industry but that market values have increased substantially in the subsequent years.[42]

Although each of the three previous definitions has the benefit of objectivity and replicability, and consequently can be applied to large data sets in an automated fashion, they sacrifice a great deal of accuracy. The classification system does not closely match the variety of patents obtained on software, and keyword searches produce at best a crude match to software patenting. On that point, Allison's studies of tens of thousands of computer-related patents convince us that different patent owners make highly idiosyncratic uses of language in the titles, abstracts, written descriptions, and claims of patents even in closely related technology fields. Moreover, as software is a critical part of inventions in far-flung fields, reliance on particular search terms will produce a data set that is substantially overinclusive and underinclusive at the same time.[43]

---

applications. Allison & Tiller, *supra,* at 261 n.10. Although internet-related patents are a subset of software patents, the objective of the Allison-Tiller study was to identify that subset of patents, not software patents in general.

37. Hall & MacGarvie, *supra* note 12.

38. *Id*. at 14–15.

39. *Id*. at 15–16.

40. *Id*.

41. *Id*. at 5–6.

42. *Id*. at 31–32.

43. To illustrate the problem, consider a few examples of overinclusiveness. Among the patents that satisfy the Bessen-Hunt algorithm are the following:

- Genetic Control of Flowering, U.S. Patent No. 6,265,637 (filed Jan. 11, 1999). The patent claims cover genetic engineering methods and products. *Id*. Although the claimed invention did not involve data processing, the Bessen-Hunt algorithm identified the patent as software because the inventors employed an existing software program in their research to predict the probability of a genomic sequence, and the word "software" thus appeared twice in the patent specification.
- Frozen Food Product, U.S. Patent No. 6,096,867 (filed July 22, 1997). The patent covers a frozen dessert product with a specified composition. *Id*. The Bessen-Hunt

Because of the problems with the earlier definitions, our work rests on a different approach—a careful patent-by-patent examination of the nature of the invention. As we discuss below, we have attempted to make our classification process as objective as possible. We start with the following definition of a software patent: "a software patent is one in which at least one claim element covers data processing—that is, the act of manipulating data—regardless of whether the code carrying out that data processing is on a magnetic storage medium or embedded in a chip."

Based on Allison's experimentation with various definitions in the course of individually studying tens of thousands of patents in several different studies over the years, we believe that this definition is both appropriately inclusive and susceptible to principled, consistent application. It also is important that our definition captures the realities of claim drafting. It is common for most of the elements of a patent claim to describe the prior art, with one or two elements describing the purportedly novel and nonobvious advance. For example, claims in the patents of computer hardware manufacturers often purport to cover items such as a generic router, server, printer, scanner, or other hardware, with perhaps only one claim element consisting of a function carried out by algorithms.[44] The same is true of patents issued to manufacturers outside

---

algorithm identified the patent as software because the inventors used the word "software" in the specification when they described how they employed an existing software program in their research for imaging and measuring the size of ice crystals.

- Hammer Device, U.S. Patent No. 5,305,841 (filed Oct. 9, 1992). This invention consists of a rock-crushing device. *Id.* The algorithm identified the patent as software because the inventors used the phrase "computer program" in the specification when they described their use of an existing software program for simulation purposes while designing the mechanical device.
- Thermoplastic Polypropylene Blends with Mixtures of Ethylene/Butene & Ethylene/Octene Copolymer Elastomers, U.S. Patent No. 5,985,971 (filed Oct. 29, 1997). This invention consists of certain groups of chemical compositions claimed to have superior adherence for coating materials such as automotive paints. *Id.* The algorithm identified the patent as software because the specification used the term "software" in connection with the inventor's use of a commercially available software program for simulating molding conditions in which the claimed compositions would be useful.
- Storage Case for Compact Discs, U.S. Patent No. 5,954,197 (filed Dec. 16, 1996). This invention consists of a mechanical storage and shipping container for CDs. *Id.* The algorithm identified the patent as software because the word "computers" appeared in the specification when the inventors noted than were CDs in jewel cases were easier to load into the CD drives of personal computers than CDs were stored in older types of cases. Separately, the specification used the word "program" to refer to audio program material commonly found on CDs. Thus, both "computer" and "program" were in the patent's written description.

44. *See, e.g.*, Method and Apparatus for Distinguishing Transparent Media, U.S. Patent No. 6,497,179 (filed July 19, 2001). Claim 1 of that patent reads:

the computer industry, such as a patent on a magnetic resonance imaging machine, in which perhaps only one claim element covers algorithms enabling the machine to accomplish a given imaging task.[45] A claim covers the entire invention; and in a case like this, the entire invention is not just the algorithms in isolation but instead is a piece of hardware that allegedly does something different because of the new algorithms. Moreover, patent attorneys do substantial harm to their client-applicants if they include an element in a claim that is not essential, because doing so narrows the scope of the patent. Thus, including a claim element covering the manipulation of data reveals that, at a minimum, data processing is a critical part of the invention.

Applying that methodology, we examined each of the 20,000 patents issued to firms other than IBM to determine whether it was a patent on a software invention. For the 14,000 IBM patents, we read a random sample of over 300 patents and extrapolated from that sample. Using that methodology, about 68% (13,500) of the non-IBM patents and about 55% of the IBM patents (extrapolating from the sample that we examined)

> 1. A printer (10) having a transparency film discrimination system, the discrimination system comprising: feed mechanism (18) for feeding a print medium (12) toward a print mechanism (20), the print medium (12) being one of a plurality of different types, each type having a print surface (14); illumination source (42) for providing light to impinge on the print surface (14); detector (48) for detecting one of reflected and transmitted light from the print surface (14) to provide a detection signal representing the print surface (14) so as to allow identification of transparency type of the print medium (12); and processor (28) for applying metric criteria to the detected signal to identify type of transparency of the print medium (12) and for providing control to the print mechanism (20) dependent on the identified transparency type so that damage to the printer (10) is avoided.

45. *See, e.g.*, Nuclear Magnetic Resonance Diagnostic Apparatus, U.S. Patent No. 4,656,423 (filed Nov. 4, 1983). Claim 1 reads:

> 1. An apparatus for examining an object by nuclear magnetic resonance comprising: magnet means for applying to the object a static magnetic field along an axis thereof; gradient coil means arranged along said axis, for applying to said object a first symmetric gradient magnetic field that in conjunction with said static magnetic field gives a predetermined constant magnetic field in a substantially two-dimensional plane of said object perpendicular to said axis, and for applying to said plane at least a second symmetric gradient magnetic field perpendicular to said first gradient magnetic field for defining a line within said plane along which nuclear magnetic resonance signals are read out; probe head coil means for applying RF pulses to said plane to excite nuclei therein, and for detecting nuclear magnetic resonance signals derived from said line; shifting coil means, arranged along said axis and disposed adjacent to and independent from said gradient coil means for superimposing an additional asymmetric gradient magnetic field parallel to and on said first gradient magnetic field independently of generation of said first magnetic gradient to shift said plane of predetermined constant magnetic field intensity in a direction perpendicular to said plane; and reconstruction means for receiving said nuclear magnetic resonance signals and for reconstructing a computerized tomographic image for a plane selected by the cooperation of said additional magnetic field with said first gradient magnetic field.

qualified as "software patents," for a blended total of about 62% (21,200) software patents. Table 1 summarizes the major classes of software patents using both the IPC and USPTO classification systems.

TABLE 1: LEADING SOFTWARE PATENT CLASSES

IPC CLASS	S/W PATENTS IN CLASS	% OF ALL S/W PATENTS	CUM. % OF S/W PATENTS	USPTO CLASS	S/W PATENTS IN CLASS	% OF ALL S/W PATENTS	CUM. % OF S/W PATENTS
G06F	8875	63.2	63.2	707	1551	11.0	11.0
H04L	583	4.2	67.4	345	1434	10.2	21.2
G06K	502	3.6	71.0	709	1264	9.0	30.2
G06T	438	3.1	74.1	711	890	6.3	36.5
H04N	329	2.3	76.4	710	753	5.4	41.9
H04B	317	2.3	78.7	714	730	5.2	47.1
G09G	262	1.9	80.6	717	664	4.7	51.8
G10L	241	1.7	82.3	713	637	4.5	56.3
B41J	227	1.6	83.9	370	598	4.3	60.6
H04J	206	1.5	85.4	382	420	3.0	63.6

The most obvious problem with our methodology is that it requires reading every patent, an extraordinarily slow and laborious process. Although the appropriate treatment of many patents is obvious under this definition, a substantial percentage must be studied with care.[46] Claims are often obtuse, and, in the computer field, they are frequently broad in scope. As a result, in many cases one must not only read the independent claims closely, but also examine the dependent claims and study the written description for assistance in interpreting the claim language.[47]

In the end, we acknowledge an irreducible element of subjectivity that undermines the ease with which our determinations can be replicated. At the same time, however, it seems clear that our determinations will be more accurate than the comparatively crude ones made under the automated class-based or keyword-search methodologies discussed above. For example, although the Graham-Mowery class-based searches are not significantly overinclusive because they limit their data set to patents owned by prepackaged software firms, the searches are significantly underinclusive. They exclude seven of the ten largest classes of software patents, including several classes that are dominated by software patents in

46. Because patents from firms in computer-related industries are more likely to include patents that are close to the line between software and nonsoftware patents, the percentage requiring careful scrutiny is far higher for a project like this one than it is for a project (like Allison's previous projects) studying a population of patents across a broad array of fields.

47. *See* Phillips v. AWH Corp., 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

our data collection, such as IPC G06T (image data processing or generation)[48] and H04J (multiplex communication).[49] A similar problem afflicts their more recent definition based on USPTO classifications, where they exclude several classes that in our data set are dominated by software patents, such as classes 455 (telecommunications),[50] 712 (processing architectures and instruction processing for electrical computers),[51] and 719 (interprogram or interprocess communication for electrical computers).[52]

Similarly, the Bessen-Hunt word-search algorithm suffers not only from the problem of overinclusiveness discussed above,[53] but also from underinclusiveness, because a great many software patents do not include the terms "software" or "computer" and "program" in their specifications.[54] The integral role of computerized data processing in the invention may be so clearly understood to the inventors and to other experts in a given field that these terms are just not mentioned in the patent specification even though a claim element may cover a computerized data-processing function.[55] Moreover, the exclusion of

48. 438 of the 444 patents in this class are software patents.

49. 206 of the 225 patents in this class are software patents.

50. 281 of the 332 patents in this class are software patents.

51. 327 of the 331 patents in this class are software patents.

52. All of the 271 patents in this class are software patents.

53. *See supra* note 35.

54. *See, e.g.*, Distributed Routing, U.S. Patent No. 6,370,584 (filed Sept. 1, 1998) Claims clearly cover data processing, thus making it a software patent, but written description contained only the word "computer" and not "program" or "software," which would exclude it from the Bessen and word-search algorithm. *Id.*

55. Rai and coauthors analyze a sample of over 7,600 patents, all patents issued in 1982, 1987, 1992, 1997, and 2002 to universities identified by the Carnegie Commission on Higher Education as research and doctoral universities. Rai et al., *supra* note 30, at 7. Allison coded these patents as software or nonsoftware by manual examination—applying the same protocol as used in this paper. *Id*. at 12–15. To analyze differences between the Allison method and the Bessen-Hunt keyword-search algorithm, Bhaven Sampat (one of the coauthors of the Rai et al. study), applied the Bessen-Hunt algorithm to the 2,942 university-owned patents issued during 2002 and compared the results to Allison's assessment of those patents. *Id*. at 12 n.32, 46 tbl.1. Sampat identified large numbers of patents that were treated differently, including many software patents that were not identified with the Bessen-Hunt algorithm, illustrating significant underinclusiveness. *Id*. Of the 2,942 patents, the Bessen-Hunt keyword search identified 221 patents as software that were not (overinclusiveness), and 198 as nonsoftware patents that did in fact cover software inventions (underinclusiveness). *Id*. To provide a few examples of underinclusiveness for the present study, Allison studied the first 214 patents in the list of 2,942 and identified twenty-one of that first 214 as software patents; the Bessen-Hunt search missed ten of those twenty.

To illustrate our view that these patents should qualify as software patents, we provide here descriptions of the first five of those ten:

> (1) Variable Resolution Imaging System, U.S. Patent No. 6,335,957 (filed Jan. 12, 1999). The patent claims an imaging system for medical and industrial purposes including software algorithms that purportedly improve the resolution of images. Software is at the heart of the

patents with terms such as "circuit," "chip," "circuitry," "bus," and "semiconductor" in the title removes many software patents, given the frequency with which the title of a patent will fail to reflect the true nature of the claimed invention.[56]

III. WHAT IS THE "QUALITY" AND "VALUE" OF SOFTWARE PATENTS?

A. Defining Patent "Quality" and "Value"

1. Different Meanings

Just as the rapid pace of innovation and the heterogeneous nature of software make it difficult to develop a definitive classification of

---

invention, but the Bessen-Hunt keyword search did not identify it as software because the terms "software" or "computer" and "program" do not appear in the specification

(2) Apparatus & Method for Improving Vision & Retinal Imaging, U.S. Patent No. 6,338,559 (filed Apr. 28, 2000). The patent claims: "a method and optical device for improving human vision," which includes in the claims software for generating high-resolution images of the retina. The claims include the term "computer means," and the specification twice includes the term "computer," but the specification does not explicitly refer to "software" or "computer" and "program."

(3) Genomics Via Optical Mapping with Ordered Restriction Maps, U.S. Patent No. 6,340,567 (filed Oct. 3, 2000). The patent abstract states: "[a] method of producing high-resolution, high-accuracy ordered restriction maps based on data created from the images of populations of individual DNA molecules (clones) digested by restriction enzymes. Detailed modeling and a statistical algorithm, along with an interactive algorithm based on dynamic programming and a heuristic method employing branch-and-bound procedures, are used to find the most likely true restriction map . . . ." In the data set used in the Rai et al. study, Allison categorizes this as one of a set of "pure" software patents that consist of nothing but algorithms. The Bessen-Hunt keyword search did not retrieve the patent because the specification did not refer to "software" or "computer" and "program.

(4) Method for Determining Storm Predictability, U.S. Patent No. 6,340,946 (filed Aug. 3, 2000). The patent claims a method for determining the predictability of elements in a weather radar image—more specifically, a method for generating a predictability score indicative of the predictability for a pixel in the weather radar image. Allison coded this patent as not only software, but also "pure" software, because the entire invention consisted of data processing. Although the specification referred to a "computer system," it did not refer to "software" or a "computer program."

(5) Methods for Analysis & Sorting of Polynucleotides, U.S. Patent No. 6,344,325 (filed Feb. 8, 2000). The patent claims a software-implemented method for analyzing and isolating polynucleotide molecules based on size by using an optical signal. Although the patent plainly claims data processing techniques, and the specification refers three times to a "computer," it does not refer to "software" and does not use the term "program."

56. *See, e.g.*, Universal Serial Bus Controlled Connect & Disconnect, U.S. Patent No. 6,415,342 (filed July 27, 1999). The invention in this patent covers a USB device and software for communications between the device and the computer to which it is connected and disconnected. The specification includes the word "software" once, but the Bessen-Hunt word search would exclude this patent because of the word "bus" in the title.

"software" patents, the vagueness of common criticisms of software patents makes it difficult to be sure what it would mean for a patent to have the "quality" that would deflect criticism. A major component of "quality" is a strong likelihood that a patent actually covers a novel and nonobvious invention and would withstand a challenge to its validity on those grounds.[57] Another is that the patent's written description and drawings provide enough explanatory support for its claims so as to enable a "person having ordinary skill in the art" ("PHOSITA") to make the invention and put it into practice without an undue amount of additional experimentation—the so-called "enablement requirement."[58] Although the literature suggests reasonably good proxies for novelty and nonobviousness,[59] we know of none available for descriptive adequacy.

This Article, for the most part, considers the extent to which software patents are likely to withstand challenges to their validity by examining the input of the drafter and the patent office. From that perspective, we could think of quality in two distinct ways. One possibility—captured in complaints that software patents are too broad or fail to account adequately for prior art—is that quality refers to the accuracy and completeness with which the patent defines an invention and distinguishes it from prior art. Thus, this concept of quality is one of craft, focusing on the efforts of the patentee and the examiner to produce an appropriate description of the invention and to determine that the invention satisfies the statutory standards of novelty and nonobviousness.[60] One common concern here is that current incentives motivate software firms to patent too quickly, seeking patent protection without investing the appropriate effort to reduce an abstract idea to a useful product.[61] Related to that point is the problem that patent examiners unfamiliar with a cutting-edge technology like software may be less capable of assessing the quality of the disclosure or of the innovation than they are in technological areas with which they are more familiar. This problem raises the question

57. Under section 102(a), an invention is novel if no identical invention appears in a single piece of prior art. 35 U.S.C. § 102(a). To be patentable, an invention must not only be novel but also must represent more than an obvious, or trivial, advance over the cumulative prior art. *Id*. § 103.

58. *Id.* § 112.

59. *See infra* notes 62–63 and accompanying text.

60. 35 U.S.C. §§ 102, 103.

61. This idea is voiced frequently in Mann's interviews with venture capitalists and investors in software firms, who worry that a focus on rapid-fire patenting distracts from a focus on successful product design.

whether the patented invention in fact represented a novel and nonobvious advance over what had been done before.[62]

A second possibility—captured in complaints that software patents are trivial or too easy to obtain—is that quality refers to the economic value of the patent.[63] This conception of quality overlaps with the first, because a patent granted without adequate consideration of the prior art is more likely to be held invalid. In addition, a patent in which the invention is inadequately claimed may fail to capture all of the infringing activity that it could have. Yet, this conception of value is only partly dependent on quality, because it also encompasses the value of the markets and products with which the patent is concerned: the economic value of a patent relates directly to the value of the commerce over which the patent grants a right to exclude. Thus, value in its broadest conception is not, strictly speaking, relevant to the question on which we focus here: how well the system is functioning to distinguish "good" and "bad" patents.

Both of these conceptions of quality have important policy implications. The first conception of quality (the *ex ante* conception) is important in efforts to improve the processes by which the PTO considers and issues patents. The second conception (the *post hoc* conception) is important in efforts to improve the system by which patents are licensed and enforced, to ensure an appropriate balance of access to technology and incentive to invent. As we emphasize above, however, this Article focuses on the former *ex ante* conception of quality.

2. Indicators of "Quality" and "Value"

The academic literature has been much more successful in efforts to identify objective indicators of the *post hoc* conception of value than it has been in identifying objective indicators of the *ex ante* conception of quality. Generally, the strategy has been to identify some objective

62. The requirement that an invention be novel (that there be no single piece of prior art disclosing an identical invention) appears in 35 U.S.C. § 102(a) (2000). The more onerous requirement that an invention be "nonobvious" (i.e., that the patent claim something more than an obvious advance over the prior art as a whole) appears in 35 U.S.C. § 103 (2000). In the statutory phrasing, the nonobviousness determination depends on the perspective of a "PHOSITA," a hypothetical "person having ordinary skill in the art." *Id.*

63. We refer here to "private value" (the value of a patent to its owner) and not to "social value" (the value of a patent to society). Moreover, like most other scholars, we refer to the value of a stand-alone patent and not to the value that a patent may contribute to a patent portfolio. At least in theory, a patent might have little value by itself, but may have value in its contribution to a company's portfolio of patents in which the whole is greater than the sum of its parts. *See, e.g.*, Gideon Parchomovsky & R. Polk Wagner, *Patent Portfolios*, 154 U. PA. L. REV. 1 (2005).

characteristic of a patent or an event in the life of a patent that provides an indication that the patent has value, and to search for notable patterns in the characteristics of the affected patents.[64]

For a single patent, or a small number of patents, a reasonable test of quality would be to have a person with ordinary skill in the given art conduct a thorough search of the prior art and evaluate the likelihood that a patent discloses a novel and nonobvious invention. Such an approach is simply not feasible for data sets that contain a large enough number of patents for statistical analysis. In any event, the results often would still be subject to doubt: this is what is done in patent infringement litigation, which frequently involves genuine issues of disputed fact on which patent validity depends.

We explain and measure several indicators of patent quality and value: the number of different types of prior art references, the informational quality of so-called "nonpatent" prior art references, the number of total and independent claims, and the number of "forward citations" (references to patents in our data set by later patents). We also mention several other potential indicators of quality or value that we are not able to use in our study.

Prior Art. Putting that impractical "first best" solution to one side, it is reasonable to conclude that the best proxy we have for patent quality[65] is the number of prior art references together with (to the extent practicable) some assessment of the types and informational content of those references and their sources.[66] Prior art is the objective evidence of what

64. *See, e.g.*, Allison et al., *supra* note 30; *see also* Allison & Tiller, *supra* note 30.

65. We note that the quantity of prior art may relate not only to quality but also to value. For example, if a thorough search of prior art is expensive, a greater quantity of prior art suggests that the patent applicant perceived that its invention was important enough to make this investment. In addition, we know from prior research that patents involved in infringement litigation have significantly more prior art references than nonlitigated patents. Allison et al., *supra* note 30. This makes sense because, all other things being equal, litigated patents should be more valuable to their owners than the general population of patents. However, factors other than the size of the stakes also affect the propensity to litigate patents. *See, e.g.*, Jean O. Lanjouw & Josh Lerner, *The Enforcement of Intellectual Property Rights: A Survey of the Empirical Literature*, 49/50 ANNALES D'ECONOMIE ET DE STATISTIQUE 223, 223–46 (1998) (Fr.) (surveying empirical studies in economics on intellectual property litigation, including the relationship between patent litigation and patent value); Jean O. Lanjouw & Mark Schankerman, *Enforcing Intellectual Property Rights* (Nat'l Bureau of Econ. Research, Working Paper No. 8656, 2001) (finding that the fact, but not the outcome, of litigation correlates with patent value).

66. *See* John R. Allison & Starling D. Hunter, *On the Feasibility of Improving Patent Quality One Technology at a Time: The Case of Business Methods*, 21 BERKELEY TECH. L.J. 729, 751–57 (2006); Allison & Tiller, *supra* note 30, at 1045–52; Deepak Hegde & Bhaven N. Sampat, Examiner Citations, Applicant Citations, and the Private Value of Patents (Dec. 2006) (unpublished manuscript, on file with authors).

previously has been done. The most important types of prior art that can be gleaned from patents themselves are references to prior U.S. and foreign patents and to prior printed publications of various types (often called "nonpatent prior art"). It is intuitively appealing to view the quantity and, to the extent we can measure it, the quality of prior art cited in patents as indicators of patent quality. Both should correlate with (a) the seriousness of the patent applicant's effort to identify previous inventions and distinguish the new invention from prior ones, and with (b) the rigor and thoroughness of the PTO's examination.[67] Furthermore, the most common basis for judicial invalidation of patents is prior art that was not considered by the PTO.[68]

That is not to say that the number and quality of references is a perfect indicator of patent quality. For example, some applicants load up patents with very large numbers of references, in an apparent effort to distract attention from the most important prior art references. Yet, in a large data set of patents, it is reasonable to think that patents disclosing more prior art references reflect a level of effort on the part of the applicant and examiner that is higher on average than the effort involved in patents that disclose fewer references. For the applicant, greater effort also means greater investment.

Although other objective indicators might seem at first to be more relevant to "value" than to "quality," we believe that they are important as well. First, given the imperfection of inferences drawn from prior art, additional information about patent value buttresses those inferences. For example, suppose, as seems likely, that patent owners (patentees) care more about quality when they expect their patents to be valuable. Patentees' *ex ante* perceptions about value should have some reliability, because patentees at the time of filing and prosecution have a considerable

67. Regarding the thoroughness of both the applicant's prior art search and the examination process, patent and nonpatent prior art references may be provided by either the applicant or the examiner. Intuitive arguments suggest that patent applicants should be responsible for more prior art references than examiners. *See* Allison & Tiller, *supra* note 30, at 1037–38 n.167. Recent empirical work by Bhaven Sampat quantifies the accuracy of this intuition. *See* Bhaven N. Sampat, When Do Applicants Search for Prior Art? A Window on Patent Quality (Nov. 2006) (unpublished manuscript, on file with authors) (concluding that applicants provided 59% of references to prior patents and 90% of references to NPPA, with examiners adding 41% and 10%, respectively).

68. *See* John R. Allison & Mark A. Lemley, *Empirical Evidence on the Validity of Litigated Patents*, 26 AIPLA Q.J. 185, 231–34, 251 (1998) (examining the population of litigated patents leading to final written decisions on validity or invalidity during 1989–96). The "challenger" to a patent's validity is usually the defendant in a patent infringement suit, but may be a plaintiff in a declaratory judgment action.

amount of value-relevant information about the market in which they will deploy the claimed invention.

Claims. Like the number of prior art references, the number of claims in a patent relates intuitively to patent value. Following the detailed written description of the invention in a patent,[69] claims identify the invention with linguistic precision. Patent claims define the patent owner's property interest.[70] Having a patent attorney draft more claims necessarily costs more money.[71] Moreover, increasing the number of claims in a patent sometimes increases the universe of potential infringers and the likelihood that the patent will be held to extend to competing products. This seems particularly true with respect to independent claims.[72] Empirical research has validated that intuition with respect to the total number of claims, illustrating that litigated patents have significantly more claims than nonlitigated ones.[73] At the same time, as with prior art references, the relation between the number of claims and the quality of the patent is more ambiguous. A patent with a large number of claims often might include broad and ill-confined descriptions of technology already in use at the time the applicant filed the application.

Forward Citations. Another intuitive and empirically validated indicator of patent value is the number of times that later patents cite a particular patent as prior art (referred to as "forward citations" or "citations received"). It is reasonable to expect a correlation between the number of forward citations and the relevance of the patent to continuing developments in the applicable technological field. The number of forward

69. 35 U.S.C. § 112 ¶ 1 (2000).

70. *Id.* ¶ 2.

71. More claims in a patent application also modestly increase USPTO examination fees. *See* 37 C.F.R. §§ 1.116(h), (i) (2007).

72. Patent claims are either independent or dependent. As the term implies, an independent claim stands by itself. Each dependent claim adds more specificity to one of the independent claims, narrowing its technological reach. The advantage to the patent drafter of the dependent claim is that a narrower dependent claim might be held valid, even if the broader independent claim is held invalid. Conversely, patent drafters often write multiple independent claims that cover the same invention, using different verbal formats to increase the odds that a later product that is substantively identical or similar to the disclosed invention will not "slip through the cracks." Consequently, we should expect a positive correlation between the number of independent claims and patent value.

One recent example illustrating the private value of using multiple independent claims is the well-known case of *NTP, Inc. v. Research in Motion, Ltd.*, 418 F.3d 1282 (Fed. Cir. 2005), in which NTP successfully sued RIM, the maker of the Blackberry personal communication device, for infringement of several NTP patents. *Id.* at 1287–88. The Federal Circuit concluded that the Blackberry functioned in a way that did not violate the "method" (i.e., process) claim of the NTP patent. *Id.* at 1317–18. However, it did infringe the NTP patent's "system" claim. *Id.* at 1317.

73. *See* Allison et al., *supra* note 30, at 451–52; Jean O. Lanjouw & Mark Schankerman, *Characteristics of Patent Litigation: A Window on Competition*, 32 RAND J. ECON. 129 (2001).

citations similarly relates to the likelihood that the patent disclosed a fundamental development in the particular technology field, thus giving the patent owner a valuable "head start." Moreover, later citations to a patent by the owner of the earlier patent ("self-citations") suggest that the patent owner is building a group of patents on closely related technological advancements, which suggests a greater likelihood of commercial exploitation of the patent.

Hence, it is no surprise that empirical studies have found significant positive correlations between the number of forward citations and value. One study, for example, concluded that forward citations relate to the market value of the firms that own the underlying patents.[74] Other research shows that litigated patents have significantly more forward citations than unlitigated patents.[75]

There are, of course, many reasons unrelated to technological import why later patent applications might, or might not, cite a particular patent as prior art. Some applications might avoid citing a patent in an effort to undermine perceptions of the patent's quality. This is particularly true in the milieu of biotech start-ups, where the details of patent applications are a topic of interest to both investors and competitors. It seems a much less important concern in the start-up sector of the software industry, in which it is considerably less customary for competitors or investors to study the details of patent applications.[76] Thus, we believe that forward citations are a plausible indicator of the *post hoc* value, and thus an indirect confirmation of the *ex ante* quality.

Maintenance Fees. Another possible indicator of value is maintenance fees. Intuitively, the willingness of patent owners to keep their patents alive by paying maintenance fees should relate to the value of the patent to its owner.[77] The failure of the owners of most patents to pay the relatively modest maintenance fees necessary to keep their patents in force suggests

74. Bronwyn H. Hall, Adam Jaffe & Manuel Trajtenberg, *Market Value and Patent Citations*, 36 RAND J. ECON. 16 (2005); Dietmar Harhoff, Francis Narin, F.M. Scherer & Katrin Vopel, *Citation Frequency and the Value of Patented Inventions*, 81 REV. ECON. STAT. 511 (1999); Manuel Trajtenberg, *A Penny for Your Quotes: Patent Citations and the Value of Innovations*, 21 RAND J. ECON. 172 (1990).

75. *See* Allison et al., *supra* note 30, at 454–55; Lanjouw & Schankerman, *supra* note 65. Interestingly, the number of self-citations—citations in later patents granted to the same patent owner—correlates even more significantly with litigation propensity than the number of forward citations by others. Allison et al., *supra* note 30, at 454.

76. *See* Mann, *supra* note 12, at 1004.

77. Such fees are due in increasing amounts at 3.5 years ($900), 7.5 years ($2,300), and 11.5 years ($3,800) after the patent issues. 37 C.F.R. § 1.20(e)–(g) (2006). Those fees are halved for small entities. *Id.*

that more than half of all patents are not worth even a few thousand dollars a few years after they are issued.[78] That intuition is buttressed by empirical research identifying strong positive correlations between the indicators of value discussed above and maintenance fee payments. Specifically, Kimberly Moore concluded that patents with more claims and forward citations are more likely to be maintained.[79] Although the recency of patents in our data set precludes us from using maintenance fees as an indicator of value in our work, the relation between maintenance fees and the indicators on which we rely adds validation to those indicators.

Patent Families. The last important indicator of value used in the existing literature is the number of countries in which the owner has obtained patent protection on the same invention (often referred to as a "patent family"). This makes sense because of the large expense of patenting in multiple countries. For example, Jean Lanjouw and Mark Schankerman have developed a patent quality index that relies on the number of claims in the patent, prior art references in the patent, the number of forward citations to the patent, and the number of countries in which the patentee sought protection for the invention.[80] Although the number of countries in which an applicant seeks patents on the same invention is almost certainly a valid value indicator, it would have little, if any, meaning in our study because innovation and patenting in the software industry is dominated by the United States.

---

78. *See* Jean O. Lanjouw, Ariel Pakes & Jonathan Putnam, *How to Count Patents and Value Intellectual Property: The Uses of Patent Renewal and Application Data*, 46 J. INDUS. ECON. 405 (1998); Mark A. Lemley, *Rational Ignorance at the Patent Office*, 95 NW. U. L. REV. 1495, 1504 tbl.3 (2001).

79. Kimberly A. Moore, *Worthless Patents*, 20 BERKELEY TECH. L.J. 1521, 1530–52 (2005). Moore's bivariate analysis found significant positive relationships between the payment of maintenance fees and the number of claims, number of prior art references, and number of forward citations. *Id*. at 1530. In her multivariate logistic regression, however, the number of prior art references lost its significance. *Id.* at 1537–38.

This loss of significance for number of references in the multivariate analysis apparently is caused by the interaction between numbers of prior art references and numbers of claims. For discussion of the correlation between the number of claims and number of prior art references, see Allison & Tiller, *supra* note 30, at 1055 (showing high correlation between number of claims and number of prior art references).

80. Jean O. Lanjouw & Mark Schankerman, *Patent Quality and Research Productivity: Measuring Innovation with Multiple Indicators*, 114 ECON. J. 441, 441–44 (2004).

3. Our Approach

Writing against the backdrop of that literature, we decided for our assessment of the quality of software patents to focus on the three data points that dominate the existing empirical literature: (1) the number of claims in the patent, (2) the number of prior art references in the patent, and (3) the number of forward citations to the patent. In an effort to obtain more nuanced information about the nature of patenting in the industry, we broke down two of those three principal variables into subcategories. First, we analyzed not only total claims but also independent claims.[81] Second, we broke down total prior art references in the patent into three categories: U.S. patent references, foreign patent references, and nonpatent prior art (NPPA) references.[82] Given the wide disparity in the informational value among NPPA references, we also made a rough estimate of the quality of these references by categorizing them according to the sources from which they came.

In total, our approach produced seven different data points that we could use to describe the patents: the numbers of total claims, independent claims, total references, U.S. patent references, foreign patent references, nonpatent references, and forward citations.[83] Table 2 sets forth some

81. The only previous analysis of independent claims separate from total (independent plus dependent) claims appears in Allison's previous study. Allison et al., *supra* note 30, at 451–52, 478–79. The large population study reported in that paper analyzed only the number of total claims and—using a multivariate logistic regression—found a highly significant relationship between the number of total claims and litigation (litigated patents being viewed as a subset of valuable patents) ($p = <0.0001$). *Id*. at 478. The smaller, more finely graded sample study reported in that paper analyzed the number of independent and dependent claims separately. Bivariate analysis showed highly significant relationships between number of both independent and dependent claims and the fact that the patents were litigated ($p = 0.0011$ for independent claims and 0.0044 for dependent claims). *Id*. at 452 n.68. When using a multivariate logistic regression for the sample study, the positive relationship between number of independent claims and litigation remained highly significant ($p = 0.006$), but the relationship between number of dependent claims and litigation ceased to be significant at the .05 level although significant at the .10 level ($p = .08$). *Id*. at 452 n.68, 479.

82. Delphion shows both the references themselves and the number of references for U.S. and foreign patents. *See* https://www.delphion.com/research/. For nonpatent references (called "other references" in Delphion), however, Delphion only shows the references themselves and does not report a count. *Id*. Moreover, these references are run together as lines in a text file. We developed a computer program to count the number of nonpatent references.

83. Because some of the patents were issued quite recently, there is a substantial amount of truncation in the forward citations. Accordingly, although Table 2 reports a simple descriptive statistic for the entire data set, our statistical analysis follows Jaffe and Trajtenberg in analyzing for each patent the number of forward citations divided by the average number of citations per patent application filed in the same year. *See* ADAM B. JAFFE & MANUEL TAJTENBERG, PATENTS, CITATIONS, AND INNOVATIONS: A WINDOW ON THE KNOWLEDGE ECONOMY 434–46 (2002).

simple descriptive statistics for those variables for the entire data set, which includes data for both software and nonsoftware patents.

TABLE 2: DESCRIPTIVE STATISTICS

VARIABLE	MEAN	MEDIAN	MIN	MAX	SD	MEDIAN (BOTTOM DECILE)	MEDIAN (LEAST CITED)
TOT. CLAIMS	20.1	18	1	318	13.8	4	16
IND. CLAIMS	3.7	3	1	92	2.7	1	3
TOT. REFS	16.0	11	0	470	22.9	3	9
U.S. PAT. REFS	11.8	9	0	352	14.3	2	8
FOR. PAT. REFS	0.9	0	0	88	2.5	0	0
NONPAT. REFS	3.3	1	0	396	12.8	0	0
FORWARD CITATIONS	6.7	3	0	291	10.3	0	--

As the large standard deviations suggest, there is substantial variance among the quality and value indicators in our data set of patents. To quantify that variation, the last two columns in Table 2 describe the worst of the patents by two separate metrics: the median of the bottom decile of each characteristic and the median on each characteristic of the bottom decile of patents by forward citations. As those columns show, a substantial number of patents fare poorly on the metrics we examine. Most notably, one in twenty of the patents have three references or fewer, a strikingly low number for any field of search. Thus, although the data do undermine the idea that software patents, *as a group*, are notable for their poor quality, they suggest at the same time that a substantial number are of low quality. Prior research has shown, however, that patents of all kinds vary greatly in their quality and value, many faring very poorly on the metrics we use.[84]

B. Software and Nonsoftware Patents

The central question is whether information about the characteristics of software patents on the data points summarized in Table 1 suggests that software patents are "better" or "worse" in some objective way than other patents. The problem in answering that question lies in selecting an appropriate baseline. If we were to match the software patents to other

84. Allison & Tiller, *supra* note 30, at 1036–58 (documenting large variances in these indicators in the course of finding that internet-related business method patents had substantially higher value and quality indicators than the average patent and than patents in most other technology areas).

randomly selected patents issued on the same dates, we might determine how software patents differ on the selected characteristics from typical patents. However, we ultimately concluded that the most informative way to evaluate these patents was to compare the software patents in the data set to the nonsoftware patents. Thus, we reduce the number of potential confounding variables and study a sample of patents obtained by the same set of firms during the same periods, with the only difference being the particular types of technology covered by the patents.

We compared the software and nonsoftware patents among the 20,000 patents issued to firms other than IBM. As Table 3 illustrates, the software patents (SWP) had significantly more total prior art references, claims, independent claims, and forward citations than the nonsoftware patents (NSWP). All of those differences are significant at least at the 99% confidence level ($p = 0.01$ or less).[85] For comparative purposes, we also include parallel data on the number of claims and references from the roughly contemporaneous sample of one thousand randomly selected patents that John Allison and his coauthors analyzed in *Valuable Patents*.[86] Because the general patents for the most part have even lower indicators of quality than the nonsoftware patents in our data set, that comparison further buttresses our results.

---

85. We analyze the differences in Table 3 with a two-sample t test with equal variances. As the Statistical Appendix discusses, we investigated the robustness of those differences with a variety of controls and multivariate regression models, which buttresses the results discussed in the text.

86. Allison et al., *supra* note 30.

TABLE 3: SOFTWARE AND NON-SOFTWARE PATENTS

	MEAN	MEDIAN	SD	N
TOT. CLAIMS				
SWP	22.9	20	15.4	14,044
NSWP	17.8	17	12.4	5,804
GENERAL	14.9	12	11.5	1,000
IND. CLAIMS				
SWP	4.2	3	2.8	14,044
NSWP	3.4	3	2.4	5,804
ADJ. FORWARD CITATIONS				
SWP	1.1	0.64	1.46	14,042
NSWP	0.9	0.49	1.24	5,799
TOTAL REFERENCES				
SWP	18.5	11	30.5	14,044
NSWP	14.1	10	16.6	5,804
GENERAL	15.2	10	16.3	1,000
U.S. PATENT REFERENCES				
SWP	12.7	8	18.3	14,044
NSWP	11.1	8	13.4	5,804
FOREIGN PATENT REFERENCES				
SWP	0.84	0	2.82	14,044
NSWP	1.3	0	2.8	5,804
NONPATENT REFERENCES				
SWP	5.0	1	17.9	14,044
NSWP	1.6	0	4.9	5,804

Generally, those results suggest a sanguine picture of the quality of software patents. To be sure, as we discuss in Part IV, it is likely that the differences between software patents and nonsoftware patents depend at least in part on aspects of software patent drafting and software technology. The data do cast doubt, however, on broad assertions and anecdotal suggestions that software patents as a group are of lower quality than patents in other areas of technology. Given the academic literature discussed in the preceding section that links the indicators in Table 3 to various indicators of the *post hoc* value of the patent, the evidence we present here suggests that any problems with quality are much more ambiguous.

The data on types of prior art references is particularly reassuring, because the differences between software and nonsoftware patents display a pattern that is consistent with the reality of software patent drafting. First, the subcategory of foreign patent references is the only one of our seven data points that appears in software patents with a lower frequency than in nonsoftware patents. The relative paucity of software patents in other countries[87] and the centrality of the United States to software innovation make the infrequency of foreign references entirely predictable.

Similarly, the recent rise of software patenting suggests that software patents would have many more nonpatent references than nonsoftware patents during our period of study. We did, in fact, find that the difference between software and nonsoftware patents in the number of NPPA references is greater than the difference on any other data point.

To provide a better understanding of the nature of the prior art, we examined the NPPA in a random sample of two hundred software and two hundred nonsoftware patents from our data set.[88] For a large set of patents, there is no practicable way to make quality distinctions among the patents referred to therein as prior art; that is, there are no feasible means to assess the informational value of such references in a large data set. NPPA references are, however, susceptible to such quality distinctions, because they can be classified in various ways to roughly reflect the probable accuracy, reliability, and objectivity of information contained in them. Any typology of the many kinds of printed publications (i.e., NPPA) is necessarily subject to some subjectivity and uncertainty, but it is possible to categorize such references in a way that enables at least a rough assessment of relative informational value. For example, an article in a journal that has been reviewed by referees or editors possesses a greater likelihood of being objective and accurate than an unmediated publication by a company or an industry group. The Appendix describes our typology of NPPA, developed by one of the authors and his coauthors for two previous research projects.[89]

87. *See, e.g.*, FLORIAN MUELLER, NOT LOBBYISTS AS SUCH (2005) (discussing failure of efforts to authorize software patents in the European Union).

88. We also analyzed NPPA references in a random sample of fifty IBM patents, thirty-three of which turned out to be software and seventeen nonsoftware. The amount of NPPA in these patents was so small that we have not reported it in table form. Only twelve out of the thirty-three IBM software patents and seven out of seventeen nonsoftware patents cited any NPPA at all. However, one notable fact about the NPPA in the IBM patents is that, in the twelve out of thirty-three software patents that cited at least some NPPA, thirty-five out of forty-six total NPPA references (76%) were cites to academic literature. In the seven out of seventeen nonsoftware patents that cited at least some NPPA, only five out of fifteen total NPPA references (33%) were cites to academic literature.

89. *See* Allison & Tiller, *supra* note 30, at 1045–52. This typology was created by carefully

Table 4 shows the results of our study of the NPPA references in a random sample of two hundred software patents and two hundred nonsoftware patents. Although our purpose here is to assess the NPPA qualitatively, the differences in the quantity of NPPA between the two samples are striking. Nonsoftware patents cite less than 25% of the NPPA cited in software patents. Moreover, 63.5% of nonsoftware patents cite no NPPA at all, while only 34.5% of software patents fail to cite any NPPA.

On the other hand, our categorization of NPPA suggests that the NPPA that *is* cited in nonsoftware patents is not qualitatively inferior to that cited in software patents—there is just much less of it. For instance, a significantly higher percentage of the NPPA found in nonsoftware patents consists of academic literature, although the gap closes considerably when we combine academic and trade publications. Like academic journal articles, practitioner-oriented trade publications typically must pass a review by editors or referees and should be viewed as possessing relatively high informational quality. Also, our sample of nonsoftware patents includes no references to the popular press, arguably a relatively low-quality source, whereas such references amount to almost 1/8 of total NPPA in the sample of software patents. One cannot make too much of these relative differences, however, when they are so overwhelmed by the quantity of high-quality NPPA in software patents: the amount of *academic* prior art in software patents is far more than the total amount of *all* nonpatent prior art in nonsoftware patents. The fairest conclusion is that, using our categories of NPPA as rough proxies for informational quality, software patents fare at least as well as nonsoftware patents, and perhaps even a bit better.

---

studying the NPPA references in over one hundred randomly selected internet-related business method patents and over one hundred randomly selected patents-in-general, and defining categories based on the nature of the reference sources we found in those patents. *Id*. at 1046. The categories were slightly modified in the second article in which one of the authors and another coauthor analyzed the sources of NPPA references. *See* Allison & Hunter, *supra* note 66, at 741–42.

In the first article, the typology combined academic and trade publications. Because of experience subsequently gained, in the second article we felt more confident in our ability to distinguish between academic and practitioner-oriented trade journals and thus separated them into two categories. We separate them in the current study.

TABLE 4: NPPA IN SOFTWARE AND NONSOFTWARE PATENTS

	Mean	Median	Std. Dev.	Min	Max	N	% of Total NPPA
Academic							
SW	2.60	0	0.43	0	40	519	37.45%
NSW	1.16	0	3.91	0	30	232	68.24%
Trade							
SW	1.75	0	1.01	0	200	349	25.18%
NSW	0.09	0	0.42	0	3	18	5.29%
University							
SW	0.23	0	0.08	0	14	45	3.25%
NSW	0.06	0	0.41	0	5	12	3.53%
Software							
SW	0.25	0	0.07	0	11	49	3.54%
NSW	0	0	0	0	0	0	0%
Patent- Related							
SW	0.26	0	0.09	0	14	51	3.68%
NSW	0.11	0	0.51	0	4	21	6.18%
Govt. Doc.							
SW	0.13	0	0.06	0	10	26	1.88%
NSW	0.07	0	0.35	0	3	14	4.12%
Comp./Ind.							
SW	0.92	0	0.17	0	19	183	13.20%
NSW	0.22	0	0.71	0	5	43	12.65%
Popular Press							
SW	0.82	0	0.77	0	154	163	11.76%
NSW	0	0	0	0	0	0	0%
Other							
SW	0	0	0	0	1	1	0%
NSW	0	0	0	0	0	0	0%
Total							
SW	6.93	1	28.7	0	384	1386	100%
NSW	1.7	0	4.44	0	31	340	100%

C. Patent Quality and Firm Specialization

The next subject that we analyze is the difference in patenting among the types of firms in our data set. We examine that question in two different ways: (1) by comparing firms that specialize in software to those that have more varied product lines, and (2) by considering a small set of large firms to our broader data set. Generally, we find that the patents of "pure software" firms are of higher quality than those of firms with substantial nonsoftware product lines. This finding is not surprising, because it is consistent with the findings above. Surprising, however—

given the persistent criticisms large firms have leveled at the patents of small firms—is the finding that, broadly speaking, there are few notable differences between the patents of large firms and those of the industry as a whole.

1. Patenting by Pure Software Firms

First, we consider the possibility that the patents that pure software firms obtain are "better" (or "worse") in some cognizable way than parallel patents obtained by firms that are not pure software firms. We are motivated to examine this question because of the apparent differences in culture and business strategy between pure software firms (e.g., Microsoft) and the large number of electronics firms that are important software developers and patentees in our data set (IBM being the most obvious example).[90]

For these purposes, we decided (somewhat arbitrarily) to treat firms with 80% or more of their revenues from software as pure software firms and those with less than 80% of their revenues from software as mixed software firms. Under this definition, of the 294 firms in our data set to which patents were issued during the five-year period (a large majority of the firms obtained no patents during that period), 208 (71%) were pure software firms and 86 (29%) were mixed firms. One reason to think that this division should be important for patenting practices is that the patents of the software firms were almost exclusively software patents (3948/4006 or 99%), while software patents represented only 64% of patents obtained by mixed firms (10,092/15,838).

The software patents obtained by software firms differed significantly in all relevant respects from the software patents obtained by mixed firms.[91] As we report in Table 7 in the Statistical Appendix, the results are strikingly similar to the results in Table 3: highly significant differences for numbers of total claims, independent claims, adjusted forward citations, foreign patent references (negative), and nonpatent references. The last finding—showing that later patents cite the software patents of pure software firms significantly more—suggests that the software patents

90. *See* Allison, Dunn & Mann, *supra* note 1 (including a detailed discussion of the differing exploitation strategies of incumbents, venture-backed start-ups, open-source developers, and independent developers).

91. In a related paper using this same data set, we find no significant relation between the share of a firm's revenue attributable to software sales and either the rate of patenting or propensity to patent. *Id*.

of pure software firms are more rapidly integrated into future software innovation than the software patents of mixed software firms.

This finding parallels the recent finding of Graham and Mowery that the quality of patents (measured solely by number of forward citations) held by software firms is higher than the quality of patents held by electronics firms.[92] They attribute this distinction to greater strategic patenting by electronics firms (that is, patenting for occupying general fields of technology rather than for protecting products of the patentee).[93] Our data make us skeptical of that explanation, primarily because the nonsoftware patents held by pure software firms do not differ nearly so much from the nonsoftware patents held by mixed software firms as do the respective sets of software patents. Of course, it is possible that electronics firms use strategic patenting more aggressively in the software area than they do in nonsoftware areas. It seems more likely to us, however, that the differences reflect the interaction of two distinct effects: (1) that software patents tend to have more substance (claims, references, and citations) than nonsoftware patents, and (2) that pure software firms are better able to integrate their software patents into their subsequent development efforts.

2. Patenting by Superpatentees

To get at the possibility of cultural differences in an alternative way, we also compared the software patents of fifteen "superpatentee" firms to the software patents of other firms in our data set. The superpatentee firms were those firms that appeared in the data set every one of the five years and that have at least fifty total patents.[94] Here, as Table 8 summarizes, we found—to our surprise—no significant differences between the software patents of the superpatentee firms and the software patents of other firms in our data set. Among other things, the data replicate the large standard deviations that characterize the software patent data for our larger data set: even for the superpatentee firms, a substantial number of the patents have very few claims, references, and forward citations. As previously observed, however, large variances and substantial numbers of patents

92. Graham & Mowery, *Software Patents*, *supra* note 14, at 24.

93. *Id*.

94. These firms are: Adobe, Apple, Autodesk, Computer Associates, EDS, EMC, HP, Mentor Graphics, Microsoft, NCR, Novell, Oracle, Qualcomm, Sun, Sybase, Synopsys, and Unisys. We excluded IBM from the list because (as explained above) we did not review IBM's patents to divide the software patents from the nonsoftware patents.

having low quality and value indicators are characteristic of patents in general.

Attempting to investigate the question more precisely, we then analyzed the software patents of superpatentee firms on a sector-by-sector basis, grouping the firms into electronics firms, prepackaged software firms, and system design and processing firms.[95] Generally, as Table 9 summarizes, our data buttress the discussion above, indicating that the distinctive software patents are most likely to be found among the relatively pure prepackaged software firms. Specifically, we find for those firms robust positive differences related to the numbers of total claims and independent claims, with marginally significant negative differences related to the number of U.S. patent references and foreign patent references.

Putting that narrow finding to one side, we are struck by our inability to find substantial differences in quality and value based purely on the size of the firms. As we discuss below, we think this has important implications for the course of patent reform.

D. Patent Quality and Value over Time

Our last inquiry into patent quality and value relates to the quality of the relevant patents over time. If the focus of the concern about quality is that the quality of software patents has been degrading rapidly since the legal environment became more conducive to them in the mid-1990s, a discussion of the quality of software patents must include some information about how their quality has changed over time.

Because the data set described above is limited to patents issued between 1998 and 2002, we used for this analysis a separate data set of all patents issued to the fifteen superpatentee firms, based on applications filed since 1990. This allows us to collect evidence about how our various indicators of patent quality changed during the 1990s for the largest firms in the three major sectors of large patentees discussed above.[96] To be sure, this data set is much less inclusive than the data set we analyze above. Moreover, this data set does not focus solely on software patents, because we did not individually separate software from nonsoftware patents. Still,

---

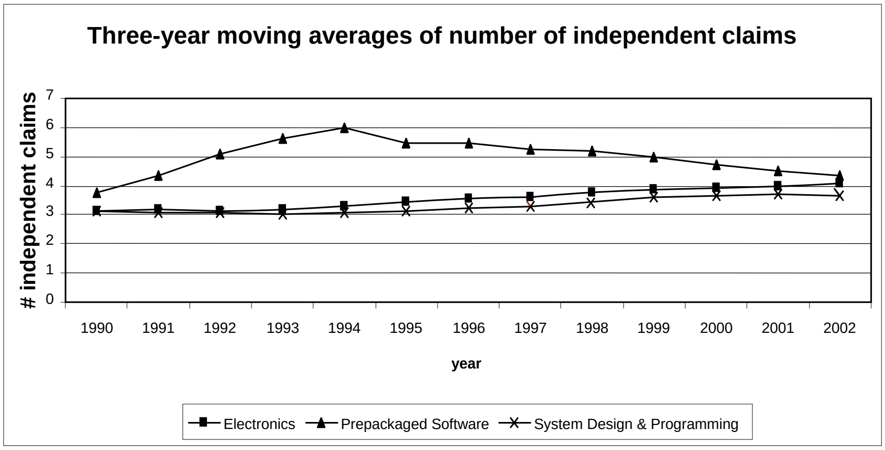
95. The firms are distinguished by three-digit NAICS codes: 334 for the electronics firms (Apple, EMC, HP, NCR, Qualcomm, and Sun), 511 for the prepackaged software firms (Adobe, Autodesk, Computer Associates, Microsoft, Oracle, Sybase, and Synopsys), and 541 for the system design and processing firms (EDS, IBM, Mentor Graphics, Novell, and Unisys).

96. Due to concerns about truncation from as-yet-unissued patents, the analysis in this section ends with patents for which applications were filed in 2002.

it is likely that most of these patents disclose software inventions because of our limitation to the group of software superpatentee firms. Thus, this data set provides considerable evidence about trends in software patenting quality over time in the industry.

In general, the data suggest that the quality and value of patents did not degrade substantially during the 1990s.[97] For example, as Figure 1 shows, there has been a slow but steady increase in the number of independent claims for electronics and system design firms. By contrast, the number of independent claims in patents by prepackaged software firms rose much more rapidly until 1994, but has fallen since then to a level that approximates the level of the other sectors. It is apparent that events at the middle of the decade affected the patenting practices of prepackaged software firms (Figure 1, sector 2) differently than the practices of the large firms in the other sectors. A possible explanation is that legal rules making it easier to patent software inventions more directly lessened the need for circuitous claim drafting and thus lowered the number of claims necessary for a sophisticated patentee to describe a particular invention. In any event, none of the three sectors had fewer independent claims in 2002 than in 1990.[98]

FIGURE 1: INDEPENDENT CLAIMS OVER TIME



97. For a typical criticism of the PTO's work during this period, see ADAM JAFFE & JOSH LERNER, INNOVATION AND ITS DISCONTENTS: HOW OUR BROKEN PATENT SYSTEM IS ENDANGERING INNOVATION AND PROGRESS, AND WHAT TO DO ABOUT IT 142–50 (2004).

98. Data on total claims show a similar pattern.

Figures 2 and 3 summarize parallel data on total prior art references and U.S. patent references, which show a steady rise throughout the 1990s, with the patents of prepackaged software firms trending upward substantially more rapidly than those of firms in the other sectors. Again, it is clear that the time pattern of the patents of prepackaged software firms is different than the pattern of other firms' patents, but the pattern suggests, if anything, that those patents gained in references even more rapidly than the patents of firms in other sectors. Presumably this is because of the rapid increase in innovation in that sector, which provided a steady increase of relevant prior art appropriate for citation.

FIGURE 2: TOTAL REFERENCES OVER TIME

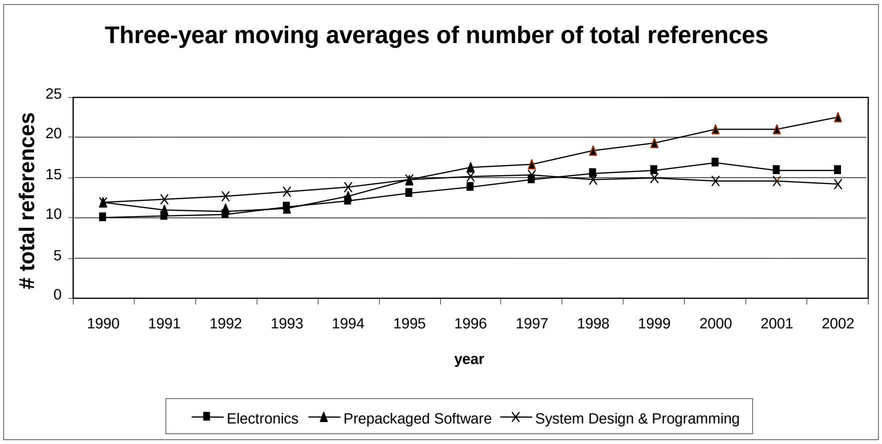
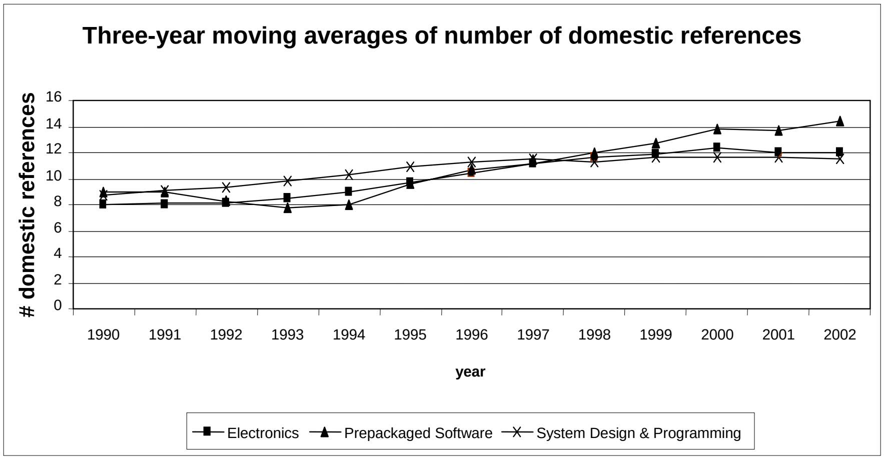


FIGURE 3: DOMESTIC REFERENCES OVER TIME



We emphasize that these time patterns do not prove that the patterns of prepackaged software firms were "better" than the patents of firms in the other sectors. These results do make it harder, however, to credit the common assertion that software patents declined rapidly in quality during that period.

IV. IMPLICATIONS AND CONCLUSIONS

As the discussion above emphasizes, our work is suggestive. The complexity of the questions that we investigate makes definitive answers almost impossible. Thus, for example, although we believe that our definition and method of identifying a software patent is better than those used by other researchers in the existing literature, we recognize that others may find other approaches more appealing, largely because of the huge opportunity costs associated with manual examination of large data sets of individual patents. However, the difficulty other scholars will find in using our definition makes this work all the more valuable, because we can provide an unparalleled examination of patents on software inventions based on a patent-by-patent review.

In our view, the data are important for two separate reasons. First, the data substantially undermine the traditional story that large firms in the software industry are plagued by a large number of low-quality patents obtained by the smaller firms in the industry. On the contrary, by objective standards, software patents as a group compare quite favorably to patents that the same firms are obtaining, at the same time, on nonsoftware inventions. Similarly, the patents obtained by small firms are no worse than the patents of the large firms.

These findings have in our view twin implications for patent policy. The first is the simplest: they undercut the common suggestions that software patents should be prohibited entirely or should face special hurdles for examination designed to stem the alleged flood of low-quality patents. If, in fact, there is no flood of low-quality patents, then there is little reason to take aggressive action to respond. The second implication is more speculative, but rests on the idea that the effort and investment of resources to obtain stronger and broader patents does not depend substantially on the applicant's size or patenting experience. To the extent that this is true, our findings undermine the concern that small firms will suffer disproportionately from reforms that raise the bar for patent grants, such as increased examination fees, special procedures for "gold-plated" patents, or additional opportunities for pregrant opposition. If patent drafting is a routine exercise in which firms of all sizes do a better (or worse) job based on the incentives that the PTO's procedures present, then this presents a reasonable case for reforms designed to reward applicants that put more effort into their application or who are willing to provide more credible support for their application (as evidenced by a willingness to submit their application to a more onerous process).

In addition to their policy implications, our findings provide a rare empirical illustration of the context of patent drafting. For example, the findings on prior art references seem to match up well with anecdotal understandings about the software industry indicating that, until recently, there were relatively few U.S. patents and very few foreign patent references available but a relatively large amount of NPPA. First, the centrality of U.S. firms to cutting-edge software innovation provides emphatic support for the finding that software patents cite relatively few foreign patent references. Similarly, our early point in the era of routine software patenting suggests that, at least as compared to other fields of technology, the balance of patent and nonpatent prior art should be weighted more heavily in favor of NPPA.

Second, our findings regarding the number of total and independent claims rest at least in part on techniques of software patent drafting. This is particularly true in the case of pure software patents, which cover those inventions that consist solely of software rather than inventions in which software is merely a critical part. Allison's examination of tens of thousands of patents over the last decade persuades him that software patent applicants are more likely than other patent applicants to claim software inventions in duplicative ways within a given patent. Therefore, for example, software patents often include separate sets of claims that characterize the invention as a method (or process), a machine or apparatus (or "device"), and a "system."[99]

We posit two possible explanations. First, this phenomenon could depend on habits developed in earlier years, when doubt about the patentability of pure software made it important for the patent to show some "physical transformation." In that era, it was common for software patents to claim as machines or devices with terminology from the older mechanical and electronic arts, often describing a phantom physical structure in the specification and then claiming the invention as a "means" for performing certain functions.[100] Later, when it became clear that pure software was patentable, applicants began to claim software inventions directly as methods or systems, but often they still included the older claiming formats.[101] Moreover, even now, many modern software patents do cover inventions that have physical elements as well as software elements. In those cases, it is quite natural to use a number of different claim formats in the same patents.[102] To be sure, patents on other types of inventions often claim an invention using two or more claim formats, such as a device and a method, but the tendency is more pronounced for software patents, likely accounting for a portion of the finding that software patents have more total and independent claims.

Second, by its nature, a software invention can be conceptualized in more ways than many other types of invention. Unlike inventions in most other fields, it is not unusual to simultaneously regard software as a method or process, a machine or device, a system, and a means for performing specified functions.

Third, the relatively high number of forward citations could partly reflect the status of software as a hot area for innovation; it is likely that there will be more patents issued in technological areas related to software than in other technological areas. This phenomenon would lead, in due course, to a substantially greater number of forward citations for software patents than for nonsoftware patents.

---

99. *See, e.g.*, Item Removal System & Method, U.S. Patent No. 5,886,634 (filed May 5, 1997).

100. *See, e.g.*, Raster Scan Waveform Display Rasterizer with Pixel Intensity Gradation, U.S. Patent No. 5,440,676 (filed Jan. 29, 1988).

101. *See, e.g.*, Hot Spot Analysis of IMS Databases, U.S. Patent No. 6,418,443 (filed Feb. 1, 2000).

102. *See, e.g.*, Scanned Retinal Display with Exit Pupil Selected Based on Viewer's Eye Position, U.S. Patent No. 6,352,344 (filed Feb. 14, 2001).

If those features of the software patenting environment lead directly to the relatively high quality and value indicators of the patents in our data set, then scholars will need to proceed with care before adapting the existing "valuable patent" methodology to studies of patent quality in particular industries—a methodology honed for the most part on economy-wide studies of large data sets. To make comparative judgments about patent quality in particular industries, it is necessary to develop quantitative measures of the environment for patenting as it exists in each industry. In this context, however, the juxtaposition of findings persuades us that the best explanation of our data is the optimistic one. It is possible for the skeptic to explain away the significance of each of the separate quantitative findings about software patents, as the last few pages have attempted to do. There comes a point, though, when the need for so many complex explanations suggests that the simpler explanation is the better one: software patents as they have been issued in this country to software firms starting in the late 1990s in fact display impressive objective indications of quality and value.

STATISTICAL APPENDIX

A. Software and Nonsoftware Patents

To test the robustness of the simple t-tests that we report in the text, we introduced a number of controls. Our analysis treats our seven indicators of quality and value as alternative dependent variables. For each of those variables, we conducted a series of multivariate linear regressions that included several controls in addition to the software patent/nonsoftware patent variable: year dummies (to account for the changes in the relevant characteristics over time),[103] industry sector dummies (to account for the differences in technology in different sectors of the software industry), and a pure-software firm dummy (that distinguishes between firms that obtain 80% or more of their revenue from software and those that have substantial nonsoftware product lines). To control for problems of autocorrelation, we cluster the standard errors for each firm.[104]

103. We attempted to analyze the nature of the changes over time, but the coefficients and t-statistics on the dummies for the individual years were unstable and did not display any obvious pattern. Accordingly, we do not report that information here.

104. *See* W.H. Rogers, *Regression Standard Errors in Clustered Samples*, 13 STATA TECHNICAL BULLETIN 19 (1993).

The first two columns of Table 5 report the results of those regressions (the coefficient on the SWP variable as an independent variable) for each of the different dependent variables. They confirm the central results in Table 3 with respect to claims, independent claims, foreign references and other references. The results for adjusted forward citations do not hold up as consistently in the various regression models. We do not weigh that analysis heavily, however, because of the likelihood that the combination of the truncation of those data and our Jaffe-Trajtenberg[105] adjustment have diluted the ability of our data to provide information on that question. With respect to prior art references in the patents, we note that the models discussed below suggest that the significance of the number of total prior art references and U.S. patent references is unstable. Still, the number of foreign patent references is significantly lower in software patents and the number of nonpatent references is significantly higher (both as in Table 3 and as discussed in the text).[106]

TABLE 5: SOFTWARE PATENT REGRESSION MODELS

DEPENDENT VARIABLE	LINEAR COEFF.	LINEAR T-STAT.	XTREG FE COEFF.	XTGREG FE T-STAT.
TOTAL CLAIMS	1.87	4.09	1.98	7.50
INDEPENDENT CLAIMS	0.40	3.35	0.44	8.73
ADJ. FORWARD CITATIONS	0.09	1.13	0.06	2.18
TOTAL REFERENCES	2.31	2.81	2.69	5.41
U.S. PATENT REFERENCES	0.04	0.04	0.21	0.68
FOREIGN PATENT REFERENCES	–0.50	–2.45	–0.49	–10.35
NONPATENT REFERENCES	2.77	5.28	2.97	10.47

105. *See supra* note 83 and accompanying text.

106. We conducted similar regressions to consider whether IBM patents (almost half of our data set) differ from patents held by other firms. Those regressions suggested that IBM patents generally have fewer claims, references, and forward citations. We do not weigh those results heavily (and do not report them here), because they seem to reflect the fact that the share of IBM's patents that are software patents (55%, by our estimate) is smaller than the share of the patents of other firms in our data set that are software patents (68%).

We next estimated a firm-level fixed-effects model (using the xtreg function in Stata). This truncates our data considerably, because it analyzes only the patents of those firms that have both software and nonsoftware patents. Nevertheless, as the third and fourth columns of Table 5 report, those regressions produced results quite similar to those from the simple linear regressions reported in the first two columns of Table 5. To control for skewed data distributions, we also conducted a parallel set of regressions using the log of the dependent variables and Poisson regressions. (We use Poisson regressions rather than logs for the data on adjusted forward citations because many of the data points are zero.) Table 6 reports those results, which are quite similar to the results in Table 5. Finally, although we do not report them here, we also estimated parallel models controlling for both national and international patent classes. The results are similar to those in Tables 5 and 6.

TABLE 6: CONTROLS ON SOFTWARE PATENT REGRESSIONS

DEPENDENT VARIABLE	COEFF. (T) ON LOG OF DEPENDENT VARIABLE	XTREG FE COEFF. (T) ON LOG OF DEPENDENT VARIABLE	POISSON COEFF. (Z)	XTPOISSON FE COEFF. (Z)
TOTAL CLAIMS	0.10 (4.33)	0.11 (8.27)	--	--
INDEPENDENT CLAIMS	0.13 (4.87)	0.13 (11.71)	--	--
ADJ. FORWARD CITATIONS	--	--	0.10 (1.15)	0.05 (2.76)
TOTAL REFERENCES	0.05 (0.58)	0.07 (4.27)	--	--
U.S. PATENT REFERENCES	–0.009 (–0.13)	–0.002 (–0.12)	--	--
FOREIGN PATENT REFERENCES	–0.14 (–4.08)	–0.13 (–4.79)	--	--
NONPATENT REFERENCES	0.23 (4.00)	0.30 (9.48)	--	--

B. Nonpatent Prior Art (NPPA)

When there was any doubt about how to categorize a particular reference, we thoroughly searched the internet to achieve a high degree of confidence about the appropriate classification. Our nine categories of NPPA are as follows:

(1) *Academic Publications*: This category represents publications of a type for which there is an *independent intermediating influence* such as one or more editors or referees to increase the probability of accuracy, reliability, and objectivity, and which are targeted primarily at an *academic, scholarly* audience. The primary components of this category are academic books, book chapters, journal articles, and academic proceedings papers, all of which have been independently screened for accuracy and objectivity.

(2) *Trade Publications*: This category includes trade books and chapters, trade journal articles, and similar items. Trade publications are targeted primarily at a *practitioner* audience rather than an academic one, and *report on* developments in a field rather than create new knowledge in that field as academic works are more likely to do. Like academic publications, trade publications are a type of nonpatent prior art for which there is an *independent intermediating influence* such as one or more editors or referees to increase the probability of accuracy and objectivity. Although these publications are quite unlikely to be subject to the same degree of rigorous peer review as academic publications, they nevertheless constitute prior art of relatively high quality and are a good reflection of the state of the art at the time of publication.

(3) *University Publications*: This category includes publications from universities or consortia of universities, such as those from university research labs, departments (such as computer science, electrical engineering, information systems, business, etc.), individual faculty, and graduate student theses/dissertations. Because these types of publications are developed in an environment of objective academic inquiry, they typically will be prior art of good quality, although this quality is probably quite variable.

(4) *Software*: This category includes software programs and software documentation. These are separated from other company- or industry-sponsored publications because of their functional nature and obvious need for a high degree of accuracy and objectivity compared with less functionally motivated company-sponsored prior art. Software and software documentation therefore represent prior art of comparatively high quality.

(5) *Patent-Related*: This category includes published patent applications and patent office search reports, such as PCT (Patent Cooperation Treaty) and EPO (European Patent Office) search

reports. Such publications are likely to be of highly variable quality as prior art. Published patent applications are of uncertain quality as prior art because they have not yet been examined or otherwise tested. Published search reports are likely to be more objective and reliable than published applications because of the involvement of independent search authorities.

(6) *Government Documents*: This category includes documents published by U.S. and foreign governments and by international government organizations such as the World Intellectual Property Organization (WIPO), as well as websites sponsored by such entities. The category does not include U.S. and foreign patent-related documents such as published patent applications and search reports, which are treated separately because of their special nature. The quality of government documents as prior art is likely to be extremely variable.

(7) *Company/Industry Publications*: This category includes press releases, websites, advertisements, technical disclosure bulletins, and various other publications that were produced by individual companies or industry groups and published with *no independent intermediating influence* to increase the probability of accuracy and objectivity. It does not include software and software documentation, however, because the latter are sufficiently distinct from and inherently more reliable than other types of publications from companies or industry groups. After removing software and software documentation from the category, company- and industry-sponsored publications overall cannot be treated as high quality prior art.

(8) *Popular Press*: This category includes not only newspapers, magazines, and other publications of general interest, but also news publications aimed at general business and legal audiences. The relative quality of such publications varies greatly but overall is relatively low.

(9) *Other*: This category includes sundry items such as individual websites, but most references placed in this category are those in which insufficient information was provided for determining what the item really was, even after we conducted a thorough internet search of key names and terms in the incomplete reference. One example is a reference to a partial title of an item, followed by "found on the web on *x* date."

C. Pure and Mixed Software Firms

TABLE 7: PURE AND MIXED SOFTWARE FIRMS

DEPENDENT VARIABLE	SW_SWF COEFF. (T)	NONSW_SWF (T)
TOTAL CLAIMS	4.04 (6.76)	0.18 (0.10)
INDEPENDENT CLAIMS	0.45 (2.17)	–0.05 (–0.19)
ADJ. FORWARD CITATIONS	0.19 (3.29)	0.004 (0.12)
TOTAL REFERENCES	2.45 (1.65)	–0.34 (–0.21)
U.S. PATENT REFERENCES	0.71 (0.78)	–1.96 (–1.42)
FOREIGN PATENT REFERENCES	–0.16 (–0.93)	–0.024 (–0.51)
NONPATENT REFERENCES	1.90 (3.18)	1.86 (2.44)

The results here reflect linear regressions on our seven dependent variables by comparing software patents obtained by pure software firms to software patents obtained by mixed software firms—controlling, as in Tables 4 and 5, for year and sector and clustering on the individual firm. The first column compares software patents obtained by pure software firms (SW_SWF) to software patents obtained by mixed software firms. The second column compares nonsoftware patents obtained by pure software firms (NONSW_SWF) to nonsoftware patents obtained by mixed software firms.

D. Superpatentees and Other Software Firms

TABLE 8: SUPERPATENTEES AND OTHER SOFTWARE FIRMS

DEPENDENT VARIABLE	LINEAR COEFF. (T)	LOG COEFF. (T)	POISSON COEFF. (Z)
TOTAL CLAIMS	–1.21 (–0.65)	–0.01 (–0.08)	--
INDEPENDENT CLAIMS	0.151 (0.49)	0.047 (0.67)	--
ADJ. FORWARD CITATIONS	–0.188 (–1.66)	--	–0.163 (–1.68)
TOTAL REFERENCES	–4.24 (–1.49)	–0.185 (–1.92)	--
U.S. PATENT REFERENCES	–3.65 (–1.73)	–0.197 (–2.42)	--
FOREIGN PATENT REFERENCES	–0.067 (–0.33)	–0.067 (–1.11)	--
NONPATENT REFERENCES	–0.529 (–0.60)	–0.120 (–1.09)	--

The results here reflect regressions on our seven dependent variables comparing software patents obtained by superpatentee firms to software patents obtained by other software firms—controlling for year and clustering on the individual firm. The first column reflects simple linear regressions. The second reflects regressions on the log of the dependent variable. As in Table 6, we use a Poisson regression (reported in the third column) for adjusted forward citations, because of the large number of data points that are zero.

TABLE 9: PREPACKAGED SOFTWARE SUPERPATENTEES

DEPENDENT VARIABLE	LINEAR COEFF. (T)	LOG COEFF. (T)	POISSON COEFF. (Z)
TOTAL CLAIMS	3.34 (2.71)	0.212 (4.14)	--
INDEPENDENT CLAIMS	0.85 (3.39)	0.209(3.87)	--
ADJ. FORWARD CITATIONS	–0.064 (–0.61)	--	–0.053 (–0.62)
TOTAL REFERENCES	–3.45 (–1.19)	–0.065 (–0.76)	--
U.S. PATENT REFERENCES	–3.60 (–1.77)	–0.152 (–2.19)	--
FOREIGN PATENT REFERENCES	–0.34 (–2.00)	–0.095 (–1.73)	--
NONPATENT REFERENCES	–0.49 (–0.52)	–0.064 (–0.77)	--