Portfolio Patents and the Survival of Public Software Firms

Alexander Wallace Trahey

Professor Kent Kimbrough

Advisor Arie Beresteanu

Duke University

Durham, North Carolina

April 2010

Honors thesis submitted in partial fulfillment of the requirements for Graduation with Distinction in Economics in Trinity College of Duke University

Acknowledgements

I am very thankful for the help of Dr. Arie Beresteanu and Dr. Kent Kimbrough in completing this research. Also, the comments from my classmates in Econ 199S were immensely helpful in revising the paper.

Abstract

Recent legal changes relaxing patent restrictions have made patents in the software industry increase by a large margin. There is a growing sentiment that many firms are building portfolios of patents that do not contribute to innovation, leading to market inefficiency. This paper distinguishes between portfolio and innovative patents through estimating a patent production function and then determines what effects each class of patent has on its firm's chances of survival in the technology industry between 1994 and 2006. Innovative patents are found to have a large and significant positive effect on firm survival, while portfolio patents have zero effect.

I. Introduction

Over the past two decades, legal decisions in the US have led to a drastic increase in the number of software patents filed. The first software-related ruling came in 1972, when *Gottschalk v. Benson* stated that software could not be patented. In 1981, software became patentable through *Diamond v. Diehr* in the context of physical processes only. This meant that any piece of software that ran a manufacturing process or some other machine could be patented, but the algorithms could not. In the 1994 decision *In re Alappat*, the patent restrictions were significantly lowered when unpatentable software was determined to be "a disembodied mathematical concept...which in essence represents nothing more than a 'law of nature,' 'natural phenomenon,' or 'abstract idea.'" Patentable software included "a specific machine to produce a useful, concrete, and tangible result." (Cockburn and MacGarvie 2006)[1]. The law continued to be relaxed until the newest guidelines in 1995 allowed any software on physical media to be patented.

While there is extensive literature regarding the effects of patents in generic firms and the pharmaceutical industry specifically, the literature on software patents is still in its infancy.[2] A common trend observed by economists is the creation of software patent portfolios by firms that do not publish software. There are many firms which engage in "patent trolling", or the systematic collection of patents in order to charge licensing fees or pursue lawsuits. There is theoretical consensus that patents provide a tradeoff between incentives to innovate and positive network effects of sharing ideas freely. The ideal equilibrium between these tradeoffs is still entirely undecided, and must be explored

[1] Original source *In re Alappat*, 33 F. 3d 1526, 1544 (Fed. Cir. 1994), quoted from Cockburn and MacGarvie.
[2] Many of the papers cited here are working papers, and thus are subject to some change.

further. Another potential benefit of patents is the increased propensity to share ideas after having them protected by law. While an idea is protected, an inventor can publish the details of the invention without concern of losing his rights. This allows other inventors to extend the work if licensing is available, but problems can still arise if the published information is incomplete or the owner will not provide a license. Whether or not the length of monopoly is too long for the sharing to have a positive effect is unclear.

The increase in patenting has led to what many label "patent portfolios". Many firms have taken advantage of lighter restrictions in order to build up a collection of patents for legal purposes alone. While many potential patent disputes are never brought to court, firms have the ability to fight both defensively and offensively against rivals while holding patent portfolios. A number of economists have mentioned the existence of patent portfolios in relation to inefficiency, but few have explicitly studied it.[3] While explicit distinction between portfolio patents is not present in any patent studies, many recognize the importance of quality in determining the value of patents (Cockburn and MacGarvie, 2006; Cockburn and Wagner, 2007; Hall and Trajtenberg, 2000). The distinction between these types of patents is the focus of this paper.

This paper shows how the ownership of innovative patents in contrast with portfolio patents affects firm survivability in the software industry from the mid nineties until 2006. I find that innovative patents, or those representing a true invention and that are based on a solid foundation of research, have a positive significant effect on the chance of survival. On the other hand, portfolio patents, or those obtained strictly for legal battles and without invention, have an insignificant impact on survival. In this study

[3] An overview of this discussion is given by Bronwyn Hall, 2009

I separate each firm's patents into innovative patents and portfolio patents through a Cobb-Douglas production function. I assume that the increased patent production from relaxed restrictions is captured with time dummies, allowing the production function to capture innovative patents with pre-1994 coefficients and portfolio patents with the error term. The separate classes of patents are then used as inputs to a hazard function in order to estimate the chances of survival for publicly traded firms.

The structure of this paper is as follows: I begin with a review of relevant literature in section II. Section III covers the theoretical background, section IV describes the data used in this study, and section V explains the empirical methodology. I end with a discussion of the results and some concluding remarks.

II. Literature Review

The literature on patents covers a number of areas, with several focused around the value of software patents to companies. Schwartz (2003) discusses the individual value of patents by framing them as real options, coming to unclear conclusions. Hall and MacGarvie (2006) analyzed the aggregate value of patents to software firms, finding that initially patents cause harm to the entire industry, but eventually create positive value for some sectors. Hall, Thoma and Torrisi (2007) estimate the value of individual patents between the United States and the European Union, finding that US patents are valued positively while European patents are of insignificant value. This paper takes the lessons regarding patent production functions and patent citations from the study of patent value and applies them to the study of firm survival, an area which currently lacks these features.

Two recent studies regarding exit in relation to software are relevant to this research. Cockburn and Wagner (2007) study the effect of patent ownership on firm survival, finding that ownership of software patents increases the chances of survival throughout the technology bubble. Cockburn and MacGarvie (2006) find that more patents within an industry sector slow both entry and exit. Neither study accounts for any measure of patent quality, a key component of this paper. This study uses methods similar to those of Cockburn and Wagner to study survival while adding a distinction of quality in innovative and portfolio patents.

The upward spike in software patenting has likely been a direct result of changes in patent law rather than the technology boom in the nineties. Bessen and Hunt (2007) claim that neither investment in software, R&D, nor the employment of software engineers explains the increased growth in software patents. They also find that only 5% of software patents are owned by software publishers, leading to the conclusion that strategic portfolios are the driving force behind the growth. The ability to patent software so readily came from court decisions in the mid nineties, beginning with *In re Alappat* in 1994, increasing the cost effectiveness of patenting software (Bessen and Hunt, 2007). The aim of this paper is to identify the patents involved in this spike and to analyze their worth.

In order to accomplish this goal, this study employs a form of a knowledge production function to estimate the number of patents produced. There exists extensive literature on knowledge production, with a very small portion related to software specifically. Pakes and Griliches (1984) come to the conclusion that past success in patent production matters in current production. With a wealth of patents for a firm to build on, production in the future is significantly easier. There is consensus that research and development spending is a key factor in the production of knowledge, though some controversy exists surrounding the effects of R&D on firm performance directly.[4] By using R&D as an estimator to a patent production function which will then be an instrument for a measure of firm performance, this study can bypass some of that problem. Some authors (Hausman, Hall, and Griliches, 1984) have found that R&D spending exhibits constant returns to scale in patent production, but others (Crepon and Duguet, 1996) found the opposite. This study maintains a Cobb-Douglas production function which can exhibit various returns to scale based on coefficients.

After estimating a production function for patents, this study also creates a measure of patent quality. A number of studies have attempted to model patent quality using measures of citations (Lanjouw and Schankerman, 2004; Hall, Jaffe, and Trajtenberg, 2001). While citations are a good measure of quality, the software patents in question may still be too young to rely on citations alone, as it takes nearly 20 years for the full scope of citations to play out (Hall, Jaffe, Trajtenberg, 2001). This study makes a concrete distinction between invention based and portfolio patents rather than using a quality scale.

The value of individual patents is another area with important literature. While this study does not use stock price or a related measure to value patents, much of the framework in patents' effects on firm performance is still applicable. Schwartz (2003) discussed the value of patents in the health care industry as real options. In pharmaceuticals, there is a much larger debate about patenting behavior because of the

[4] See Hall and Mairesse (1995) and Griliches, Hall, and Pakes (1988).

volume of money that drug companies bring in through monopolies on their research. There is also considerable risk of research leading to nothing, resulting in the treatment of these patents as options by Schwartz (2003): there is always an option to back out of a research track if the projected value is too low. Though unable to empirically analyze the model, Schwartz created a framework for valuing a patent through different stages of development.

A different look at single patent value was done by Hall, Thoma and Torrisi (2007), in which the market value of the patents is directly estimated. The study analyzes European firms that have taken out patents in the European Patent Office (EPO) and the US Patent and Trademark Office (USPTO). Some of the firms had patents in only the EPO, while others had patents duplicated in both offices. The result of the analysis was a positive value of patents owned in both offices or the USPTO, but an insignificant value for those in only the EPO. This provides an interesting caveat to the study of general patent economics, as conditions in the US may be too unique to make general statements about patent implications on firms.

On an industry wide scale, the aggregate value of patents to software firms was found to be initially negative, but eventually positive for firms that are not dependent on licenses for compatibility (Hall and MacGarvie, 2006).[5] The study separates firms into a number of different categories, such as the level of dependence on other firms, labeled "upstream" and "downstream". It distinguishes between patents associated with R&D increases and patents that go above and beyond the level of R&D in which a firm invests,

[5] Firms with end-user software, software used by an individual on their computer, frequently must license the ability to work with Microsoft Windows or Mac OS as well as various pieces of hardware that drive functionality. At the level of operating system developer, holding these patents gives value in the form of licensing fees and requires very little licensing from other firms.

signaling strategic rather than inventive patents. I find this to be a key feature of the study, and apply the distinction to a different model that lacks it.

Knowing how patents apply to firm performance, the next step is to relate performance to survival for the context of this study. I assume that exit is related to failure in the first stage and then go on to distinguish between failure and acquisition. Exit in the context of this study is delisting from a public stock exchange. When a firm is delisted from a public stock exchange, there are several possibilities for why it happened. It might have gone bankrupt, been privately bought out, or merged with another company. Dunne, Roberts and Samuelson (1988) find that age and size are the two dominating factors in determining the whether or not a firm will exit an industry. Another result of their analysis is the correlation between entry and exit within an industry. Any industry with a high number of entrants will also have a high number of exiting firms. The time period of focus for my study within the software industry has a high number of entrants and a high percentage of firms as small startups, so we should expect to see many of the firms exit. This study distinguishes firms which are delisted due to acquisition from firms which are delisted due to failure or unknown reasons.

The survivability of firms throughout the burst of the technology bubble was found to be significantly related to the number of patents held by the company (Cockburn and Wagner, 2007). The study controls for a number of financial statistics, including age, venture capital backing, income, and a measure of liquidity, but fails to include any statistic on research and development. Firms patenting with higher R&D should have a higher rate of survival than firms that patent strategically. In this paper, I extend the results of Cockburn and Wagner's study to include such measures as generally outlined by Hall and MacGarvie (2006) and Bessen and Hunt (2007). In other words, I create a hazard function of failure for IPO firms as Cockburn and Wagner did, and I identify portfolio patents through a patent production function and a measure of patent quality used by Hall and MacGarvie. No previous attempt has been made to explicitly identify portfolio patents, and the theoretical basis for patents would imply vastly different effects.

Cockburn and Wagner find more specifically that not only is survival positively affected by patents, the chance for a failing company to be acquired is significantly higher for companies holding patents. In an attempt to measure the value of individual patents, the study includes a measure for citations on patents, finding little significant difference except for a slight increase in the chance of acquisition while having highly cited patents.

One interesting extension which is closely related to firm survival is the study of entry and exit in software done by Cockburn and MacGarvie (2006). The study finds that patents within an industry sector slows entry and a potential entrant's ownership of patents increases the likelihood of entry. The study finds the same general conclusions on exit as with survival, with the key distinction being any firm exiting at any time rather than public firms collapsing after the dot com bubble. The study has the same shortcomings in the lack of treatment of research and development. Cockburn and MacGarvie claim that "R&D spending may be a poor proxy for the rate of innovation" (2), but then go on to focus on the number of patents as the desired proxy. With patents as a function of R&D, my paper is able to expand on these results.

With millions of patent applications and grants, a key consideration for any data set is how that set will be classified and organized. For classifying software patents in particular, there are several distinct methods. The most basic of these methods is the use of the classification code used by the USPTO. While this provides a simple solution to the question, the USPTO uses the codes to find prior art rather than to classify data for researchers, so many software patents are found in other classifications and many non-software patents are found in the software class (Hall and MacGarvie, 2006). There are three main definitions outside of the USPTO created by economists, begun by Graham and Mowery (2003). The Graham-Mowery definition includes patents in a subset of the USPTO classes and subclasses, decided by looking at the largest software firms in 2003 and determining which classes their patents were concentrated in. The second definition was created by Bessen and Hunt (2003), and is a significantly more customized definition.[6] The third definition is that of Hall and MacGarvie (2006), which includes all USPTO classes and subclasses in which fifteen of the largest software companies owned patents.

With the implications of legal changes on patents still very much under debate, the study of patents' effects on firms is an important one for policy. Viewing the trend of strategic patents that many claim are of low quality, it may be true that the US needs to tighten patent requirements. It may also be true that weak requirements provide value to the economy for firms willing to use them. The aim of my paper is to look at one aspect of these patents, with innovative and strategic patents distinguished as accurately as

[6] Software patents by the Bessen-Hunt definition are those that have the word "software" or "computer" and "program" in the specification of the patent. Following that search, any patents including the words "chip", "semiconductor", "bus" or "circuitry" in the title where excluded, as they commonly were patents for machines running software rather than the software itself.

possible. While my research is an important angle in the debate, there are vastly different alternatives in the industry that present new questions. One such alternative is the growing open source movement, which may be a significantly more efficient way of producing innovative material. Restrictive patent law requires complex effort and financial resources that may not be available to open source development projects. Another policy alternative is specialized patent law for various industry sectors. Potentially, software, business methods and other high-tech patents could all be regulated independently, allowing for more control over the quality of patents. While software innovation has been described to be more collaborative and incremental than other technology sectors (Hall and MacGarvie, 2006), this seems to be an attractive option for future study.

III. Theoretical Framework

A patent provides the owner a temporary monopoly on an innovation, allowing the owner to better capitalize on the invention. The stated rationale behind allowing patents to exist is that patents spur innovation by incentivizing invention. An inventor will not spend time or money inventing a product if another company can steal the invention and sell it as soon as he invents it. On the contrary, economics states in most cases that monopolies are inefficient, and in the context of patents, they can slow subsequent inventions or eliminate the need to improve the product. While there are two forces competing against one other, the likely equilibrium comes somewhere within the length of patents or the barriers to getting a patent. A shorter length of monopoly would reduce the inefficiency that a monopoly brings, but it would also lessen the incentive to innovate. A higher cost of obtaining a patent would limit the number of monopolies on small or insignificant inventions and still allow significant inventions to be patented.

The ownership of monopoly on a certain product is a valuable economic right, and should contribute to a firm's value. A firm is more attractive to venture capitalists when it has a clear profit earning potential. Between the intellectual capital that a patent provides and its attractiveness to investors, we should expect a patent to decrease the chances of failure for its owner. The more valuable a patent is, the more it will decrease this chance. A patent that has no profit potential behind it should not contribute to firm value because a monopoly on an unsellable good is worth nothing. In fact, one might argue that outside of the legal power that a patent like this signals, it only creates inefficiency for the entire industry.

Within software, the networking effect of inventions is much higher. Many algorithms are used in multiple programs, and the digital format makes implementing other code extremely simple. The larger network effect would lead to the conclusion that monopolies create more inefficiency within software. If so many firms could benefit from using a new algorithm, the monopoly would bar significant growth. At the same time, the software industry is characterized by a large number of firms and inventors, due to the low barriers to entry and the low cost of capital needed to perform research. Patents allow such a large number of small firms to exist because they are not required to compete directly with large firms that have better access to the market. The theory gives us no clear answer to where the equilibrium should lie, but the software industry is significantly different from manufacturing and biochemical industries where patents are also prominent. Over the last two decades, the large increase in software patents and the volume of venture capital infused into the software industry speak to this difference.

Copyright also exists as a substitute for patent protection. Copyright laws protect the specific piece of work created by an author, but do not protect the mechanism or idea behind the work. In software, this prevents another person from using the actual code, but does not stop another firm from recreating the algorithm. We expect that stronger copyright protection would lower the propensity to patent and vice versa. Lerner and Zhu (2007) study the effect of *Lotus v. Borland*[7] on patent use, and conclude that weakening copyright law disproportionately increases patent use among software companies. The study also finds that the increased patent use has little to no effect on firm performance, leading to the potential conclusion that the use of patents does not increase or decrease innovation a great deal. Relatively little has been done comparing the uses of patents and copyrights in the software industry, and it remains a topic for further study.

IV. Data

This paper draws from a combination of data on publicly traded firms and from an updated version of the NBER patent and citations dataset. The patent assignees were matched to Compustat data using Hall's name matching algorithm. In some cases, multiple entries within the Compustat database represent one firm, so a unique identifier was given to each firm. For each patent listed in the database, a matching identifier PDPCO was given to represent the firm. Another data file exists where each Compustat

[7] This case weakens the copyright protection on software, specifically establishing that copyright does not protect the text or layout of a program's graphical user interface.

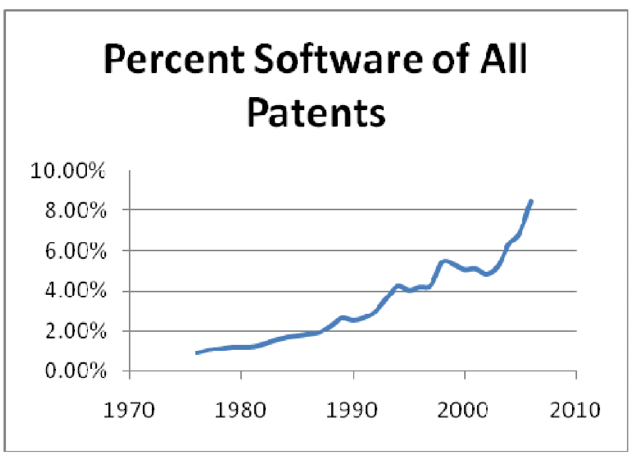
database entry is matched with the appropriate PDPCO identifier.[8] One concern with the data is the lack of awareness of private selloffs of patents. The database tracks ownership changes of patent-owning companies, assuming that such an ownership change results in the patent moving as well. It does not account for any situations in which an owner of a patent sells an individual patent to another entity. Within this dataset, there are four important data files. First, *phpcohdr.dta* contains each of the Compustat entities matched to their unique firms. Second, *assignee.dta* contains each of the names listed on patents and a matched identifier that combines multiple names representing the same owner. Many patents are assigned to the same company with different names, for example IBM vs. International Business Machines. Third, *dynass.dta* contains a dynamic match of patent assignees to their Compustat entities, giving the last five owners of each patent and the years they were owned. Last, *patassg.dta* contains each patent-assignee pair from 1976-2006.

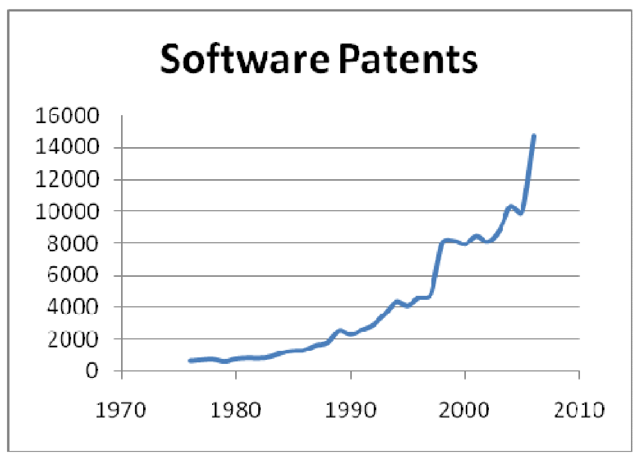
In order to obtain the subset of patents concerning software, I have used the methodology outlined by Graham and Mowery (2003). Specifically, a software patent is identified by a subset of International Patent Classification tags.[9] The end result reaches fewer software patents than does the Bessen and Hunt definition, but that definition is beyond the scope of my time and resources as it requires a running a keyword search through each of the three million patents in my data set. Hall and MacGarvie (2006) found that this definition accounts for about 57% of all patents assigned to the hundred largest software firms.

[8] The entire name matching algorithm was done by Bronwyn Hall and published along with more details at http://emlab.berkeley.edu/users/bhhall/pat/namematch.html, my effort consists of using the existing matched names to draw relevant firm information from the Compustat database.
[9] The classifications included are G06F: 3,5,7,9,11,12,13,15; G06K: 9,15; H04L: 9.

The software patents in my dataset are distributed among 11,692 firms, where roughly 60%(7,053) of those firms own only software patent, and only 10%(1,356) own more than five software patents. The average number of software patents held by a firm with at least one is 10.5, with a standard deviation of 161.7. Twenty firms own more than one thousand software patents, with the largest, IBM, owning nearly fourteen thousand. I have included here two graphs representing the patent data using the Graham and Mowery definition of software patents.

Between 1976 and 2006, the data contains 3,209,376 patents of which 128,757 are software patents by their classification. The graphs here show the total number of software patents granted in each year (left) as well as the percentage of all patents which are software patents (right). The graphs also show an accelerating growth of software





patents as the numerous cases loosened restrictions on software patents, leading to a much higher percentage of all patents distributed as to software. The increased percentage supports the hypothesis that software patents have changed uniquely due to legal changes while overall patenting has been increasing at a much slower rate.

The Compustat data was obtained through the Wharton Data Research Services web interface using the list of matching names as a filter. I have obtained data on all firms in a number of industry classes identified as related to software, as well as each specific firm included in the patent database. The data as used in this study is yearly financial data published by the firm in their annual reports. Each firm is identified by a *gvkey-year* pair. The *gvkey* is a unique identifier in the Compustat database for each given year, but it can be reused as a company fails or merges with another firm. Research and development expenditures, total assets and total employees were gathered from this database among other firm characteristics not currently used in this study's regressions. The data again was collected between the years 1976 and 2006, and the observations are by *gvkey-year* pairs. The data contains an average of just over 133 firms per year which contain all relevant pieces of data, though there are many more which have only one or two missing elements. Of all the observations in the dataset, 4004 have all five values available. Summary statistics are shown below.

Variable	Observations	Mean	Std. Dev.
Number of Patents	4551	12.93804	70.45268
R&D (millions of $)	4218	345.0738	809.6522
Age	4551	18.61459	14.943
Assets[10]	4544	12281.18	62179.68
Employees	4371	36.82595	84.14934

IV. Empirical Specification

In order to achieve the desired results, this paper studies the lifespan of firms within the Compustat database as a function of the number of patents they own. I use a Cobb-Douglas production function in order to estimate the number of patents, which is then used as an input to the hazard function of survival. Rather than using the actual patent numbers, estimating a production function and including time dummies for important legal events will allow me to distinguish between innovative patents and portfolio patents for a given firm. The number of patents for firm *i* in year *t is* estimated using measures of the firm's size, age, R&D and capital.[11]

The regression specifications follow:

$$(1)\ \boldsymbol{Patents_{it}} = \alpha_t * Employees_{it}^{\beta_1} * \frac{R\&D_{it}}{Employee_{it}}^{\beta_2} * \frac{Capital_{it}}{Employee_{it}}^{\beta_3} * Age_i^{\beta_4}$$

[10] I was unable to obtain a pure measure of capital for a given firm, so I use total assets as a proxy for capital.
[11] This is an adapted method from Bessen and Hunt (2007); here, a new entrant dummy is replaced by Age.

Equation 1 specifies the Cobb-Douglas production function given by the theory. Equation 2 follows as a regression where the log of both sides is taken and an error term is added.

$$(2)\ \ln(Patents_{it}) = \ln \propto_t + \beta_1 \ln Employees_{it} + \beta_2 \ln \frac{R\&D_{it}}{Employee_{it}} + \beta_3 \ln \frac{Capital_{it}}{Employee_{it}} + \beta_4 \ln Age_i + u_{it}$$

In Equation 3, X_{it} is the set of variables included in the previous equations, and Post1994 is a dummy equal to one after the year 1994, when the *In re Alappat* made software patents significantly easier to obtain.

$$(3)\ \ln(Patents_{it}) = \ln\alpha + \beta\, Post\mathbf{1994} + \beta \ln X_{it} + u_{it}$$

The final specification (4) allows for a production function that captures the changes in the effects of each independent variable represented by the vector X_{it}.

$$(4)\ \ln(Patents_{it}) = \ln\alpha + \beta \ln X_{it} + \beta\, Post1994 \ln X_{it} + \varepsilon_{it}$$

The results of (2), (3) and (4) are shown in Table 1 below.

Table 1: Patent Production Functions

Independent Variables	Eq. (2)	Eq. (3)	Eq. (4)
Employees	.3891 (.0303)**	.3917 (.0301) **	.2266 (.0328)**
Employees* Post1994	--	--	.2573 (.0236)**
Capital/Employees	.2573 (.0358) **	.2008 (.0371)**	.3033 (.0439)**
Capital/Emp. * Post1994	--	--	-.1584 (.0363)**
Age	.2660 (.0417)**	.2031 (.0431)**	.3844 (.0493)**
Age * Post1994	--	--	-.2389 (.0453)**
R&D/Employees	.3071 (.0362)**	.2906 (.0362)**	.03928(.0432)
R&D/Emp.*Post1994	--	--	.4368 (.0433)**
Post1994	--	.2419 (.0455)**	--
Constant	-1.870 (.149)**	-1.870 (.149)**	-1.927(.159)**
R^2	0.2929	0.2943	.2800
N	4004	4004	4004

* Denotes 95% significance

** Denotes 99% Significance

Standard errors in parentheses

The coefficients for all of the independent variables in the regressions matched expectation. The results for (2) show the effects of each input without any time dummy. Higher capital intensity and R&D intensity will produce a higher volume of patents for the firm, as employees have more resources to work with and can be more effective in producing the knowledge output. Larger firms as measured by the number of employees will also produce more patents, as they have more labor inputs. The total employees may have less to do with patent production than the number of researching employees, but due to data limitations, only total employees are included. Other support or sales staff may also contribute to the streamlining of knowledge production, so it seems reasonable to include all employees. The age of a firm also contributes positively to patent production, which can be explained by increased skill in production after years of experience.

Column (3), showing the addition of a time dummy representing the events of *In re Alappat*, again meets expectations on all accounts. As previously discussed, the court decision in 1994 loosened the restrictions on acquiring patents, so we expect to see firms produce more patents after 1994 given equal inputs.

In the results of (4), the changes in each variable over time are displayed. The coefficients of Employees, Capital intensity and Age all drop after 1994, but they all remain positive. One plausible explanation for these results is that software firms have become smaller and more agile as one individual or a small team can produce good software. Before the restrictions were lowered, significantly more infrastructure was required to produce a machine which demonstrated a patentable piece of software, while presently patentable software can be produced by almost anyone with a computer and a good enough idea. Firms like IBM and Microsoft still have huge infrastructure to support their programmers in the production of knowledge, so they are still more effective. The reduced magnitudes, however, point towards the lessening need for such infrastructure.

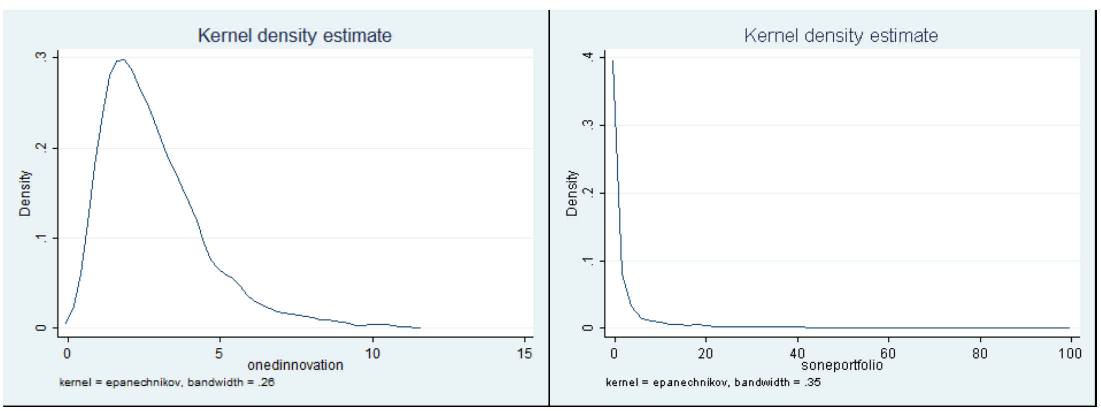
The R&D intensity coefficients add an interesting element to the story as well, as pre-1994 R&D intensity had an insignificant effect and post-1994 it had a large effect on patent production. We would expect R&D to always have a positive effect on patent production, so the insignificance is somewhat surprising. Firms before 1994 may not have spent as much on R&D as firms do now, or R&D may have been accounted for in a different way. It is also possible that the R&D affected other areas of patent production while not changing software patents. The data does not capture the percentage of R&D devoted to software. As software patents were harder to obtain, firms would spend less on software R&D and more on other R&D. This is likely to account for at least some of the result because the larger firms producing software patents also had divisions for hardware.

The results show significant changes after 1994 resulting in an increased number of patents produced. I assume that firms have a similar production function before and after 1994 outside of the court decision, so I use the coefficients from (3) and (4) excluding any post-1994 variables to predict the number of patents a firm should produce. These are considered innovative patents. Any patents found above that level are considered portfolio patents. Equation 4 shows this relationship, where $Patents_{it}$ is the estimated patents found from the production function in (3).

$$(5)\ PortfolioPatents_{it} = ActualPatents_{it} - EstimatedPatents_{it}$$

The portfolio patents are essentially the error term in the regression, but the large majority of the observations are positive due to the elimination of the post-1994 dummy variables. There still remain some negative observations, however; all of which need to be treated carefully. One could assume that negative portfolio patents simply mean better innovative patents, where if a firm spends a large amount of R&D and only comes out with a few patents, they must be of high quality. This approach makes some sense intuitively, but practically a firm can never have negative patents of any kind, so I use another measure. For any firm with negative portfolio patents, I assume every patent that it owns is an innovative patent and thus they have zero portfolio patents. Most firms have a small number of portfolio patents, but in 785 firm-years, more than five portfolio patents were produced. In an environment where most firms produce under ten patents in a given year, it is surprising to see so many portfolio patents. A total of 747 firm-year observations saw more than ten *total* patents produced, so most firms producing a large number of patents also produced several portfolio patents. Furthermore, of those 747, an average of 54.6 portfolio patents were produced as compared with 8.25 innovative patents. This data illustrates the large patent portfolio phenomenon in the industry. The graphs below show the distribution of innovative patents(left) and portfolio patents(right). I have only included the graphs from the single dummy equation (3), as the graphs produced from (4) look nearly identical and illustrate nothing new.

Figure 1: Innovative Patent and Portfolio Patent Densities



In order to determine the chance of survival, I construct a hazard function using patent characteristics of the firm as inputs. I employ a Cox proportional hazard model, which makes no assumptions about the hazard rate other than it being a function of the independent variables. The patent characteristics consist of a measure of innovative patents and of portfolio patents. The regression is of a form similar to that used by

Cockburn and Wagner (2007).[12]

$$(6)\lambda(i,t) = \lambda_0(t) * exp(PortfolioPatents_{it}\beta_1 + EstimatedPatents_{it}\beta_2)$$

where *EstimatedPatents*$_{it}$ is the estimated number of patents in the previous regression and *PortfolioPatents*$_{it}$ is the number of patents owned by a firm above the expected count. The results are displayed in Table 2. $\lambda(i, t)$ is the hazard of failure for firm i in year t.

Table 2: Cox Hazard Model Results (General Exit)

Independent Variables	(3) Single Dummy	(4) Dummy-inter.
Innovative Patents	0.8694**	0.8247**
	(0.00980)	(0.0109)
Portfolio Patents	.9995	.9994
	(0.00121)	(0.00105)

** Denotes 99% Significance

Standard errors in parentheses

Table 2 reports the hazard ratios given by the Cox regression. A value less than one signifies a lower likelihood of exit, where a value above one signifies a higher likelihood of exit. Column 1 presents the results using the regression with a single time dummy (Equation 3), and Column 2 uses the regression with time dummy interactions for each variable (Equation 4). The signs are as hypothesized, with innovative patents giving a firm a higher chance of survival and portfolio patents producing a very small decrease in the chance of survival. The first regression reports a hazard ratio of .869, meaning for

[12] The regression mentioned in this study is not exactly specified, so this is an approximation with some modifications.

each innovative patent a firm owns, it is 13.1% less likely to exit than its counterpart. This magnitude points to the ownership of one good patent being paramount to success, which is in line with Cockburn and Wagner's findings with regards to the first patent a company owns. Using the time dummy interaction variables to predict innovative patents, a firm is 17.5% less likely to exit for each innovative patent it owns. The magnitudes of these hazards are slightly different, but economically not different enough to cause concern. The portfolio patents in both cases have no effect on survival, which implies that the effort spent acquiring portfolio patents is not worthwhile in relation to survival.

Table 3: Cox Hazard Model Results (Failure vs. Acquisition)

Independent Variables	Eq. (3) Failure	Eq. (3) Acquisition	Eq. (4) Failure	Eq. (4) Acquisition
Innovative Patents	0.8552**	0.9301**	0.8603**	0.8736**
	(0.0121)	(0.0128)	(0.0158)	(0.0156)
Portfolio Patents	1.0004	1.0072**	.9998	1.00754**
	(0.00099)	(0.00175)	(0.00102)	(0.00142)
N	4004	1608	4004	1608
Failure[13]	666	666	666	666
** Denotes 99% Significance				
Standard errors in parentheses				

Table 3 presents the hazard data separated by failure and acquisition as the reasons for exit. They columns are labeled by single dummy and dummy interaction equations (3) and (4), and then are separated by the reason a company was delisted. Failure means a company went bankrupt or was delisted for undisclosed reasons, while

[13] This is failure in the sense of hazard event, not necessarily firm failure.

Acquisition means the firm was directly acquired by another public firm or privately bought out. In the columns labeled failure, every firm is included and only firms which failed were subject to the hazard, as opposed to Table 2 which included all delistings. In the columns labeled acquisition, the observations include only those firms which were delisted, and thus study the chances of failure versus acquisition. Because firms were labeled failed for unknown reasons, each of the firms that "survived" were actually acquired. The results continue to support the hypothesis that innovative patents improve a firm's chance of survival while portfolio patents are less effective if not ineffective. Firms are roughly 14% less likely to fail for each innovative patent. The magnitudes are very similar here as compared with Table 2. Similar to the results of previous studies, delisted firms are also between 7 and 13% less likely to fail over the alternative of acquisition and thus are more likely to be acquired for each innovative patent.

V. Discussion

By constructing a software patent production function based on values before legal changes made large portfolios common, this study is able to provide some distinction between classes of patents. The results present strong evidence that portfolio patents have little to no effect on survival, while innovative patents are quite effective in ensuring survival. Each innovative software patent reduces the chance of failure by between thirteen and eighteen percent, while portfolio patents have generally insignificant effects on survival. Such a reduction is economically significant, supporting the use of software patents for many firms. This study does not attempt to derive the aggregate value of either innovative or portfolio patents, so we can only conclude the usefulness of an innovative patent to a single firm. While the distinction between patents is imperfect at best, it is a useful distinction to make based on patenting behavior we observe in practice.

With such evidence against the usefulness of portfolio patents while others complain of their inefficiency, the question of appropriate patent laws is brought to question. Does software patenting need to return to a more restrictive state, or do more precise policies need to be created for different sectors of software? There are many options for dealing with patents, and further study of this topic is vital to our understanding of the matter.

A good avenue for future research might be to improve the distinction between innovative and portfolio patents through a number of measures. Several studies have used patent citations to measure quality on a continuous scale, and similar methods could be used in conjunction with methods presented here to create a more accurate measure of quality. While the separation of classes itself is a useful tool, looking at the same patent distinctions as they relate to other measures of firm performance such as stock price or Tobin's q would be another interesting extension.

References

Audretsch, D. B. , and T. Mahmood, 1995. New Firm Survival: New Results Using a Hazard Function, *Review of Economics and Statistics* 77, 97-103.

Bessen, J., and R. M. Hunt, 2007. An Empirical Look at Software Patents, *Journal of Economics and Management Strategy* 16, 157-189.

Cockburn, I. M., and M. J. MacGarvie, 2006. Entry, Exit and Patenting in the Software Industry, *NBER Working Paper Series*.

Cockburn, I. M., and S. Wagner, 2007. Patents and the Survival of Internet-Related IPOs, *NBER Working Paper Series*.

Dunne, T., Roberts, M. J., and Samuelson, L., 1988. Patterns of Firm Entry and Exit in U.S. Manufacturing Industries. *The Rand Journal of Economics* 19, 495-515.

Graham, S. J. H., and D. C. Mowery, 2003. *Intellectual Property Protection in the U.S. Software Industry*(National Academies Press, Washington, DC).

Griliches, Z., 1990. Patent Statistics as Economic Indicators: A Survey, *Journal of Economic Literature* 28, 1661-1707.

Hall, B. H., 2009. Business and Financial Methods Patents, Innovation, and Policy, *NBER Working Paper Series*.

Hall, B. H., A. B. Jaffe, and M. Trajtenberg, 2000. Market Value and Patent Citations: A First Look, *NBER Working Paper Series*.

Hall, B. H., A. B. Jaffe, and M. Trajtenberg, 2001. The NBER Patent Citations Data File: Lessons, Insights and Methodological Tools, *NBER Working Paper Series*.

Hall, B. H., A. B. Jaffe, and M. Trajtenberg, 2005. Market Value and Patent Choice, *Rand Journal of Economics* 36, 16-38.

Hall, B. H., and M. MacGarvie, 2006. The Private Value of Software Patents, *NBER Working Paper Series*.

Hall, B. H., G. Thoma, and S. Torrisi, 2007. The Market Value of Patents and R&D: Evidence from European Firms, *NBER Working Paper Series*.

Hall, B. H., and M. Trajtenberg, 2004. Uncovering GPTS With Patent Data, *NBER Working Paper Series*.

Hunt, R. M., 2006. When Do More Patents Reduce R&D?, *American Economic Review* 96, 87-91.

Klepper, S., 1996. Entry, exit, growth, and innovation over the product life cycle, *The American Economic Review* 86, 562-584.

Lanjouw, J. O., and M. Schankerman, 2004. Patent Quality and Research Productivity: Measuring Innovation with Multiple Indicators, *The Economic Journal* 114, 441-565.

Lerner, J., 2002. 150 Years of Patent Protection, *American Economic Review* 92, 221-225.

Lerner, J., and F. Zhu, 2006. What is the Impact of Software Patent Shifts? Evidence from Lotus v. Borland, *International Journal of Industrial Organization* 25, 511-528.

Mann, R. J., and T. W. Sager, 2007. Patents, Venture Capital, and Software Start-ups. *Research Policy* 36, 193-208.

Schwartz, E. S., 2003. Patents and R&D as Real Options, *NBER Working Paper Series*.

Trajtenberg, M., G. Shiff, and R. Melamed, 2006. The "Names Game": Harnessing Inventors' Patent Data for Economic Research, *NBER Working Paper Series*.