

# The Private Value of Software Patents

Bronwyn H. Hall and Megan MacGarvie<sup>1</sup>

April 2007

## *Abstract*

We investigate the value creation or destruction associated with the introduction of software patents in the United States in two ways. The first looks at the cumulative abnormal returns to Information and Communication Technology (ICT) firms around the time of important court decisions that impacted software patents, and the second analyzes the relationship between firms' stock market value, the sector in which they operate, and their holdings of software patents. We find that the extension of patentability to software was initially negative for software firms, especially for those producing application software or services. However, we also find that the ownership of software patents is positively and significantly associated with Tobin's Q, and that the market's valuation of software patents increased following changes in the USPTO's treatment of software patents in 1995.

---

<sup>1</sup> Hall: 549 Evans Hall, Berkeley, CA 94720, [bhhall@econ.berkeley.edu](mailto:bhhall@econ.berkeley.edu). MacGarvie: 595 Commonwealth Ave., Boston, MA 02215, [mmacgarv@bu.edu](mailto:mmacgarv@bu.edu). This is a revision of a paper written in April 2005. We thank Anne Layne-Farrar and LECG for the use of the Corptech data, and we thank Jim Bessen for helpful suggestions and for supplying us with the list of software patents used in Bessen and Hunt (2003). We are also grateful to Stu Graham, Bala Iyer, Lucia Silva, numerous participants in the Empirical Patent Research Conference, St. Helena, California, February 2005, the IIOC, Atlanta, April 2005, the ZEW Conference on Patents and Innovation in Mannheim, September 2005, the Seventh CEPR Conference on Applied Industrial Organization, Madeira, May 2006, the NBER Summer Institute 2006, the First Annual EPIP Conference, Munich, September 2006, and seminars at UCSD, University of Kansas, University of Michigan, University of Maastricht, LMU-Muenchen, and RIETI, Tokyo for extremely helpful comments.

# **The Private Value of Software Patents**

**Bronwyn H. Hall and Megan MacGarvie<sup>2</sup>**

**March 2006**

## ***1. Introduction***

Ever since software became generally patentable in the United States in 1995, the wisdom of such a change has been widely debated. Proponents of the change argue that there is no statutory reason to exclude software (or computer-implemented) inventions from patentability, and also that patenting software has social benefits from disclosure and from the fact that it enables software components to be reused by others (Cohen and Lemley 2001). Critics, who are numerous, base their arguments on a series of considerations: that such patents have often been of low quality, that they discourage rather than encourage innovation, and that they have a negative effect on the growing open source model of innovation (Kasdan 1994; Barton 2000, 2002; Hunt 2001; Bakels and Hugenholtz 2002; Lunney 2001; Dreyfuss 2000, 2001; Meurer 2002). In a recent article in PC magazine, the columnist John Dvorak argued that software patenting is even bad for Microsoft, the largest patent holder among the pure software firms (Dvorak 2005). In Europe, patentability of software is still being debated, with no clear conclusion emerging (Blind et al. 2001; European Commission 2002; Spindler 2003).

---

<sup>2</sup> UC Berkeley, University of Maastricht, NBER, and IFS London; Boston University and NBER.

The economic view of the patent system sees this debate as an inevitable one, given the nature of patents: with a patent grant we trade off short term exclusive (monopoly) rights to the use of an invention in return for two things: 1) an incentive to create the innovation; and 2) early publication of information about the innovation and its enablement. The argument is that without the patent system, fewer innovations would be produced, and those that were produced would be kept secret as much as possible to protect their returns from appropriation by others. In considering the economic impacts of a subject matter extension to software and business methods, the tradeoff between these benefits and the welfare cost of the grant of a monopoly right are at least as important as they are in any other technological arena.

Recent analysis also says that although *competition* may suffer when we grant a monopoly right to an inventor, it will benefit if this right facilitates entry into the industry by new and innovative firms (Hall 2003)<sup>3</sup>. Second, *innovation* will benefit from the incentive created by a patent but may suffer if patents discourage the combining and recombining of inventions to make new products and processes (Scotchmer 1991, 1996; Heller and Eisenberg 1998). There are several reasons to think that facilitating entry (the benefit of patents for competition) and discouraging recombination of elements (the cost of patents for innovation) may be particularly salient factors when considering the effects of patentability on the software industry. The software sector is characterized by low tangible assets and high intangible assets, which means that it is more important to have

---

<sup>3</sup> Cockburn and MacGarvie (2006) find evidence that firms holding patents are more likely to enter software markets. However, increases in the extent of patenting by other firms in the market reduce the rate of entry. Hall (2005) finds that firms in the ICT sector that enter the publicly traded sector after changes in patenting behavior in this sector in the mid-1980s have much higher patent valuations than those that enter earlier.

intellectual property rights in order to secure funding for investment. Software itself is typically cumulative and composed of a number of components which may come from a number of different sources, and where interoperability standards are important. Both factors mean that combination and recombination of inputs is an essential part of software creation.

Evaluating the tradeoff between the benefits and costs of the introduction of software patents is a formidable, perhaps impossible, task. It is not made easier by the fact that the most important expansion of software patentability occurred in 1994-1995, exactly coincident with the internet revolution and the beginnings of its impact on the global economy's use of software for a wide range of new applications.<sup>4</sup> To evaluate the impact of software patentability properly would require an examination of a counterfactual world without software patents and measurement of the differential impact on welfare and innovative activity in existing firms, new entrants, downstream buyers, and consumers. In this paper we look at one ingredient of the overall problem: the private returns to established computer hardware and software firms from the expansion of software patentability. If the patentability of software is beneficial for the economy, we expect that such firms will experience positive returns to its introduction, even if the overall economic effect is neutral or negative. That is, a finding that existing firms do not

---

<sup>4</sup> It is possible that this timing is not a coincidence. The expansion of patentability has often come about as a result of the expansion of economic activity in a particular area. E.g., see Murmann (2003) on the chemical industry in Germany and Britain in the late 19<sup>th</sup> century. However the rapidity with which attempts to patent software followed on the introduction of the internet makes the causal explanation improbable. It seems more likely that the changes in software patenting in the mid-nineties were driven by the introduction of the personal computer in the mid-eighties and the consequential increased importance of stand-alone software firms.

benefit financially from the introduction of software patents makes it probable that the impact on the economy as a whole is not positive either.

Existing research on software patents reveals a dramatic increase in the propensity to patent software over the last two decades, and argues that this indicates that holding patents on software has become both easier and more valuable to firms (Bessen and Hunt 2003). In this paper we explore the relationship between firm value and the patenting of software in two different ways. First we conduct a series of event studies that look at the immediate market impact of changes in the patentability of software on firms in sectors where software patenting is prevalent.<sup>5</sup> Second, we use the methodology of Hall, Jaffe, and Trajtenberg (2005) to estimate the market value of software patents using various patent-related measures and we explore the effect of changes in the patentability of software over time on the market value of publicly traded firms that hold such patents.<sup>6</sup>

In performing these analyses, we separate the software firms in the sample into those that are “upstream” and those that are “downstream,” which corresponds to the position of their products in the computing stack, from the basic operating system layer to end user application software products (Raduchel 2006). We hypothesize that firms producing software that relies on hardware or software that is upstream from their software in order to operate may face more negative consequences than other firms from

---

<sup>5</sup> Based on a review of the industry sectors where firms obtain software patents described later in the paper, we identified the relevant two-digit industries as 35 (machinery including computing equipment), 36 (electrical machinery, 38 (instruments), 48 (telecommunications), and 73 (business services including software).

<sup>6</sup> Noel and Schankerman (2006) perform a similar analysis of the market value effects of software patents, and obtain conclusions similar to ours. They also study the relationship between market value and measures of the cost of transacting with rival patenters, and conclude that market value is enhanced when rivals’ intellectual property is concentrated among fewer parties.

the introduction of software patents (at least initially), because they are more likely to need licenses for patented software technology in order to ensure that their products will operate successfully in a particular computer environment.

The paper is organized in the following way: the next section reviews the recent history of software patenting in the United States. This is followed by a discussion of our data, which consists of a panel of publicly traded firms in the ICT sector; we also spend some time evaluating a number of definitions of “software” patents that are based on keyword searches and patent classification systems. The next two sections present our empirical results: first a series of studies of the immediate stock market impact of software patentability decisions in the courts, and then the results for the relationship between firm market value and the ownership of patents. The latter results are presented for two subperiods, before and after the general acceptance of software patenting by the USPTO guidelines. The final section concludes.

## **2. *Background***

The law concerning the patentability of software in the United States has evolved through a series of decisions following the passage of the 1952 Patent Act (which did not exclude any subject matter statutorily), to the point where algorithms may be patented if “there is practical application for the algorithm or if it is associated with a tangible medium (Sterne and Bugaisky 2004, p. 221). Although a complete history of the use of intellectual property protection in the software industry is beyond the scope of this paper, we provide a brief summary here and refer the reader to Graham and Mowery (2003) for more detail.

In 1972, the Supreme Court's ruling in *Gottschalk v. Benson* held that because software is essentially a collection of algorithms, it could not be patented. In 1980, Congress confirmed that the appropriate intellectual property protection for software was copyright (Samuelson 1984). However, in 1981 the court allowed for patenting of software tied to physical or mechanical processes, such as the program implemented in the method for curing rubber at issue in *Diamond v. Diehr*, a decision which seems consistent with the present day European Patent Convention.<sup>7</sup> However, this and subsequent decisions by the court during the next 15 years led the US Patent and Trademark Office (USPTO) to modify its position with respect to software patents, ultimately to one that allowed them even when not embodied in a physical process. As Figures 1 and 2 show, the growth of software patenting increased from a 5-10 per cent level prior to 1981 to a 15 per cent level afterwards, and then increased again in the mid-1990s, as the internet boom took off and the USPTO issued new guidelines for software patenting in response to a series of court rulings described later in the paper.

Following the *Diamond v. Diehr* decision, the software patenting standard was substantially unchanged until the early 1990s when the rise of the personal computer industry and the consequent vertical disintegration of the computing sector encouraged software and information producers to test the subject matter exclusions with respect to

---

<sup>7</sup> The European Patent Convention's (EPC) treatment of patentable subject matter differs markedly from that of the United States Patent Act. Whereas in the US, the 1952 Patent Act opened the door to judicial expansions of patentable subject matter by not including any explicit limits on statutory subject matter, Article 52 of the EPC expressly excludes several categories of inventions, among them mathematical methods, schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers "to the extent to which a European patent application or European patent relates to such subject-matter or activities *as such*" (EPO 1989, emphasis added). In the case of software, the phrase "as such" has led to a number of software inventions actually being patented via their embodiment in hardware. See Bakels (2005) for a detailed discussion of the evolution of European policy in this area.

software again. In 1993 Compton Encyclopedia attempted to enforce a broad patent on the display of multi-media content on CD-ROMs against several firms; in response to complaints by these firms, the USPTO re-examined and revoked the patent in March 1994, which halted assertion of this particular software patent.

But then later in 1994 an important change occurred when the Court of Appeals of the Federal Circuit (CAFC) stated (*In re Alappat*) that unpatentable software was that which represented “a disembodied mathematical concept...which in essence represents nothing more than a ‘law of nature,’ ‘natural phenomenon,’ or ‘abstract idea.’” Software that could be patented was “rather a specific machine to produce a useful, concrete, and tangible result.”<sup>8</sup> A series of additional decisions, described in more detail in section 4 of this paper, culminated in 1995 with *In re Beauregard*, in which the CAFC ruled that the Patent Office’s rejection of a software patent application by IBM was erroneous. The Commissioner of Patents then issued new guidelines on the patenting of software, which allowed inventors to patent any software embodied in physical media (which essentially reversed the *Gottschalk* decision). (Sterne and Bugaisky 2004, p. 223) These decisions presumably lie behind the huge increases in software patenting during the 1996-1999 period shown in Figures 1 and 2.

A second important expansion of patentability took place in 1998 when Judge Rich issued the famous opinion in *State Street Bank and Trust v Signature Financial Corporation*.<sup>9</sup> The Signature patent at issue was a “pure” number-crunching type of

---

<sup>8</sup> *In re Alappat*, 33 F.3d 1526, 1544 (Fed. Cir. 1994), cited in Sterne and Bugaisky (2004), p. 222.

<sup>9</sup> *State Street Bank and Trust Co., Inc. v. Signature Financial Group, Inc.*, 149 F.3d 1368 (Fed. Cir. 1998). See Lerner (2002) for a fuller discussion of this case.

application, which implemented financial accounting functions. The Federal Circuit Court decision stated clearly that Section 101 (the section of the patent code that deals with subject matter for patentability) is unambiguous - “any” means ALL, and it is improper to read limitations into 101 not intended by Congress. Therefore, mathematical algorithms are non-statutory only when “disembodied” and thus lacking a useful application. The court went on to make sure that the decision was precedent-setting by stating that with regard to the business method exception, “We take this opportunity to lay this ill-conceived exception to rest.”

This decision was followed by an increase in applications for “business method” patents, most of which are arguably also software patents, because they describe the implementation of a business method on a computer or the internet. However, they are still a small fraction of all software patents (Hall 2003), at least when we use the relevant patent class (705) to define them. Most of the analysis in this paper uses data that ends in 1999 or earlier, so these patents will play a relatively small role.

## **2.1 The critique of software patentability**

Much of the critique of software patents centers on the perception that such patents are often of low quality, rather than the principle that software should never be patentable. The “quality” of a patent is a somewhat ill-defined catchall term for all the characteristics that the particular analyst would like patents to have. Hall (2003) and Hall et al (2003) review these criteria and argue why the failure of issued patents to satisfy them may be costly for firms and society. These criteria include satisfying the statutory

definition of a patentable invention (novelty, non-obviousness, and utility),<sup>10</sup> sufficient disclosure to enable those “skilled in the art” to understand the invention, and relatively little uncertainty over the validity of the patent. A number of legal and technical scholars have critiqued the software patents issued after the series of court decisions in 1994-1995 on all of these grounds.

First, because software development took place over a long period before software was patentable, when the USPTO began to handle large numbers of patents in this area in 1994-1995, they did not have examiners with the relevant training and lacked adequate data bases with software prior art. For example, on February 7, 1995, the following exchange between a patent examiner and the editor concerning access to non-patent prior art took place in Aharonian’s Patent Newsletter:<sup>11</sup>

The examiner: “The problem with obviousness is evidence. When an examiner uses common sense, attorneys scream hindsight. Also, a problem with ordering non-patent publications or translations of foreign documents is the time it takes to process these requests. An examiner cannot simply call a company whose making, or is believed to have made, a product which appears to infringe on a claim. At best, the examiner could ask a librarian at our library to call a company to request info, but again that takes time. With ten hours to do a case, movement is paramount.”

---

<sup>10</sup> See Lunney (2001) for an argument that the non-obviousness test has been weakened since the creation of the Federal Circuit Court of Appeals in 1982.

<sup>11</sup> Aharonian (1995). <http://www.bustpatents.com/>

Aharonian: “Additionally for subjects like software, the cost of purchasing copies of technical papers would exceed the application fee, so I doubt many examiners would get the authority to spend such sums. Since for most software patent applications, the most relevant prior art is non-patent materials, between the statistics I cited on citing prior non-patent prior art (an average of two out of about 30) plus the above and below comments, one could make a good case that it is impossible for the PTO to conduct adequate novelty analyses.”

The result of lack of access to adequate prior art was that many poor quality software patents were issued (Barton 2001; Kasdan 1994, 1999; Lunney 2001).

Second, this sector is an extreme case of cumulative innovation, where one person’s invention depends on those of many other people and the transactions costs associated with licensing a large number of patents for any particular software product may exceed the profits attainable from its creation and therefore discourage innovation altogether. Third, some claim that the disclosure function of patents is particularly badly served by software patents, which rarely include the source code implementation and are often quite vaguely and broadly worded. Even in the case of patents in general, both Cohen et al. (2002) (using US survey data) and Arundel et al. (2002) (using European survey data) report that firms rank a number of other means of acquiring information such as customers, exhibits, conferences, journals, suppliers, competitors, and nonprofit institutions ahead of patent disclosures as technical information sources. Furthermore, in contrast to other industries, it has been argued that patents on software do not generally serve to diffuse information about the patented technology. Mann (2004) states that:

“It is clear that [disclosure] is an important benefit of patents in some industries, although the software industry in its current form is probably not one of them...As others have noted, with software cases the Federal Circuit has interpreted the disclosure requirement in such a way as to minimize the likelihood that disclosures will require information that is directly useful to competing inventors. Moreover, given the rapid pace of innovation, it will not often be the case that information disclosure in a patent application filed years earlier will be of immediate value to competitors in the industry.”<sup>12</sup>

Finally, the growing open source movement has been extremely critical of such patents, because of the obvious fact that coexistence of open source with patented software is problematical. When independent invention is not a defense against infringement (as in the case of patents), a developer that issues a GPL or copyleft license to others on code that he has developed cannot be sure that he has the right to do so. Thus the presence of patents on some innovations implemented via software may foreclose many avenues of open source development, unless low cost methods of licensing are developed.<sup>13</sup>

### **3. Data**

To perform our study, we combine data on publicly traded firms and their market value for the 1975-2002 period with a version of the NBER patent and citations dataset

---

<sup>12</sup> Mann (2004), p. 49-50

<sup>13</sup> For more on the debate on software patents and open source, see Evans and Layne-Farrar (2004).

that has been updated to 2002.<sup>14</sup> The patent data are matched to Compustat data on firms in the following SIC categories: 35 (machinery including computing equipment), 36 (electrical machinery), 38 (instruments), 48 (telecommunications), and 73 (business services including software). This match was done using a version of Hall's Compustat-assignee match updated to include firms that were not included in the previous (1995) version.<sup>15</sup> Using this updated match, we are able to identify patenting entities associated with 1,290 of the 2,379 Compustat firms in the SIC classes listed above. Our dataset is an unbalanced panel, and Appendix A3 lists the number of firms and the number of software firms in the dataset in each year of the sample. Both series have increased during the period, but software firms have clearly become more important as a share of the ICT sector.

Table 2 contains summary statistics for the dataset. In order to identify the particular software area in which our firms operate, we have also incorporated information from the Corptech directory of technology companies on the type of software they produce (systems software, middleware, applications, or software-related services).<sup>16</sup>

### 3.1 Defining a software patent

One difficulty that all researchers in this area encounter is that the definition of a software patent is rather unclear (Layne-Farrar 2005). Many scholars in the area may feel

---

<sup>14</sup> The update is available at <http://emlab.berkeley.edu/users/bhhall>

<sup>15</sup> Details on the matching algorithm can be found at <http://emlab.berkeley.edu/users/bhhall/pat/namematch/namematch.html>

<sup>16</sup> We gather this information from Corptech's SOF product codes and the classification developed in Gao (2005), which is listed in the appendix to this paper.

that "we know one when we see one" but this is not a practical way to choose a particular set of patents out of the 3 million or so in our datasets. Although all patents are classified into a number of technology classes and simply choosing those patents that are associated with software classes might seem the desirable way to go, it is an unfortunate truth that many of the relevant classes are broad enough to contain both software and hardware patents, and some software patents end up classified in classes that do not appear to have anything to do with software at first glance.<sup>17</sup> For this reason, different researchers have taken a number of approaches to identify software patents. In early versions of this paper, we explored the use of several different definitions and finally settled on a combination of them.<sup>18</sup>

The results reported in the body of the paper were obtained using a definition of a software patent that combined three ways of defining software patents. The three definitions are that used by Graham and Mowery (2003), that used by Bessen and Hunt (2003), and one that we constructed based on the patent classes and subclasses that contain patents assigned to fifteen of the largest software firms (which we call Hall-MacGarvie). Our combined definition was the union of the set of patents in all relevant IPC and US patent classes (the union of Graham-Mowery and Hall-MacGarvie) intersected with the set of patents found using a keyword search of title and abstract (Bessen-Hunt).

---

<sup>17</sup> The four US patent classes with the largest number of patents assigned to our software firms are 382 (Image Analysis), 345 (Selective Visual Display Systems), 341 (Coded Data Generation or Conversion), and 700 (Data Processing: Generic Control Systems or Specific Applications), all of which can hold hardware as well as software applications. On the other hand, there are over 1000 such patents in 435 (Molecular Biology) and over 500 in 84 (Music).

<sup>18</sup> For comparison, results using each of the definitions are found in an appendix.

As a check, we compare the results of these automated classification systems with the sample of software and internet business method patents identified manually by John Allison via a reading of the claims and description in the patents (Allison and Lemley 2000; Allison and Tiller 2003). In the remainder of this section of the paper, we describe the definitions more fully and present some comparative statistics based on them.

Graham and Mowery (henceforth GM) identify as software patents those that fall in certain International Patent Classification (IPC) class/subclass/groups. Broadly defined, the class/subclasses are “Electric Digital Data Processing” (G06F), “Recognition of Data; Presentation of Data; Record Carriers; Handling Record Carriers” (G06K), and “Electric Communication Technique” (H04L).<sup>19</sup> Graham and Mowery selected these classes after examining the patents of the six largest producers of software in the U.S. (based on 1995 revenues) between 1984 and 1995. Patents in these classes account for 57% of the patents assigned to the hundred largest firms in the software industry.<sup>20</sup>

Bessen and Hunt (henceforth BH) define software patents as those that include the word “software,” or the words “computer” and “program,” in the description and/or specification. Patents that meet these criteria and also contain the words “semiconductor,” “chip,” “circuit,” “circuitry” or “bus” in the title are excluded, as they are believed to refer to the technology used to execute software rather than the software

---

<sup>19</sup> The groups included are G06F: 3,5,7,9,11,12,13,15; G06K: 9,15; H04L: 9.

<sup>20</sup> Graham and Mowery, p. 232. The firms are Microsoft, Adobe, Novell, Autodesk, Intuit, and Symantec.

itself.<sup>21</sup> Patents containing “antigen”, “antigenic”, or “chromatography” in the description/specification are also excluded.

Our third algorithm for defining software patents (which we label HM) is similar to that used by Graham and Mowery. We identified all the U.S. patent class-subclass combinations in which fifteen software firms (Microsoft, Adobe, Novell, Autodesk, Symantec, Macromedia, Borland, Wall Data, Phoenix, Informix, Starfish, Oracle, Veritas, RSA Security, and Peoplesoft) patented and then categorized patents falling in these class-subclass combinations as “software.”<sup>22</sup> We refer to this definition of software patents as the Hall-MacGarvie definition.

As a check of the accuracy of these various definitions, we used a sample of manually identified software and business method patents compiled by John Allison.<sup>23</sup> Using a sample of 1000 patents issued between 1996 and 1998, Allison and Lemley (2000) identified 100 software patents by reading the claims and descriptions. Allison and Tiller (2003) augmented this sample with 230 internet business method patents, most of which are arguably members of the class of software patents. Table 1 shows the results of comparing this list with the patents selected by the three definitions, Bessen-Hunt, Graham-Mowery, and Hall-MacGarvie. From this procedure we can learn whether our samples include patents identified by Allison, although we are unable to ascertain how

---

<sup>21</sup> Bessen and Hunt, p. 4.

<sup>22</sup> One complication in using the U.S. patent classification system is that patents are continually reclassified as new classes are opened up. Software has been particularly subject to this reclassification, because there were no specific software classes until the “700” classes were created after the issuance of the USPTO guidelines in 1996. We are using the 2002 classification of all the patents in our dataset, which is late enough so that software should be correctly classified.

<sup>23</sup> We are grateful to James Bessen for making these patents available to us in machine-readable form, and for updating the data to 2002.

many patents in our sample would *not* have been labeled software by his procedure. That is, we can measure Type I error (missing a software patent we should have identified) but not Type II error (calling a patent a software patent when it is not).

The results are fairly clear: the keyword method of Bessen and Hunt is much more accurate in the sense of avoiding Type I error than either the International Patent Classification (IPC)-based classification or the US patent class-based classification, either of which identify only about half of these patents. Interestingly, Graham-Mowery is better at identifying software patents, whereas Hall-MacGarvie is much better at internet business method patents. This is probably because the US patent class system explicitly admits the existence of such patents, while the IPC does not, so they may end up classified in a wide range of IPC classes.

Layne-Farrar (2005) describes an attempt to determine the Type II error in these different definitions of software patents. She reports that when software experts read a random sample of patents from each dataset to see if they were truly for software, they classified about 5 per cent of the patents held by packaged software firms in the Graham-Mowery and Allison-Tiller sets as not software.<sup>24</sup> Layne-Farrar finds that a larger proportion of the Bessen-Hunt sample were classified as not relating to software, which would suggest that using only this definition is problematic. However, since the latter sample (BH) includes patents held by firms across all industries, the results of the comparison between the three samples are somewhat difficult to interpret.

---

<sup>24</sup> Layne-Farrar did not have access to the sample selected by our US patent class rule, because her paper was written in 2005.

However, because the GM definition does well on Type II error, and because GM and HM seem to select different groups of patents, whereas BH seems to be more comprehensive (better at Type I error), our preferred definition of software patents combines these definitions in the following way: we take the union of the GM and HM samples, and then intersect this with the BH sample. On the assumption that the HM US class/subclass approach does something similar to the GM IPC class/subclass approach, this intersects a sample that should have approximately 5% type II error (that is, it rarely misidentifies patents as software that are not software) with an inclusive sample that covers almost all software patents plus many other computing patents. The result is shown in the final row of Table 1: our new definition captures over 80 per cent of the patents identified by Allison as software and internet patents, but not at the cost of being too inclusive. We therefore chose to use this combined definition as our preferred sample of software patents, and to present results using the other definitions in an appendix.

Figures 1 and 2 show the trends in software patent growth using the four definitions described above plus an aggregate software patenting series supplied by Aharonian (2001) in his email newsletter.<sup>25</sup> Figure 1 shows the absolute numbers and Figure 2 shows the share of software patents among all granted patents. The Aharonian definition appears to track that from the Bessen-Hunt keyword definition more closely than the others, whereas our combined definition is conservative and lies somewhere between the two patent class definitions (those using IPC and US patent classes). Using any of the definitions, there is substantial growth since 1976 that accelerates in around

---

<sup>25</sup> Greg Aharonian is a patent agent specializing in Information Technology sector patenting who has been tracking software patents using his own subjective evaluation of subject matter for some time. We include his figures to give some idea of the view of an observer familiar with the sector and its patenting.

1995-1996. The apparent jump from 1997 to 1998 using any of the definitions results from a jump in applications three years earlier, that is presumably due to the series of court decisions that are described in the next section of the paper.

#### **4. *Event studies***

As discussed earlier, in 1994 and 1995, the Court of Appeals for the Federal Circuit (CAFC) handed down a number of decisions that affected the scope of software patents. These decisions include *In re Alappat*, 33 F.3d 1526 (1994, en banc); *In re Warmerdam*, 33 F.3d 1354 (1994); *In re Lowry*, 32 F.3d 1579 (1994), *In re Trovato*, 60 F.3d 805 (1995, en banc), and *In re Beauregard*, 53 F.3d 1583 (1995). In response to the confusion created by the court's determination that much of software was patentable, the USPTO proposed new guidelines for software patentability on May 12, 1995, and published these guidelines on March 29, 1996. In this section of the paper, we describe the decisions and then present the results of several studies of the effects of the decisions on the value of firms in the software industry or holding software patents. Table 3 presents a timeline for the decisions we consider, with the abnormal returns experienced by our firms around the date of each decision (these results are discussed in section 4.1)

##### ***In re Alappat***

In 1988, Alappat filed an application for a patent on a means of smoothing the appearance of a waveform displayed by a digital oscilloscope. The PTO rejected the application's claims as non-patentable subject matter. Alappat appealed the decision to the Board of Patent Appeals, which reversed the PTO's decision on June 26, 1991. The patent examiner then requested a reconsideration of the decision by an expanded panel of

the Board of Appeals, which on April 22, 1992 reversed the decision of the original panel. On July 29, 1994, the CAFC handed down a decision stating that the invention was not an “abstract idea”, but rather, “a specific machine to produce a useful, concrete, and tangible result.”<sup>26</sup> This decision was interpreted as a clear expansion of the patentability of software.<sup>27</sup>

### ***In re Warmerdam***

The Alappat decision was followed on August 11, 1994 by *In re Warmerdam*, partly affirming and partly reversing the decisions of the Board of Patent Appeals, which had upheld the PTO’s rejection of an invention as non-patentable subject matter. The CAFC ruled that the invention, a mechanism for generating data structures for collision avoidance systems, was “nothing more than the manipulation of basic mathematical constructs, the paradigmatic ‘abstract idea’”, and sustained the rejection of five of the patent’s six claims.<sup>28</sup> However, the court affirmed one claim, implying that a machine containing in memory the data structure created by the mechanism would indeed be patentable (Huttner and Strobert 1995).

### ***In re Lowry***

*Warmerdam* was followed on August 26, 1994 by *In re Lowry*, which reversed the PTO’s rejection of an application on the grounds that a data structure held in memory

---

<sup>26</sup> *In re Alappat* 33 F.3d 1526, 31 USPQ2d 1545

<sup>27</sup> The decision was first mentioned in the Wall St. Journal on August 8, 1994. We also tried computing abnormal returns around this date, but the CARs were not statistically significant.

<sup>28</sup> *In re Warmerdam*, 3 F.3d 1354, 31 USPQ2d 1754

was unpatentable under the “printed matter” doctrine. The “printed matter” doctrine provides that “an article of manufacture which consists of printed matter on a substrate cannot be statutory if it differs from other substrates only by the informational content of the printed matter except where there is some functional interaction between the printed matter and the substrate.”<sup>29</sup> In this instance, the PTO had held that Lowry had not shown that the data structures were functionally related to the memory in which they were stored, but the CAFC ruled that the printed matter doctrine did not hold when the information in question was processed by a machine rather than by the human mind.

### ***In re Trovato and In re Beauregard***

In *In re Trovato* (December 19, 1994), the CAFC upheld the PTO’s rejection of a software patent for containing non-statutory subject matter. The patent covered a new technique for calculating the most efficient path of an object between states in physical space. The court ruled that “[p]utting Trovato’s claims in their most favorable light, the most they provide is a systemic way in which to compute a number representing the shortest path. A new way to calculate a number cannot be recognized as statutory subject matter.”<sup>30</sup> On May 12, 1995, the CAFC ruled in *In re Beauregard* that the PTO’s rejection of a patent’s claims under the printed matter doctrine was incorrect given the precedent set in *In re Lowry*. The patent office admitted its mistake in the case.<sup>31</sup>

### ***New UPSTO Guidelines***

---

<sup>29</sup> USPTO 1995. See also *In re Gulack*, 703 F.2d 1381, 217 USPQ 401.

<sup>30</sup> *In re Trovato*, 60 F.3d 805.

<sup>31</sup> *The Computer Lawyer*, Vol. 12, No. 5; Pg. 30, May 1995

In response to the above decisions, the USPTO proposed new guidelines for software patentability on May 12, 1995, and published these guidelines on March 29, 1996. The guidelines expanded on *Warmerdam*'s finding with respect to data held in the memory of a machine, stating that the PTO would begin to allow claims related to software embedded in physical media. Claims should be considered processes unless they relate to some type of machine or physical apparatus. The guidelines clarified the definition of the following criteria to be used by examiners to distinguish between

- “a) a computer or other programmable apparatus controlled by software as a statutory ‘machine’;
- b) a computer-readable memory used to direct a computer, such as a memory device, a compact disc or a floppy disk as a statutory ‘article of manufacture’; and
- c) a series of steps to be performed on or with the aid of a computer as a statutory ‘process.’”<sup>32</sup>

The guidelines stated that inventions that were to be considered non-statutory included data compilations or structures independent of physical elements, encoded information representing creative or artistic expression, and processes that do “nothing more than manipulate abstract ideas or concepts.”<sup>33</sup>

#### **4.1 Null hypotheses on the effects of the patentability decisions**

As stated previously, the expansion of the patentability of software may have had both positive and negative effects. Firms already holding patents at the time of the

---

<sup>32</sup> USPTO guidelines quoted in Huttner and Strobert (1995).

<sup>33</sup> *op. cit.*

decision may be positively affected due to an increased ability to exclude competitors from a market or collect licensing revenues. We will thus compare the effects of patentability on patent holders and non-patent holders as of the time of the event, with the expectation that the market reaction should be larger for patent holders. Note that another set of firms that may benefit from software patents are start-ups for whom patents may help secure financing. Because we are focusing on the market value of publicly traded firms, we are unable to study this effect.<sup>34</sup>

We also consider the potential negative effects of the decisions. For all firms, the decisions may have increased costs by creating the need to engage in licensing negotiations and by increasing the potential for hold-up. We hypothesize that the latter effect is especially relevant for firms in “downstream” market segments, i.e., firms producing software that must interact with or operate on top of other software or platforms. We thus test whether the market reacted negatively for firms specializing in applications software and software-related services relative to firms producing for “upstream” segments (middleware, systems software, and hardware).<sup>35</sup>

## 4.2 Results of the event studies

Table 3 contains the results of the event studies that identify the market’s response to each of the above-mentioned decisions, for the entire sample of firms and for software firms only. We calculate the cumulative abnormal returns (CARs) associated with the events, and their standard errors, using the standard market model approach as

---

<sup>34</sup> Cockburn and MacGarvie (2006) study the effects of patent “thickets” on entry in narrowly-defined software product areas.

<sup>35</sup> See the appendix for our classification of firms into these segments.

described in Appendix B. To obtain the market model for each stock, we regress the stock returns for firms in our sample on the market return (using the CRSP value-weighted market index) using data from the period (-90, -30) relative to each event.<sup>36</sup> We then estimate the excess returns for the period (-1,+3) around the event by summing the forecasted residuals from the market model and computing their standard errors, taking account of any correlation that might be present.

For each event we show the average CAR for all firms in our sample and for four subgroups of firms: 1) firms that only produce software (firms whose primary SIC code is 73XX and who do not also produce hardware), 2) firms that specialize in applications software or software-related services, as defined by the Corptech directory or by our investigation using web searches,<sup>37</sup> 3) firms that had not applied for patents in the year prior to the event, and 4) firms that had not applied for software patents in the year prior to the event. We also performed rank tests of the difference in the CAR distribution between the each group and other firms using the methodology described in Appendix B.<sup>38</sup> We group the CARs in this way because we expect the effect of the decisions to differ depending on whether firms can expect to take advantage of the increased strength of software patents. We expect that firms holding software patents at the time of

---

<sup>36</sup> Our study includes some events that are less than one month apart. We choose the 30 day lag for the window to minimize any contamination arising from overlapping events.

<sup>37</sup> As of the date of our data (2002), Compustat classified IBM into software services. Because of the importance of this firm in software patenting in general and because for most of our time period IBM was primarily a hardware firm, we have reclassified it into computing machinery. However, throughout this paper, we found that estimates with and without including observations on IBM were substantially the same.

<sup>38</sup> We found no significant differences for software firms versus hardware firms or application/services firms versus the others, so the results of these tests are not shown in the table.

decisions that broaden patentable subject matter to include some types of software will be positively affected relative to firms without patents. We also expect that firms in “downstream” sectors – like applications software and software-related services – will be negatively affected relative to upstream firms in hardware and middleware or systems software, because they may be forced to license in patented technology in order to ensure that their products will function on top of middleware or the operating systems.

The first event we consider is the Supreme Court’s decision in *Diamond v. Diehr*. Surprisingly, this decision has a positive overall return of almost one per cent in our sample, and is also positive for all of the subsamples, although the number of software firms in the sample is too small in 1981 to yield significant results. One possible interpretation is that this development, which was a rather narrow extension of patentability to software embodied in hardware, was viewed as positive for the majority of firms then in the sample, almost 90 per cent of whom were not pure software firms.

The next two events were the assertion of the Compton patent and its subsequent revocation.<sup>39</sup> Clearly the assertion was viewed as highly negative news for all ICT firms, and especially for software firms, although the difference is not significant. There was little significant reaction to the revocation of the patent, suggesting that the event was anticipated. What reaction there was is slightly negative.

The *In re Alappat* decision is widely viewed as the groundbreaking one on software patentability and this is reflected in the abnormal returns at the time of the

---

<sup>39</sup> Unfortunately, it is not possible to measure the reaction of the patent owner value to this event. Compton Encyclopedia had been purchased by Encyclopedia Britannica, a privately-held firm, in 1961, so no market value data is available.

decision.<sup>40</sup> All firms experienced negative abnormal returns, statistically significant at the 5 per cent level using the rank test. The negative returns were primarily due to those firms that had not applied for software patents by 1993 (the year preceding). However, the magnitude of the effects is small, less than one half of a percent.

After *Alappat*, the next three decisions, *In re Warmerdam*, *In re Lowry*, and *In re Trovato* had little impact. The first two followed closely on *Alappat* and all three took the form of clarifying, partially affirming, and partially reversing that decision. *In re Warmerdam* was viewed as having ambiguous implications, and was seen to counteract previous extensions of software patentability while simultaneously affirming the patentability of a machine storing “data structures” in memory.<sup>41</sup> While this latter aspect of the decision is today viewed (with the benefit of hindsight) as a step towards patentability, contemporary press reports suggest it may have been primarily interpreted as a reversal. The headline in *The Computer Lawyer* in September 1994 was “Federal Circuit Holds Collision Avoidance System Unpatentable.”<sup>42</sup> *In re Lowry*, a clear expansion of patentability but one that essentially confirmed *Warmerdam* and *In re Trovato* had little impact.

*In re Beauregard* was a test case submitted by IBM in the wake of the *Lowry* decision, essentially looking for confirmation from the courts that the “printed matter”

---

<sup>40</sup>In reference to *In re Alappat*, Evans and Layne-Farrar (2004) state that “This ruling cemented the statutory standing of software patents.” (p. 6). Mann (2004) says *Alappat* “cleared the way for software patents” Cohen and Lemley (2001) state that “In 1994, the en banc Federal Circuit decided *In re Alappat*, opening a new era in software patent protection”(p. 10).

<sup>41</sup> Hollander (2003), Paragraphs 18-22.

<sup>42</sup> See also Stobbs (1995), who contrasts *Warmerdam* with *Lowry* by writing “*Warmerdam*'s data structure is not patentable subject matter... *Lowry*'s data structure is patentable subject matter.”

doctrine did not apply to software on physical media. In fact, the PTO conceded its mistake in rejecting the patent application in this case and the CAFC never rendered a decision. The result was PTO acknowledgement that a computer-readable medium qualifies as a statutory article of manufacture to the extent program code stored on the medium and executable by a processor functionally defines the medium. The overall market reaction to this acknowledgement was significantly positive, but that for firms without patents and without software patents was significantly lower than for firms with patents.

On the heels of the *Beauregard* decision, the USPTO announced a set of proposed new guidelines for software patents in May 1995. The announcement was viewed by the stock market as significantly negative for all firms in ICT, and especially so for software firms. However, the final issuance of the guidelines somewhat reversed this reaction, suggesting either the resolution of uncertainty or confirmation that the expansion of patentability would not be as broad as feared.

## ***5. Market value and software patenting***

In this section of the paper we employ the Hall, Jaffe, Trajtenberg (henceforth, HJT 2005) methodology to estimate the contribution of software patents to Tobin's Q. We compare the value of these patents to patents in general, both before and after the changes in software patentability rules in 1995. In order to adjust for differences in the quality of the patents, we perform the same exercise with cite-weighted patents, and with cite-weighted patents excluding self-cites.

The HJT 2005 paper specifies a firm-level market-value equation that is linear and additively separable (following Griliches (1981)). We follow the HJT paper closely

in the ensuing description of the model. The market value of firm  $i$  in year  $t$  is modeled as:

$$V_{it} = q_t (A_{it} + \gamma K_{it})^{\sigma_t} \quad (1)$$

where  $A_{it}$  stands for physical assets, and  $K_{it}$  the firm's knowledge assets. Taking logarithms yields the following equation:

$$\log V_{it} = \log q_t + \sigma_t \log A_{it} + \sigma_t \log[1 + \gamma(K_{it} / A_{it})] \quad (2)$$

Assuming constant returns to scale (that is,  $\sigma_t = 1$ ) and moving  $A_{it}$  to the left hand side, we can express the model with Tobin's  $q$  as the dependent variable:

$$\log Q_{it} \equiv \log(V_{it} / A_{it}) = \log q_t + \log[1 + \gamma(K_{it} / A_{it})] \quad (3)$$

We specify the knowledge capital  $K_{it}$  as a function of the stock R&D spending and the stock of patents, both unweighted and weighted by citations. Given our specific interest in this paper, we further break  $K$  down according to whether the patents are in software-related fields. This results in the following estimating equation:

$$\log Q_{it} = \log q_t + \log[1 + \gamma_1(RD_{it} / A_{it}) + \gamma_2(P_{it} / RD_{it}) + \gamma_3(SP_{it} / RD_{it})] + \varepsilon_{it} \quad (4)$$

in which  $RD_{it}$  is firm  $i$ 's R&D capital stock in year  $t$ ,  $P_{it}$  is a measure of patent stock in year  $t$ , and  $SP_{it}$  is a measure of the software patent stock in year  $t$ . The coefficient  $\gamma_2$  measures the impact of patents above and beyond that of the R&D that produces them and  $\gamma_3$  measures the premium or discount associated with software patents when compared with ordinary patents.  $P$  and  $SP$  are based on patents dated by year of application, citation-weighted patents dated by year of application, or citations excluding self-citations to patents dated by year of application.

A full set of two-digit industry and year dummies for the sample period (1980-1999) are also included, and in some cases industry-year interactions.<sup>43</sup> The R&D and patent stock measures are constructed using the usual declining balance method with a depreciation rate of 15 per cent:

$$K_{it} = (1 - \delta)K_{i,t-1} + R_{it} \quad (5)$$

where  $R_{it}$  is R&D spending, granted patents applied for in year  $t$ , or granted software patents applied for in year  $t$ . Patents are either simple counts, counts weighted by citations they subsequently received, or counts weighted by citations excluding self-citations.<sup>44</sup>

Interpretation of the coefficients in an equation like equation (4) can be difficult, since the variables are in a variety of units (dollars per dollar, counts per dollar, etc.). To enhance comparability, we computed the elasticity of Tobin's Q with respect to each of our variables using the following equation:

$$\frac{\partial \log Q_{it}}{\partial \log X_{it}^j} = \frac{\gamma_j X_{it}^j}{1 + \gamma_1(RD_{it} / A_{it}) + \gamma_2(P_{it} / RD_{it}) + \gamma_3(SP_{it} / RD_{it})} \quad (6)$$

where  $j = 1, 2, 3$  and  $X_{it}^j$  is the corresponding right hand side variable. We then averaged these elasticities across the observations and reported them in the bottom panel of the table.

---

<sup>43</sup> We include these interactions to deal with the large increase in valuation for software firms at the end of our sample period, which was presumably due to the dotcom "bubble".

<sup>44</sup> Our patent data end in 2002 and our regressions in 1999. This means that we have three years to observe the issuance of the youngest patents in the sample, which is enough time to observe most of them. However, it is not sufficient time to observe forward citations for the most recent patents, so we have adjusted the number of citations received by each patent by the ratio of the total number expected to be received by a patent in that technology class to the average number such patents have received by the time of the relevant citation lag, using a methodology described in Hall, Jaffe, and Trajtenberg 2001.

We estimate equation (4) using nonlinear least squares and report the results in Tables 4. Eicker-White standard errors are computed to ensure robustness to heteroskedasticity. We included a complete set of year dummies for each of the 5 two-digit industries in our sample (machinery, electrical machinery, instruments, telecommunications, and software). Although we were concerned that general trends in the valuation of software firms, especially during the late 1990s, might be contaminating our results, there was in fact almost no difference between the results in Table 4 and those using a single set of year dummies for all of the sectors. There are three sets of columns in the table, one set corresponding to the whole period, one set for the pre-guidelines period (1980-1994) and one for the post-1994 period, in order to focus on the question of whether the value of software patents increased following the *Beauregard* decision and the Commissioner of Patents' issuance of new guidelines on software patentability in 1995.

The overall results are similar to those in HJT 2005, with R&D having a strong relationship with Tobin's Q, and either patents or citation-weighted patents having a somewhat weaker but still significant relationship in the presence of R&D. Software patents are valued the same by the market as ordinary patents before the 1994/1995 changes in software patenting policy, but they are valued twice as much as ordinary patents following the changes. The average elasticities suggest that doubling the total patent yield per R&D dollar increases market value by approximately one to two per cent, increasing slightly after 1994, whereas if the patents in question were software patents, a doubling in the number of patents would be associated with a similar increase in value before 1995 and a four per cent increase after (significant at the 1% level). The

results for citation-weighted patents are similar, in that the total patent elasticity is stable over time and the software patent elasticity increases after 1995, but the boost from patenting in software is no longer as apparent when the patents are weighted by citations, especially when self-citations are excluded. This suggests that, relative to other patents, mere ownership of software patents may be more important than their actual invention value. However we are reluctant to draw strong conclusions from this finding because there has been limited time to observe the citations during the later period.

In the top section of Table 5 we split the sample into software and hardware (machinery, electrical machinery, instruments, and telecommunications) firms and show the results for the same estimation as in Table 4, but only for unweighted patent stocks and for cite-weighted patents excluding self-citations, and only for the pre-1995 period and the 1995-1999 period separately. The bottom of Table 5 presents results from the same regressions estimated separately for middleware/systems software and applications software/software-related services. The elasticities from these regressions are found in Table 6.

These tables reveal some interesting detail obscured by the results based on aggregated data in Table 4. First, for hardware firms, the value of software patents is insignificantly different from that of ordinary patents prior to 1995. Second, the patents held by software firms prior to 1995 are actually *negatively* valued by the market although their R&D clearly is valued.<sup>45</sup> However, *software* patents held by software firms have a positive elasticity, counteracting (but not completely) the negative overall patent

---

<sup>45</sup> Note that this is not simply because they do not have software patents. Prior to 1995, 13 per cent of the software firms hold software patents by our definition and 21 per cent have patents. After 1994, the numbers are 22 per cent and 30 per cent respectively.

elasticity prior to 1995. The overall negative valuation of patents held by software firms prior to 1995 may reflect the market's perception that expenditures of resources to obtain patents were not fruitful in an environment of weak patent rights over software. Because most software firms do not possess any patents, it may also reflect lower valuations for firms that are in submarkets where patents are important. In contrast, after the court decisions of 1994, software patents held by either hardware or software firms become more valuable, with a premium on the patent elasticity of 0.02 to 0.03. Combining this result with the elasticity for patenting in general, if a hardware firm doubles its software patent yield per R&D dollar, this is associated with a 4.7 per cent increase in market value; for a software firm, it is associated with a three per cent increase in market value.

Third, citation weighted patents are valued very differently in the two sectors. For hardware firms, there is a no premium for citations to software patents in either period. For software firms, the negative effect of owning patents disappears when we look at citation-weighted patents with self-citations excluded. Ordinary citation-weighted patents are not valued at all in either period, whereas citation-weighted software patents have an elasticity of about 0.02 after 1995. One simple interpretation of these results is that patent rights on more valuable inventions are more valuable to the firm when they are rights to the main technology of their industry, so cite-weighted software patents are especially valuable to software firms, whereas ordinary patents are not particularly valuable for these firms. This may suggest that holding patents in software are valuable only for strategic reasons (portfolio building) in hardware, whereas in software they serve more a traditional exclusion function, protecting individual inventions.

In the lower portion of Table 5 and the second panel of Table 6, we dig deeper by separating the software category into middleware/systems software (MS) and applications/software-related services (AS). We find considerable differences between these two groups of firms. First, the middleware/systems group generally holds more patents, even when we measure this relative to the size of their R&D program. Second, the negative value of holding patents before 1995 was due only to the AS firms; for the MS firms, the elasticity of value with respect to patent yield is even higher than for hardware. Post-1994, this value elasticity is associated only with software patents for these firm, perhaps reflecting the change in patent classifications that accompanied the introduction of “true” software patents. For these firms, similar results obtain when the patents are weighted by citations. The conclusion is that at least for the MS firms engaged in patenting activity, patent ownership enhances value at a slightly higher level than that reported by HJT (2005) for the entire manufacturing sample.<sup>46</sup>

However, for applications and software services firms, neither citation-weighted nor unweighted software patents are very related to market value, before 1995 or after. The only exception is the net negative relationship of software patents to market value prior to 1995. These findings are consistent with the event study results described in the previous section of the paper, in which it was found that applications/services firms were more negatively affected by the expansion of software patentability than were the more “upstream” middleware/systems software firms. That is, whatever benefits there may be to holding software patents are limited to middleware/systems software and hardware firms. These findings indicate that assessments of the implications of software

---

<sup>46</sup> The similar elasticity reported by HJT is approximately 0.016.

patentability for the ICT industry should consider differences across sectors in the value of software patents to individual firms.

## 6. *Conclusions*

We obtain two sets of conclusions based on our two approaches to measuring the private value of software patents. First, we conclude that, as measured by the stock market's reaction to legal decisions expanding the patentability of software, there is some evidence that the expansion of software patentability was initially negative for firms in the software industry. There is limited evidence of relatively negative effects on some types of software producers – applications producers and providers of software-related services, and firms with no patenting experience at the time of the decisions. The most important events from the perspective of news to the market are the original assertion of the Compton multimedia patent, which affected all types of firms negatively on average, and the announcement of proposed new guidelines for software patenting.

Our second approach examines the relationship between Tobin's Q and firms' patent and citation stocks. Following the expansion of software patentability in the mid-1990s, software patents seem to be more highly valued by the market than ordinary patents, even in the presence of controls for the overall increase in the market valuation of software firms. However, we uncover important differences within the software industry in the value of software patents, with applications software producers and providers of software-related services appearing to derive no benefit from software or other patents, either before the change in patentability or after. Combining these two sets of findings, we conclude that the market evaluated software patents as a negative development *ex ante* and expected that the expansion of software patentability would

affect firms in downstream sectors and firms without patents relatively more negatively. *Ex post*, of course, a greater number of firms in all ICT sectors invested in these patents. In addition, firms in the ICT sector that hold software patents are found to be valued at a premium relative to firms without software patents. However, there is variation within software in the ability of firms to reap benefits from holding software patents, with the benefits primarily accruing to the more “upstream” middleware and systems-software producers.

This story paints a picture of firms adjusting to a new environment and set of rules, which although they were initially perceived as negative for the software sector, have provided some benefits, at least for firms large enough to be publicly traded on the market. Nevertheless we should also emphasize that the majority of the software patents acquired during the past twenty years have been acquired by non-software firms in the ICT sector (in our sample the figure is almost 90 per cent), which suggests that the growth in these patents is driven to a great extent by the needs of hardware firms for large patent portfolios rather than by the needs of software firms to protect their inventions.

## 7. References

Allison, J. R. and M. A. Lemley. 2000. "Who's Patenting What? An Empirical Exploration of Patent Prosecution," *Vanderbilt Law Review* 58: 2099-2148.

Allison, J. R. and E. H. Tiller, 2003, "Internet Business Method Patents," in W. M. Cohen and S. A. Merrill (eds.) *Patents in the Knowledge-Based Economy*, National Research Council, Washington: National Academies Press, pp. 259-84.

Aharonian, G. (2005). "Critiques of Software Patent Examination," on <http://www.bustpatents.com/>

Arundel, A., J. Cobbenhagen, and N. Schall. (2002). *The Acquisition and Protection of Competencies by Enterprises*. Report for EIMS project 98/180, DG Enterprise, European Union, Luxembourg.

Bakels, R. (2005). "Current IPR Policies and Law Followed by Major Organisations in the Domain," in Ghosh et al, Report to the European Commission on The Effects of Allowing Patent Claims for Computer-Implemented Inventions, August.

Bakels, R. and P. B. Hugenholtz (2002). "The Patentability of Computer Programs," A Report to the European Parliament, Amsterdam: IViR.

Barton, J. H. (2002). "Non-Obviousness," *IDEA Journal of Law and Technology* 42.

\_\_\_\_\_. (2000). "Reforming the Patent System," *Science* 287: 1933-1934.

Bessen, J. and R. M. Hunt (2003). "An Empirical Look at Software Patents", Working Paper. <http://www.researchoninnovation.org/swpat.pdf>

Blind, K., J. Edler, and R. Nack (2001). "Micro- and Macro-economic Implications of the Patentability of Software Innovations. Intellectual Property Rights in Information Technologies between Innovation and Competition," Karlsruhe/Munich: Fraunhofer Institute and Max Planck Institute.

Cockburn, I. and M. MacGarvie (2006). "Entry, Exit and Patenting in the Software Industry," Boston University: Working Paper.

Cohen, J. E. and M. A. Lemley (2001). "Patent Scope and Innovation in the Software Industry," *California Law Review* 89(1): 1-57.

Cohen, W., A. Goto, A. Nagata, R. Nelson, and J. Walsh. (2002). "R&D Spillovers, Patents and the Incentives to Innovate in Japan and the United States." *Research Policy* 31:1349-1367.

Davidson, R., and J. G. MacKinnon. (1993). *Estimation and Inference in Econometrics*, Oxford: Oxford University Press.

- Dreyfuss, R. C. (2001). "Examining State Street Bank: Developments in Business Method Patenting," *Computer und Recht International* 2001(1): 1-9.
- \_\_\_\_\_. (2000). "State Street or Easy Street: Is Patenting Business Methods Good for Business?" In *Intellectual Property Law*, Volume 6, Chapter 14. London: Juris Publishing Company, Ltd.
- Dvorak, John. (2005). "Software Patents: Microsoft's Fatal Error," *PC magazine*, April 18, 2005.
- European Commission. (2002). "Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-implemented Inventions," Brussels, Belgium.
- Evans, D. and A. Layne-Farrar (2004) "Software Patents and Open Source: The Battle Over Intellectual Property Rights", *Virginia Journal of Law and Technology*, Vol. 9, No. 10, Summer 2004.
- Gao, L. (2005) "Essays On Investments And Financing Decisions In Complementary Networks Systems", unpublished doctoral dissertation, Boston University.
- Graham, S. J. H. and D. C. Mowery (2003). "Intellectual Property Protection in the U. S. Software Industry," in W. M. Cohen and S. A. Merrill (eds.), *Patents in the Knowledge-Based Economy*, Washington, DC: National Academies Press.
- Griliches, Z. (1981). "Market Value, R&D, and Patents," *Economic Letters* 7: 183-187.
- Hall, B. H. (2005). "Exploring the Patent Explosion," *Journal of Technology Transfer* 30: 35-48.
- Hall, B. H. (2003). "Business Method Patents, Innovation, and Policy," NBER Working Paper No. W 9717.
- Hall, B. H. and Clint Cummins. (2005). *TSP 5.0 Reference Manual*, Palo Alto, CA: TSP International.
- Hall, B. H., S. J. H. Graham, S. J. H., D. Harhoff, and D. C. Mowery. (2003). "Prospects for Improving U.S. Patent Quality via Post-grant Opposition," MIT Press: *Innovation Policy and the Economy* 4: 115-143.
- Hall, B. H., A. Jaffe, and M. Trajtenberg (2005), "Market Value and Patent Citations," *Rand Journal of Economics* 36: 16-38.
- Heller, M.A. and R.S. Eisenberg. (1998). "Can Patents Deter Innovation? The Anticommons in Biomedical Research," *Science* 280: 698-701.
- Hollander, Andrew J. (2003) "Patenting Computer Data Structures: The Ghost, the Machine, and the Federal Circuit", *Duke Law and Technology Review*, 0033.
- Hunt, R. M. (2001). "You Can Patent That? Are Patents on Computer Programs and Business Methods Good for the New Economy?" *Philadelphia Federal Reserve Bank Business Review* 2001(Q1): 5-15.

- Huttner, C. and A. Strobert (1995) "New Proposed Guidelines on Patentability Of Computer-Implemented Inventions", *New York Law Journal*, September 25, 1995.
- Kasdan, J. (1999). "Obviousness and New Technologies," New York, NY: Columbia University Center for Law and Economics Studies Working Paper No. 146.
- \_\_\_\_\_. (1994). "Fascinatin' Algorithm: Patent Protection for Computer Programs," Columbia University Center for Law and Economic Studies Working Paper No. 94.
- Lerner, J. (2001). "Where Does State Street Lead? A First Look at Finance Patents, 1971-2000," *Journal of Finance* 57: 901-930.
- Layne-Farrar, A. (2005). "Defining Software Patents: A Research Field Guide," LECG, Chicago: manuscript.
- Campbell, J. Y., A. W. Lo, and A. C. MacKinlay (1997), *The Econometrics of Financial Markets*, Princeton, NJ: Princeton University Press.
- Lunney, G. S., Jr. (2001). "e-Obviousness," *Michigan Telecommunications Technology Law Review* 7: 363-422.
- Mann, R. (2004) "The Myth of the Software Patent Thicket", working paper.
- Meurer, M. J. (2002). "Business Method Patents and Patent Floods," *Washington University Journal of Law and Policy* 8: 309.
- Murmann, J. P. (2003). *Knowledge and Competitive Advantage: The Co-evolution of Firms, Technology, and National Institutions*, Cambridge: Cambridge University Press.
- Raduchel, W. (2006). "The Economics of Software," in D. W. Jorgenson and C. W. Wessner (eds.), *Software, Growth, and the Future of the U.S. Economy, Report of a Workshop* Washington DC: National Academies Press.
- Salinger, M. (1992), "Standard Errors in Event Studies," *The Journal of Financial and Quantitative Analysis*, Vol. 27, No. 1 (March 1992), 39-53.
- Samuelson, P. (1984), "CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form," *Duke Law Journal*, pp. 663-702.
- Scotchmer, S. (1996). "Protecting Early Innovators: Should Second Generation Products be Patentable?," *Rand Journal of Economics* 27: 322-331.
- \_\_\_\_\_. (1991). "Standing on the Shoulders of Giants: Cumulative Research and the Patent Law," *Journal of Economic Perspectives* 5: 29-41.
- Spindler, G. (2003). "The European Legal Framework for Software Patents," paper presented at the EPIP conference on New Challenges to the Patent System, EPO, Munich, Germany, April 24/25, 2003.
- State Street Bank and Trust Co., Inc. v. Signature Financial Group, Inc., 149 F.3d 1368

(Fed. Cir. 1998).

Sterne, R. and L. Bugaisky (2004), “The Expansion of Statutory Subject Matter Under the 1952 Patent Act”, *Akron Law Review* 37: 216-229.

Stobbs, G. A. (1995) “Software Practice News: The Common Thread Unraveled”, *Intellectual Property Today*, Pg. 6,

USPTO (1995), “Proposed Guidelines for Computer-Implemented Inventions,” <http://www.uspto.gov>, 1 June 1995.

## Appendix A

### Additional Tables and Figures

#### *Differences between software patent definitions*

We consider several different definitions of what constitutes a software patent (see p. 12-17 for a discussion of these definitions). Figures 1 and 2 plot the total number of software patents granted and the number of software patents as a percentage of all patents granted, for each of the alternative definitions.

Table A2 presents results from the market value model for the Bessen-Hunt, Graham-Mowery, Hall-MacGarvie, and “combined” (GM or HM intersected with BH) definitions. Each column presents estimates for one of the definitions, and the differences between them highlight the importance of the choice of definition.. For software firms, looking at the entire sample period (1990-1999), software patents and forward citations are positively and significantly associated with Tobin’s Q across all the definitions. The magnitudes of the elasticities are also comparable across definitions.

However, there are differences between the definitions when we look at non-software firms and when we consider the change in valuation over time. Here, the BH definition diverges from the others, showing a positive and significant valuation of citation-weighted software patents for non-software firms, even in the pre-1994 period, as well as a *decrease* over time in the elasticity of market value with respect to software patents held by software firms. This may reflect the fact that the BH set of patents appears to contain a considerably greater number of hardware patents, at least according to the sample studied by Layne-Farrar.

**Table A1**

**Category Definitions Based on CorpTech product codes (SOF codes)**

Based on Gao (2005)

**SYSTEMS SOFTWARE:**

- A. Utility systems software 1

**MIDDLEWARE SOFTWARE:**

- A. Utility systems software 2
- B. Communications systems software
- C. Internet tools
- D. Software development systems
- E. Artificial intelligence software
- F. Database/file management software
- D. Software development systems

**APPLICATIONS SOFTWARE:**

- A. Accounting, banking, financial, government, military, legal, real estate, insurance, health services, public utilities, transportation, sales/marketing and distribution software
- B. Technical/scientific software
- C. Construction, facilities and communications management software
- D. Manufacturing software systems
- E. Media and communications software
- F. Office automation software
- G. Educational and training software
- H. Other applications software, not elsewhere specified

**SERVICES:**

- A. Software related services

**Table A2**  
**Elasticities from market value equations, by software/non-software and time period**

	Software firms				Hardware firms			
	Bessen-Hunt	Graham-Mowery	Hall-MacGarvie	Combined	Bessen-Hunt	Graham-Mowery	Hall-MacGarvie	Combined
<b>1980-1999 (2173 software, 8717 non-software)</b>								
R&D/Assets	<b>0.331</b>	<b>0.322</b>	<b>0.317</b>	<b>0.321</b>	<b>0.287</b>	<b>0.289</b>	<b>0.289</b>	<b>0.289</b>
Patents/R&D	0.000	0.000	0.000	0.000	<b>0.009</b>	<b>0.016</b>	<b>0.017</b>	<b>0.017</b>
SW Pats/R&D	0.007	<b>0.017</b>	<b>0.022</b>	<b>0.021</b>	<b>0.019</b>	<b>0.004</b>	0.002	<b>0.004</b>
R&D/Assets	<b>0.317</b>	<b>0.316</b>	<b>0.314</b>	<b>0.316</b>	<b>0.286</b>	<b>0.287</b>	<b>0.285</b>	<b>0.287</b>
Citations/Patents	0.000	0.000	0.001	0.000	<b>0.022</b>	<b>0.034</b>	<b>0.035</b>	<b>0.034</b>
SW Cites/Patents	<b>0.022</b>	<b>0.022</b>	<b>0.021</b>	<b>0.023</b>	<b>0.016</b>	0.003	<b>0.004</b>	<b>0.003</b>
R&D/Assets	<b>0.318</b>	<b>0.316</b>	<b>0.323</b>	<b>0.323</b>	<b>0.286</b>	<b>0.287</b>	<b>0.287</b>	<b>0.287</b>
Citations/Patents (excluding self-cites)	0.000	0.000	0.007	0.008	<b>0.022</b>	<b>0.034</b>	<b>0.036</b>	<b>0.032</b>
SW cites/Patents (excluding self-cites)	<b>0.022</b>	<b>0.022</b>	<b>0.011</b>	<b>0.006</b>	<b>0.016</b>	0.003	<b>0.002</b>	<b>0.005</b>
<b>1980-1994 (924 software, 5989 non-software)</b>								
R&D/Assets	<b>0.365</b>	<b>0.348</b>	<b>0.335</b>	<b>0.350</b>	<b>0.260</b>	<b>0.265</b>	<b>0.266</b>	<b>0.265</b>
Patents/R&D	<b>-0.023</b>	<b>-0.022</b>	<b>-0.019</b>	<b>-0.020</b>	<b>0.010</b>	<b>0.016</b>	<b>0.017</b>	<b>0.017</b>
SW Pats/R&D	0.008	0.015	<b>0.021</b>	<b>0.015</b>	<b>0.017</b>	0.002	-0.001	0.001
R&D/Assets	<b>0.356</b>	<b>0.352</b>	<b>0.338</b>	<b>0.351</b>	<b>0.260</b>	<b>0.262</b>	<b>0.262</b>	<b>0.262</b>
Citations/Patents	<b>-0.017</b>	-0.017	-0.008	<b>-0.015</b>	<b>0.022</b>	<b>0.034</b>	<b>0.035</b>	<b>0.035</b>
SW Cites/Patents	<b>0.021</b>	<b>0.023</b>	<b>0.019</b>	<b>0.021</b>	<b>0.014</b>	0.002	0.001	0.001
<b>1995-1999 (1249 software, 2728 non-software)</b>								
R&D/Assets	<b>0.301</b>	<b>0.295</b>	<b>0.298</b>	<b>0.296</b>	<b>0.339</b>	<b>0.333</b>	<b>0.329</b>	<b>0.333</b>
Patents/R&D	0.001	0.000	0.000	0.000	0.012	<b>0.023</b>	<b>0.021</b>	<b>0.023</b>
SW Pats/R&D	<b>0.013</b>	<b>0.025</b>	<b>0.020</b>	<b>0.026</b>	<b>0.022</b>	<b>0.018</b>	<b>0.021</b>	<b>0.017</b>
R&D/Assets	<b>0.293</b>	<b>0.292</b>	<b>0.297</b>	<b>0.293</b>	<b>0.338</b>	<b>0.336</b>	<b>0.332</b>	<b>0.336</b>
Citations/Patents	0.001	0.001	0.002	0.001	<b>0.028</b>	<b>0.036</b>	<b>0.034</b>	<b>0.036</b>
SW Cites/Patents	<b>0.029</b>	<b>0.027</b>	<b>0.021</b>	<b>0.027</b>	<b>0.016</b>	<b>0.009</b>	<b>0.014</b>	<b>0.009</b>
R&D/Assets	<b>0.293</b>	<b>0.292</b>	<b>0.297</b>	<b>0.300</b>	<b>0.338</b>	<b>0.336</b>	<b>0.335</b>	<b>0.338</b>
Citations/Patents (excluding self-cites)	0.001	0.001	0.003	0.010	<b>0.029</b>	<b>0.037</b>	<b>0.038</b>	<b>0.037</b>
SW cites/Patents (excluding self-cites)	<b>0.029</b>	<b>0.026</b>	<b>0.021</b>	<b>0.011</b>	<b>0.016</b>	<b>0.009</b>	<b>0.009</b>	<b>0.006</b>

All regressions include a complete set of sector-year dummies and a dummy for missing R&D.

The quantities shown are the average elasticity of the market value with respect to the variable in the first column. The significance tests are based on the corresponding coefficient and its heteroskedastic-consistent standard error.

**Bold text indicates significance at the 5% level; Bold italics significance at the 1% level.**

**Table A3: Total number of firms and number of software firms in the sample, by year**

Year	Market value regressions			Event studies		
	All firms	oftware firm	App/Services	All firms	Software	App/Services
1980	406	47	23			
1981	511	76	41	184	25	17
1982	555	89	50			
1983	664	121	71			
1984	709	134	83			
1985	722	141	88			
1986	751	156	93			
1987	758	155	96			
1988	731	161	103			
1989	719	159	99			
1990	708	153	91			
1991	730	163	95			
1992	763	181	112			
1993	833	201	118	625	152	93
1994	889	216	120	706	175	95
1995	1,003	262	145	749	194	104
1996	1,065	295	144	864	238	123
1997	1,029	292	144			
1998	879	269	122			
1999	999	382	164			
<b>Total obs</b>	<b>15,424</b>	<b>3,653</b>	<b>2,002</b>			

## Appendix B

### Computing the CARs and Significance Tests

There are at least two ways to compute the cumulative abnormal returns to holding a share of stock. The first is that of Salinger (1992), which simply estimates a stock market beta model for each firm that includes dummies for the event days and then combines the estimates across firms. The model is

$$\begin{aligned} r_{it} &= \alpha_i + \beta_i r_{mt} + \varepsilon_{it} & t = -90, \dots, -30 \\ r_{it} &= \alpha_i + \beta_i r_{mt} + \lambda_t + \varepsilon_{it} & t = -1, \dots, 3 \end{aligned} \quad (1)$$

where  $r_{it}$  is the rate of return to security  $i$  at day  $t$ ,  $r_{mt}$  is the market return, and the  $\lambda$ s are the excess returns for the period  $(-1,+3)$  around the event. Because the event is common across firms, estimates from this set of equations using either seemingly unrelated regression on all firms or single equation estimation firm by firm will be identical. That is, there is no efficiency gain from joint estimation across firms because the right hand side variables are the same for each firm and no cross-equation constraints are imposed (Davidson and McKinnon 1993, p. 313).

Given estimates of the  $\alpha$ s,  $\beta$ s, and  $\lambda$ s, the CAR for firm  $i$  over the period  $(-1,3)$  can be estimated as follows:

$$CAR_i = \sum_{t=-1}^3 \hat{\lambda}_{it} \quad (2)$$

The standard error of this quantity (denoted  $s_i$ ) must be estimated by taking account of the full covariance matrix of the estimated parameters for each firm, but this is straightforward using the delta method.<sup>47</sup>

---

<sup>47</sup> This method is implemented by the ANALYZ command in TSP (Hall and Cummins 2005).

Given estimates of the CARs and their standard errors, the best way to compute the mean CAR across firms is to compute a weighted average, where the weights are the square root of the precision of the estimates (the inverse of the standard errors):<sup>48</sup>

$$\overline{CAR} = \frac{\sum_{i=1}^N CAR_i / s_i}{\sum_{i=1}^N 1/s_i} \quad (3)$$

Conditional on the  $s_i$ , the variance of this estimate is straightforward to compute:

$$Var(\overline{CAR}) = \frac{N}{\left(\sum_{i=1}^N 1/s_i\right)^2} \quad (4)$$

In practice, it is easy to implement this computation of the average CAR and its standard error using weighted least squares with weights equal to the inverse of the previously computed standard errors squared.

One problem with the Salinger approach is that if the event itself changes the risk of the security, a small amount of bias may be introduced by estimating the market model over the event period. An alternative that avoids this problem is to estimate the market model for a period prior to the event window and then compute the forecasted residuals for the event window period based on the estimates. In this case, assume we have estimates for  $\alpha_i$  and  $\beta_i$ . The estimate of CAR can then be computed as follows:

$$CAR_i = \sum_{t=-1}^3 (r_{it} - \hat{\alpha}_i - \hat{\beta}_i r_{mt}) \quad (5)$$

---

<sup>48</sup> This method is optimal if the excess return due to the event is the same for all firms. See Campbell, Lo, and MacKinlay, p. 162, for further discussion.

Once again, the standard error of this estimate is straightforward to compute using the delta method, which takes account of the error in the estimated  $\alpha$  and  $\beta$ . The mean CAR for any sample can be computed as described above.

We used both methods to estimate the CARs and found that the results were qualitatively similar and for the most part also quantitatively similar. In the case of the non-parametric rank test described below, the results were identical, which suggests that the ranking of the firm's returns was unaffected by the choice of estimation method. In the text we report estimates and tests based on those computed using the pre-event market model method.

### *Nonparametric significance tests*

We used the rank test method of Corrado (1989), which is also discussed in Campbell et al. (1997), Chapter 4. Using this testing approach is desirable in the case of stock market returns, which are frequently both non-normal and non-symmetric. It corresponds to asking whether the average of the within-firm ranks of the event period returns are approximately equal to the mean rank, as they would be under the null. The test is easily implemented using OLS regression once the ranks are available. First, we obtain estimates of excess returns for the entire period:

$$\hat{\lambda}_{it} = r_{it} - \hat{\alpha}_i - \hat{\beta}_i r_{mt} \quad t = -90, \dots, -30; -1, \dots, +3 \quad (6)$$

The estimated  $\alpha$  and  $\beta$  are based on the pre-event period (-90, -30). For each period  $t$ , the returns are ranked and assigned numbers from 1 to  $N$  (where  $N$  is the number of firms in the sample), yielding a set of series  $R_{it}$  that correspond to the  $\lambda_{it}$ . Under the null hypothesis of no abnormal returns, the expected value of the rank is a constant by construction, in this case equal to  $(66 + 1)/2 = 33.5$ .

To obtain the test-statistic and its standard error given by Corrado (equations 8 and 9 of that paper), one simply regresses the ranks  $R_{it}$  on a constant and a dummy variable that is equal to one if  $t$  is in the event period, and zero otherwise.<sup>49</sup> It is straightforward to show that the estimated coefficient of the dummy and its standard error correspond precisely to the equations given in the Corrado paper. The advantage of the regression approach is that it is easy to implement further tests: for example, to test whether the shares of software firms responded differently than those of hardware firms to the event, we simply include the event dummy interacted with a dummy for being a software firm.

---

<sup>49</sup> If desired, one can center the ranks before estimation by subtracting the mean and then there is no need for a constant. Centering the ranks would also be useful if each firm has slightly different numbers of observations (and therefore the expectation of the mean rank differs across firms).

Figure 1  
Software Patents Granted

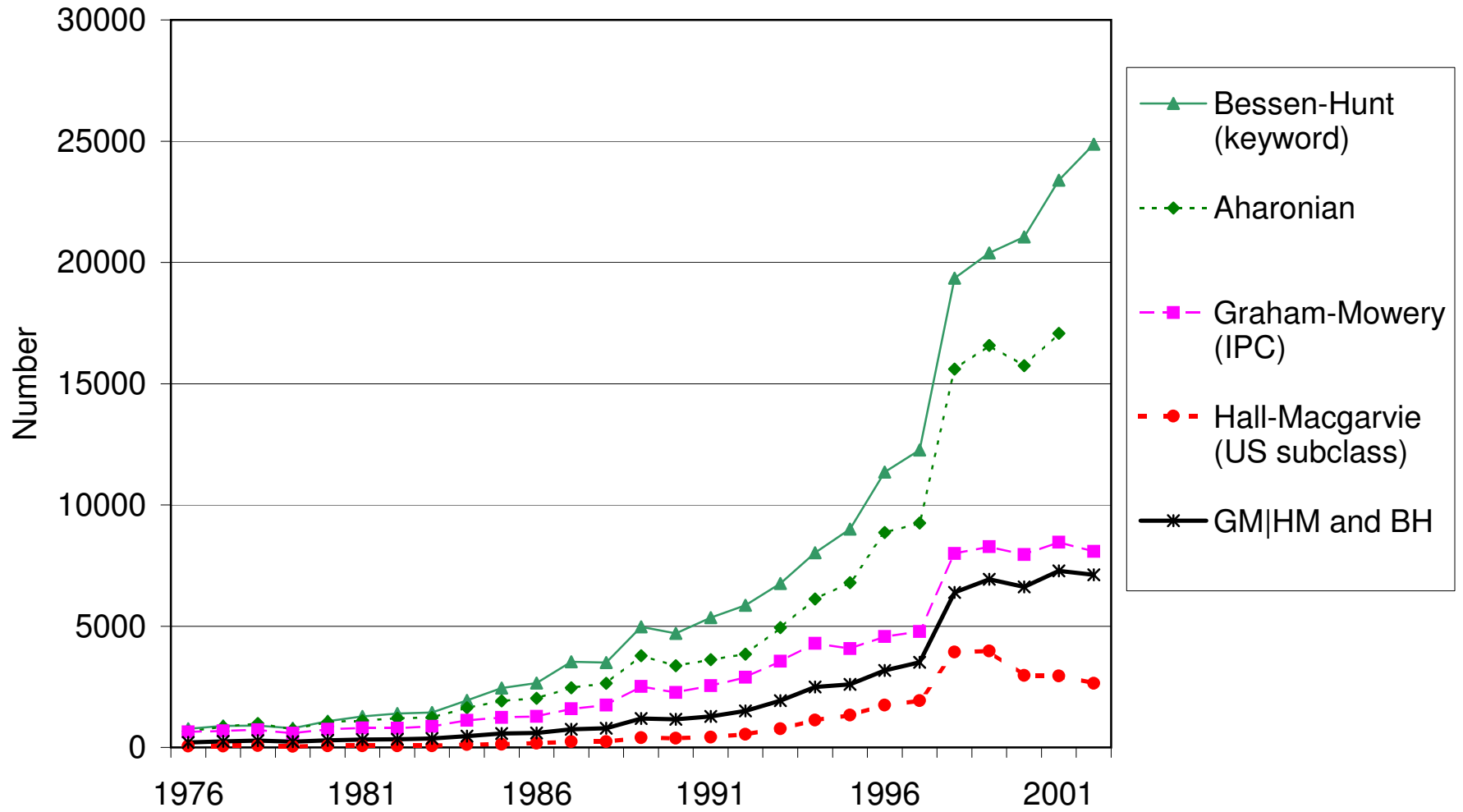
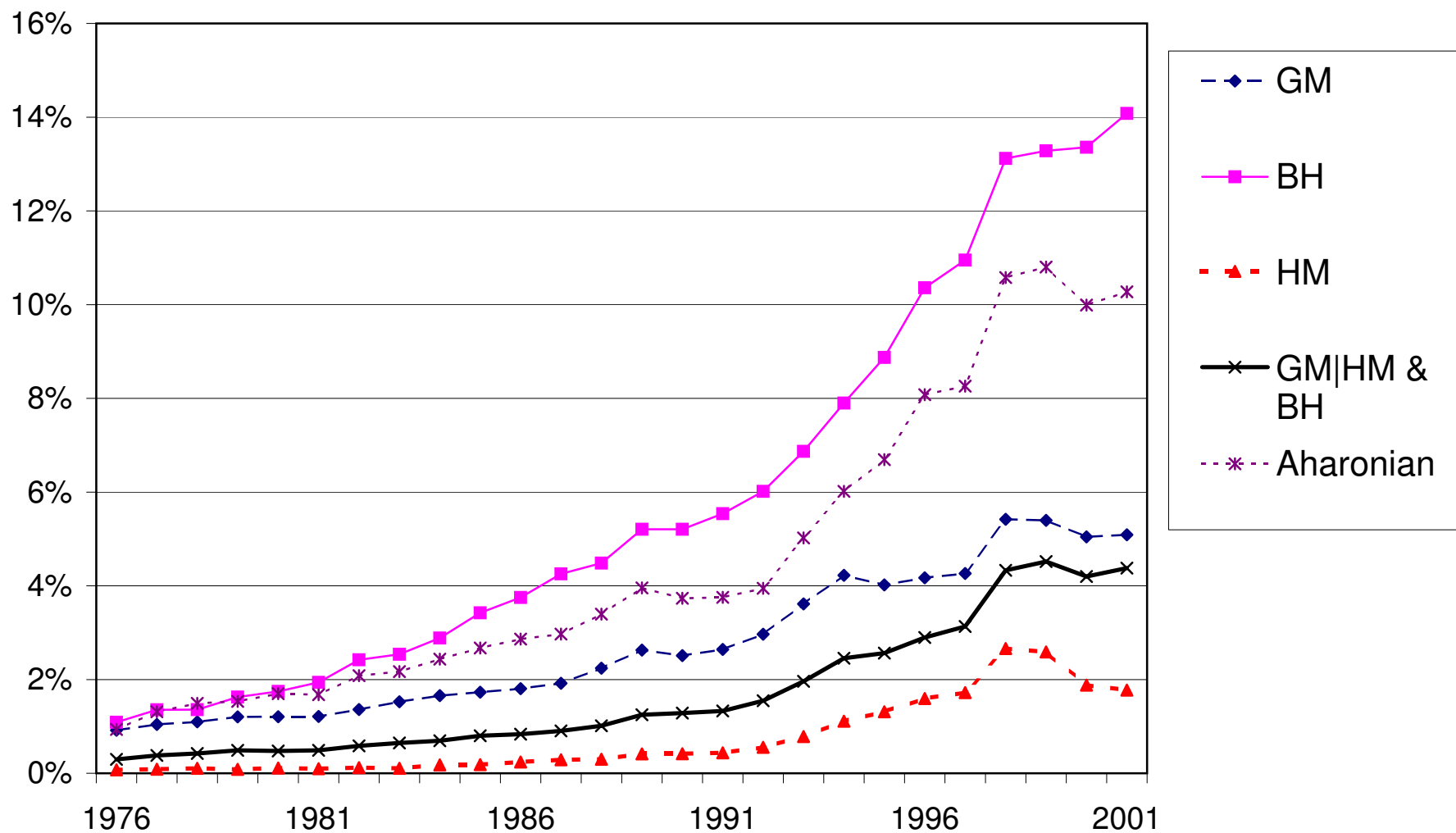


Figure 2  
U.S. Software patents granted as share of all patents granted



**Table 1**  
**Overlap with Allison's Samples**

Definition	Original software patent list (2000)		Internet business method patent list		Union of the two lists	
	Number	Share	Number	Share	Number	Share
<b>Allison+</b>	<b>100</b>	<b>100%</b>	<b>230</b>	<b>100%</b>	<b>330</b>	<b>100%</b>
Bessen-Hunt (BH)	92	92%	211	92%	303	92%
Graham-Mowery (GM)	60	60%	89	39%	149	45%
Hall-Macgarvie (HM)	40	40%	150	65%	190	58%
Combined*	83	83%	183	80%	266	81%

The table shows the number of patents according to each definition that are found in the lists supplied by Allison.

+The original list is the set of patents given in Allison and Lemley (2000); the internet business method list is that from Allison and Tiller (2003).

\*This set of patents is the union of GM and HM intersected with BH.

**Table 2**  
**Summary statistics for 2,015 firms 1980-1999 (15,424 observations)**

Variable	Mean	Standard deviation	Median	Minimum	Maximum
Number of employees	5.90	27.20	0.39	0.00	854.05
Tobin's Q (market-to-book ratio)	5.14	5.90	2.43	0.01	20.00
R&D stock (millions of \$96)	221.53	1421.39	9.73	0.00	31786.18
R&D Stock / tangible (book) capital	1.17	1.41	0.63	0.00	5.00
D(no R&D this year)	0.16	0.37	0.00	0.00	1.00
Patents per year	13.24	105.85	0.00	0.00	3474.00
Patent stock	58.05	453.77	0.09	0.00	14162.93
D(no patent stock this year)	0.49	0.50	0.00	0.00	1.00
D(no software patent stock this year)	0.82	0.38	1.00	0.00	1.00
D(never patented)	0.42	0.49	0.00	0.00	1.00
SW patent stock (B-H)	11.56	123.21	0.00	0.00	6371.54
SW patent stock (G-M)	8.54	94.43	0.00	0.00	4643.99
SW patent stock (H-M)	2.66	38.38	0.00	0.00	2120.75
SW patent stock (Combined)	5.92	78.37	0.00	0.00	4433.09
Patent stock / real R&D stock	0.30	0.98	0.00	0.00	30.74
Cite-weighted patents per year*	199.20	1534.40	0.00	0.00	50992.52
Cite-weighted patent stock*	955.05	7392.21	0.26	0.00	221306.70
Cite-wtd SW patent stock (B-H)*	256.58	2503.80	0.00	0.00	105724.60
Cite-wtd SW patent stock (G-M)*	186.77	2027.35	0.00	0.00	81877.63
Cite-wtd SW patent stock (H-M)*	62.98	858.79	0.00	0.00	42103.28
Cite-wtd SW patent stock (combined)*	139.66	1703.01	0.00	0.00	77272.84
Cite-wtd patent stock / real R&D stock*	5.11	21.86	0.00	0.00	1322.71
Cite-weighted patents per year**	164.10	1231.60	0.00	0.00	40252.71
Cite-weighted patent stock**	794.32	5972.43	0.25	0.00	173200.80
Cite-wtd SW patent stock (B-H)**	218.42	2018.24	0.00	0.00	81386.52
Cite-wtd SW patent stock (G-M)**	159.98	1609.03	0.00	0.00	61997.41
Cite-wtd SW patent stock (H-M)**	53.25	662.32	0.00	0.00	31096.84
Cite-wtd SW patent stock (combined)**	118.96	1327.87	0.00	0.00	57605.03
Cite-wtd patent stock / real R&D stock**	4.52	16.56	0.00	0.00	585.38

\*Corrected for citation truncation after 2002

\*\*Corrected for citation truncation after 2002 and self-citations excluded.

**Table 3**  
**Excess Returns [CAR(-1,+3)] to ICT Firms around Various Software Patenting Events**

Event	Date	Description	CAR (-1,+3) in %					Rank test for differences in CAR	
			All firms	Software firms	App SW/ SW services only	firms without patents	firms without SW patents	without patents vs with	without SW pats vs with
Diamond v Diehr	3-Mar-81	patenting allowed for software embodied in rubber-curing process (in a physical process)	0.83***	1.26	1.40	1.40***	0.97***	0.91	1.40
Compton patent	16-Nov-93	assertion of multi-media display patent held by Compton	-1.65***	-2.37***	-2.47***	-1.79***	-1.76***	-0.37	-0.54
Compton revocation	28-Mar-94	Compton's multi-media patent re-examined by USPTO and revoked	-0.36	-0.41	-0.73	-0.30	-0.56	0.60	-0.54
In re Alappat (5440676)	29-Jul-94	33 F.3d 1526 (Fed. Cir. 1994) (en banc). Narrowing mathematical algorithm limitation on patentable subject matter and thereby expanding patentable subject matter. - oscilloscope software partly affirmed, partly reversed PTO's rejection of a data structure for collision avoidance systems as non-patentable subject matter - machine containing the data structure created by the mechanism patentable.	-0.26**	-0.02	0.25	-0.05**	-0.32**	0.65	-1.46
In re Warmerdam (6089742)	11-Aug-94	data structures are patentable if held in a machine (printed matter doctrine does not apply) - confirming Warmerdam	0.08	0.34	0.32	0.05	-0.01	-0.86	-1.39
In re Lowry	26-Aug-94	A new way to calculate a number cannot be recognized as statutory subject matter - countered growing trend of SW patentability	0.16	-0.50	-0.69	0.36	0.23	1.35	0.86
In re Trovato	19-Dec-94	IBM appealed a decision to exclude using printed matter doctrine, PTO admits mistake (had contradicted <i>in re Lowry</i> )	-0.12	0.25	0.19	-0.04	-0.05	0.33	1.09
In re Beauregard (5710578)	12-May-95	new guidelines proposed to allow patenting on software embodied in any hardware medium.	0.76***	0.90**	-0.40	0.32	0.47***	-2.44***	-3.06***
USPTO new guidelines proposed	30-May-95	new guidelines issued allow patenting on software embodied in any hardware medium.	-1.00***	-1.43***	-1.04**	-0.85**	-0.86***	0.33	1.47
USPTO new guidelines final	29-Mar-96		0.36**	-0.61	-0.32	0.35*	0.38*	-0.56	0.01

Each cell in the first panel shows the average Cumulative Abnormal Return in per cent for a 5 day window (-1,+3) around the event date.

The cells in the second panel show the result of a rank test for differences in the distribution of the CARs during the event period.

\*\*\*, \*\*, and \* denote significance at the 1, 5, and 10 per cent level respectively, using a rank test.

**Table 4**

**Market Valuation of Software Patents (combined definition)**

	All years (1980-1999)			1980-1994			1995-1999		
R&D stock/assets	0.547 (0.025)	0.552 (0.025)	0.554 (0.025)	0.630 (0.036)	0.636 (0.036)	0.637 (0.036)	0.441 (0.034)	0.444 (0.034)	0.448 (0.034)
Pats/R&D stock	0.069 (0.015)			0.058 (0.015)			0.163 (0.055)		
SW pats/R&D stock	1.063 (0.295)			0.245 (0.196)			3.551 (0.698)		
Cite-wtd pats/ R&D stock		0.011 (0.001)			0.010 (0.001)			0.017 (0.004)	
Cite-wtd SW pats/ R&D stock		0.022 (0.008)			0.006 (0.008)			0.072 (0.020)	
Cite-wtd pats excl. self- cites/ R&D stock			0.012 (0.001)			0.011 (0.001)			0.020 (0.004)
Cite-wtd SW pats excl. self- cites/ R&D stock			0.027 (0.018)			0.004 (0.010)			0.144 (0.063)
No R&D	0.244 (0.035)	0.268 (0.036)	0.269 (0.036)	0.257 (0.042)	0.285 (0.043)	0.285 (0.043)	0.253 (0.065)	0.257 (0.065)	0.257 (0.065)
Log likelihood	-20106.9	-20037.2	-20040.2	-13466.5	-13414.6	-13414.9	-6605.2	-6596.8	-6600.3
Chow test (DF=4)*	<b>70.4</b>	<b>51.6</b>	<b>39.6</b>						
Observations (firms)		15,424 (2,015)			10,449 (1,418)			4,975 (1,469)	
<b>Elasticities**</b>									
	<b>Patents</b>	<b>Cite-weighted patents</b>	<b>Cite-wtd patents, excl. self-cites</b>	<b>Patents</b>	<b>Cite-weighted patents</b>	<b>Cite-wtd patents, excl. self-cites</b>	<b>Patents</b>	<b>Cite-weighted patents</b>	<b>Cite-wtd patents, excl. self-cites</b>
R&D stock/assets	<b>0.285</b>	<b>0.283</b>	<b>0.284</b>	<b>0.280</b>	<b>0.277</b>	<b>0.278</b>	<b>0.300</b>	<b>0.302</b>	<b>0.305</b>
All pats/R&D stock	<b>0.013</b>	<b>0.028</b>	<b>0.027</b>	<b>0.013</b>	<b>0.029</b>	<b>0.027</b>	<b>0.019</b>	<b>0.030</b>	<b>0.030</b>
Software pats/R&D stock	<b>0.006</b>	<b>0.004</b>	<b>0.001</b>	0.001	0.001	0.000	<b>0.022</b>	<b>0.011</b>	<b>0.005</b>

\*This is the likelihood ratio test for the hypothesis that the slopes are the same in the two subperiods.

\*\*The quantities shown are the average elasticity of the market value with respect to the variable in the first column. The significance tests are based on the corresponding coefficient and its heteroskedastic-consistent standard error.

The method of estimation is nonlinear least squares based on equation (4) of the text.

All regressions include a complete set of year dummies for each 2-digit sic (machinery, elec. machinery, instruments, telecomms, software).

**Bold text indicates significance at the 5% level; Bold italics significance at the 1% level.**

**Table 5**  
**Market Valuation of Software Patents (combined definition)**  
**By sector and time period**

	<b>1980-1994</b>		<b>1995-1999</b>		<b>1980-1994</b>		<b>1995-1999</b>	
	<b>Hardware</b>				<b>Software</b>			
R&D stock/assets	0.671 (0.045)	0.680 (0.046)	0.699 (0.061)	0.716 (0.062)	0.644 (0.068)	0.660 (0.070)	0.257 (0.037)	0.253 (0.037)
Pats/R&D stock	0.060 (0.016)		0.200 (0.067)		-0.472 (0.098)		0.018 (0.014)	
SW pats/R&D stock	0.200 (0.345)		3.801 (1.042)		0.844 (0.217)		3.640 (0.952)	
Cite-wtd pats excl. self-cites/ R&D stock		0.012 (0.002)		0.023 (0.005)		0.001 (0.004)		0.018 (0.013)
Cite-wtd SW pats excl. self-cites/ R&D stock		-0.001 (0.018)		0.084 (0.081)		0.026 (0.020)		0.223 (0.077)
Log likelihood	-10,201.7	-10,151.2	-4,630.1	-4,626.9	-3,150.7	-3,155.3	-1,939.6	-1,943.1
Observations (firms)	8296		3475		2153		1500	
	<b>1980-1994</b>		<b>1995-1999</b>		<b>1980-1994</b>		<b>1995-1999</b>	
	<b>Systems SW/ Middleware</b>				<b>Applications SW/Services</b>			
R&D stock/assets	0.763 (0.131)	0.729 (0.126)	0.325 (0.064)	0.309 (0.061)	0.550 (0.076)	0.581 (0.080)	0.189 (0.041)	0.193 (0.041)
Pats/R&D stock	3.555 (1.299)		0.026 (0.016)		-0.475 (0.083)		-0.052 (0.335)	
SW pats/R&D stock	1.407 (3.162)		5.396 (1.505)		0.666 (0.181)		1.498 (1.282)	
Cite-wtd pats excl. self-cites/ R&D stock		0.158 (0.044)		0.017 (0.015)		-0.002 (0.004)		0.026 (0.024)
Cite-wtd SW pats excl. self-cites/ R&D stock		0.839 (0.464)		0.303 (0.109)		0.014 (0.009)		0.137 (0.149)
Log likelihood	-1,238.1	-1,233.9	-1,032.4	-1,037.7	-1,877.2	-1,883.7	-899.1	-898.5
Observations (firms)	870		781		1283		719	

The method of estimation is nonlinear least squares based on equation (4) of the text.

All regressions include a complete set of year dummies for each 2-digit sic (machinery, elec. machinery, instruments, telecomms, software) and a dummy for firms with no R&D data.

**Bold text indicates significance at the 5% level; *Bold italics* significance at the 1% level.**

**Table 6**

**Elasticities from market value regressions (combined definition)  
By sector and time period**

	Hardware		Software		Systems SW/ Middleware		Applications SW/Services	
	1980-1994	1995-1999	1980-1994	1995-1999	1980-1994	1995-1999	1980-1994	1995-1999
Number of observations	8296	3475	2153	1500	870	781	1283	719
R&D stock/assets	<b><i>0.270</i></b>	<b><i>0.331</i></b>	<b><i>0.364</i></b>	<b><i>0.303</i></b>	<b><i>0.376</i></b>	<b><i>0.339</i></b>	<b><i>0.340</i></b>	<b><i>0.257</i></b>
Pats/R&D stock	<b><i>0.016</i></b>	<b><i>0.027</i></b>	<b><i>-0.018</i></b>	0.001	<b><i>0.029</i></b>	0.001	<b><i>-0.028</i></b>	-0.001
SW pats/R&D stock	0.001	<b><i>0.020</i></b>	<b><i>0.005</i></b>	<b><i>0.030</i></b>	0.004	<b><i>0.052</i></b>	<b><i>0.005</i></b>	0.009
R&D/Assets	<b><i>0.267</i></b>	<b><i>0.338</i></b>	<b><i>0.363</i></b>	<b><i>0.305</i></b>	<b><i>0.364</i></b>	<b><i>0.340</i></b>	<b><i>0.332</i></b>	<b><i>0.258</i></b>
Citations excl self-cites/R&D stock	<b><i>0.035</i></b>	<b><i>0.041</i></b>	0.001	0.010	<b><i>0.026</i></b>	0.011	-0.001	0.009
SW Citations excl self-cites/R&D stock	0.000	0.002	0.001	<b><i>0.010</i></b>	0.013	<b><i>0.018</i></b>	0.001	0.004

Significant at the 1% level: ***bold italic*** ; at the 5% level: **bold**.

**Simple Statistics for Patent Intensities**

Mean Patents/R&D*	0.415	0.288	0.043	0.079	0.031	0.112	0.052	0.033
Median Patents/R&D*	0.027	0.034	0.000	0.000	0.000	0.000	0.000	0.000
IQ range Patents/R&D*	0.0-0.37	0.0 - 0.29	0.0 - 0.0	0.0 - 0.016	0.0 - 0.0	0.0 - 0.036	0.0 - 0.0	0.0 - 0.001
Mean SW Patents/R&D*	0.010	0.014	0.013	0.027	0.011	0.042	0.014	0.011
Median SW Patents/R&D*	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
IQ range SW Patents/R&D*	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.012	0.0 - 0.0	0.0 - 0.0