Get With the Program:

Software-Driven Innovation in Traditional Manufacturing

Lee G. Branstetter
Matej Drev
Namho Kwon

Preliminary and Incomplete

Abstract

This paper combines a review of the engineering literature, interviews with industry experts, and empirical analysis to demonstrate the increasing importance of software for successful innovation in manufacturing sectors well beyond the traditional definition of electronics and information technology. Using panel data for 231 publicly listed firms from 17 countries across four manufacturing industries over the period 1981-2005, we find significant variation across firms in the software intensity of their innovative activity. Firms that exhibit a higher level of software intensity generate more (quality-adjusted) patents per R&D dollar, and their investment in R&D is more highly valued by equity markets. We present evidence suggesting that geographic differences in the abundance of skilled software labor across countries might be an important factor in determining sample firms’ software intensity and performance.

Key Words: innovation, technological change, software innovation
I. Introduction

This paper documents the existence of a software-biased shift in the trajectory of technical innovation in a variety of manufacturing sectors beyond the traditional definition of electronics and information technology, including automobiles and auto parts, aerospace and defense, medical devices, and pharmaceuticals. Drawing on a survey of the engineering literature and interviews with industry experts, we find broad agreement among technologists that the innovation process in these industries is becoming increasingly software-intensive. Using patent and patent citation data, we present statistical evidence supporting this expert consensus.

Building on prior work by Arora, Branstetter, and Drev (2013), who used patent citations to demonstrate an increase in the software intensity of innovation in information technology, we present evidence that more recent cohorts of patented inventions in other sectors are similarly increasingly likely to cite patented software-related technologies, even after controlling for the increase in the available pool of citable software patents.

If software has indeed become an increasingly important input into the production of new knowledge and an increasingly critical driver of product differentiation in these sectors, then firms that are able to take advantage of this software-biased shift should increasingly outperform their industry peers who cannot. Our empirical analysis suggests this is exactly what has happened. Using a large unbalanced panel of the largest publicly traded firms in four industries -- automobiles and auto parts, aerospace and defense, medical devices, and pharmaceuticals -- over the period 1981-2005 we show that more software-intensive firms in these industries have started
to increasingly outperform their less software-intensive peers in terms of patent productivity (patents per R&D dollar) and the market value of R&D investment.¹

The timing of the improvement in relative innovative performance of these firms appears to be systematically linked to the rising software intensity of innovation in these sectors. We show that the relative performance of software-intensive firms improves at the same time that the software-intensity of innovation in these sectors grows. Furthermore, using a variety of robustness checks, we show that we can exclude several competing explanations for the observed empirical results.

Why is it that some firms were able to take advantage of this software-biased shift in technological change while others were not? We provide suggestive evidence similar to that presented in Arora, Branstetter, and Drev (2013) that geographic differences in the abundance of skilled software labor might have been an important factor in determining sample firms’ software intensity and innovation performance.

This paper is structured as follows. Section II reviews research from the engineering and management literatures that points to a significant increase in the importance of software as an enabler of innovation in four “traditional” manufacturing sectors. While suggestive, this research tends to be somewhat anecdotal, relying heavily on the experience of a small number of firms and a highly selected sample of recent product development efforts. Section III presents new statistical evidence based on patent citation data that suggests the software-biased shift in the

¹ This is distinct from (but complementary to) the idea that the adoption and use of IT has made firms more productive in their manufacturing and service processes. A large literature explores this the extent, persistence, and variance of this relationship across firms and countries. Brynjolfsson and Hitt (1995) and Bloom, Sadun, and Van Reenen (2012) are just a few of the important papers in this literature. For an interesting study on the impact of “data analytics” on firm productivity, see Brynjolfsson and McElheran (2015).
direction of technological change suggested by the engineering and managerial literatures is real, broad-based, and economically and statistically significant. Section IV empirically examines the implications of this shift in software intensity for the innovation performance of firms in the four manufacturing sectors that are the focus of our study. Section V discusses several possible explanations for the trends we observe in our data and ties them to the existing literature. Section VI concludes with a summary of key results and avenues for future research.

II. The Changing Technology of Technological Change in Four Manufacturing Sectors

Arora, Branstetter, and Drev (2013) and others have pointed to the increasingly important role software has played as a driver of successful innovation and product development in information technology in the recent decades. A survey of the engineering literature and industry reports in a variety of non-IT technology fields reveals a similarly pronounced increase in the importance of software for product development and innovation across a range of other manufacturing industries. In this paper, we focus on automobiles and auto parts, aerospace and defense, medical devices, and pharmaceuticals.

In the automotive industry, the amount of software in cars has been steadily rising over the past two decades, and competitive differentiation is increasingly realized through software-based capabilities (Grimm, 2003; Frischkorn, 2004; Prasad, 2004). Up to 40% of the cost of a new vehicle is now determined by its electronics and software content (Shorey, 2011). This percentage is likely to rise further; some industry observers contend that more than 70% of all innovations in the contemporary automotive sector are driven by software (Grimm, 2003;...
Shorey, 2011). Today, premium cars are equipped with up to 70-80 microprocessors, connected by 5-6 internal digital networks (Nelson, 2004), and the latest electric vehicles such as the Chevrolet Volt rely upon more than 10 million lines of computer code, easily surpassing the numbers of lines of computer code required to run Boeing's 787 Dreamliner or the new F-35 fighter (Wired, 2011).

In a modern passenger vehicle, software manages everything from its powertrain, fuel and ignition, and carbon emissions, to the car’s power antenna. As a consequence, automotive companies are increasingly investing in the internal acquisition of software capabilities through rapid hiring of software engineers (Waterman, 2011) and are building outside competencies by working closely and signing collaboration deals with software firms (Wired, 2011). Software design teams have become increasingly prominent decision-makers at the product design stage (Mustapic et al, 2004). These trends are also evident in the auto parts sector. The high degree of interest in autonomous vehicles suggests these trends have much farther to go.

A similar trend is apparent in the aerospace and defense arena. According to many industry experts, the entire aviation industry has been undergoing a process of transformation away from dependence on traditional manufacturing towards something that “looks more like IBM and Microsoft with wings” (Hughes, 1998). The Boeing 777 contains 1,280 onboard processors that use more than 4 million lines of computer code. Blackhawk helicopters contain almost 2,000 pounds of wire connecting the on-board computers and sensors, and experts claim that designing the electronic systems for this aircraft was more difficult than designing the

---

2 Interviews with an engineer employed by a leading multinational auto parts producer indicated that this firm had undertaken a major investment in software capabilities, hired thousands of software engineers, and built up research facilities in regions as diverse as Pittsburgh (PA) and India in order to tap the right skills for its increasingly software-intensive approach to product development.
aircraft itself. Many modern aircraft cannot fly without their onboard computer systems (e.g. F-16 and F-117), air traffic control systems are wholly dependent on software systems, and modern aircraft and spacecraft systems seldom work alone - they are usually part of a system of systems (Long, 2008). While aerospace products have included embedded software at least since the 1970s, when digital electronics and software first came into use for onboard engine control on commercial aircraft (Potocki de Montalk, 1993), this trend has been quickly accelerating since the 1990s (Holloway and Hayhurst, 2003). As a result, software costs are major components of product innovation and design for large aerospace companies. Boeing, for example, has significantly increased the amount of money invested in software as part of more recent product development efforts, and outlays per product generation are now in the billions of dollars (Long, 2012).

Experts agree that software has also become ubiquitous in medical devices and is the source of critical capabilities in products ranging from digital thermometers, insulin pumps, pacemakers, and cardiac monitors to anesthesia machines, large ultrasound imaging systems, MRI scanners, chemistry analyzers, and proton beam therapy systems (Sandler et al, 2010; Bakal, 2011; Jones, Jetley, and Abraham, 2010). Mai-Duc (2011) reports that more than 50% of marketed medical devices contain software. A current state-of-the-art pacemaker contains up to 80,000 lines of software code, while a simple infusion pump can contain upwards of 170,000 lines of code (Jones et al, 2010). Kahn (1991) and Holden (1986) assert that the trend of software utilization in medical devices and equipment has been in place at least since the mid-1980s when first devices with key capabilities enabled by microprocessors and controlled by embedded software came to market (Kahn, 1991; Holden, 1986). However, the software
intensity of medical devices has been accelerating particularly quickly in the last decade (Wasden, 2011).

As medical device manufacturers reposition themselves by bundling physical devices with value-added software-based features, they require an expanding array of specialized software skillsets (Joglekar and Rosenthal, 2003). As a consequence, these firms are increasingly forced to focus on software engineering and to adopt rigorous software development processes (Denger et al, 2007). This is particularly important because software failures are becoming one of the main sources of medical device recalls and litigation (Jones, Jetley, and Abraham, 2010; Mai-Duc, 2011). Firms in the medical device industry are responding both by building closer connections to external software suppliers and by hiring large numbers of software engineers, while giving software development teams a much larger stake in the product development and strategic decision-making in the industry (Bakal, 2011).

Similarly, pharmaceutical firms have also witnessed an increasing dependence on software in product development and innovation, predominantly in the form of bioinformatics and computational biology. Bioinformatics and related domains have become key tools in drug development, even if their deployment has not prevented an apparent decline in pharmaceutical research productivity (Searls, 2000). Computer models and simulations now play crucial roles in the discovery of new substances with potential therapeutic benefits. While in the early 1990s large drug discovery screening programs produced approximately 200,000 data points annually (Drews, 2000), nowadays software advances have enabled typical pharmaceutical labs to generate more than 100 gigabytes of data in a single day (Gassmann, Reepmeyer, and Von

---

3 We confirmed these trends through interviews with engineering professors who have closely followed technological trends across a range of medical device technologies.
As a consequence, pharmaceutical and biotechnology companies rely on increasingly complex algorithms and software packages to deal effectively with this proliferation of information (Duardo-Sanchez, Patlewicz, and Lopez-Diaz, 2008).

In summary, technologists and industry practitioners assert that software has become an increasingly crucial input into innovation and product differentiation across a wide array of manufacturing industries far beyond the traditional definition of electronics and information technology. In the next section, we use patent citation data drawn from these industries to support these assertions, finding evidence of statistically significant trends in the data that are consistent with the rising importance of software as an input into invention and product development.

III. Measuring the Shift in the Technology of Technological Change

Approach

If innovation in autos and auto parts, aerospace and defense, medical devices, and pharmaceuticals has increasingly come to rely on software as an input into the production of new knowledge, then we would expect this fact to be reflected in patent data. Specifically, we should observe that more recent cohorts of patents generated by these industries cite software technologies with increasing intensity, and we would expect this to be the case even after we control for the changes over time in the volume of software patents.

The use of patent citations is common in the economic and management literatures as researchers have used patent citations as a measure of knowledge flows for decades (Jaffe and Trajtenberg, 2002). Following the approach in Arora, Branstetter, and Drev (2013), which builds on the seminal work undertaken by Caballero and Jaffe (1993) and Jaffe and Trajtenberg (1996,
2002), we employ a citation function model in which we model the probability that a particular patent $p$, granted in year $t$, cites another patent, $P$, granted in year $T$.

In line with previous work, this citation probability is modeled as the product of an exponential process by which knowledge diffuses and a second exponential process by which knowledge becomes superseded by subsequent research (Arora, Branstetter, and Drev, 2013). The resulting probability, $\Pr(p,P)$, is thus a function of the attributes of the citing patent $p$ and the cited patent $P$, captured by the term $\alpha(p, P)$ below, as well as the time lag between the grant years of the two patents, $(t-T)$:

$$\Pr(p, P) = \alpha(p, P) \exp(-\beta_1(t - T)) \cdot (1 - \exp(-\beta_2(t - T)))$$ (1)

All potentially citing patents and all potentially cited patents are sorted into cells corresponding to their patent attributes. The measured attributes of the citing patents consist of the citing patent’s grant year, the primary industry of the assignee’s firm, and a binary measure of the patent’s technology field (software or non-software). The measured attributes of the cited patents consist of the cited patent’s grant year and its technology field. As a result, the expected number of citations from a group of citing patents with a particular set of attributes to a group of cited patents with a particular set of attributes can be written out as follows:

$$E(citation_{tabtc}) = n_{tab}n_{Tc}a_{tabtc} \exp(-\beta_1(t - T)) \cdot (1 - \exp(-\beta_2(t - T)))$$ (2)

where the dependent variable measures the number of citations made by patents with grant year $t$, industry $a$ and technology field $b$ to patents with grant year $T$ and technology field $c$. The alpha terms are multiplicative effects estimated relative to a benchmark or “base” group of citing and cited patents, and $n_{tab}$ and $n_{Tc}$ are the counts of patents in the respective categories. Rewriting equation (2) gives us the Jaffe – Trajtenberg (2002) and Arora - Branstetter - Drev (2013)
version of the citation function, expressing the average number of citations from one category of patents to another:

\[ P(citation_{tabTc}) = \frac{E(citation_{tabTc})}{n_{tab}n_{Tc}} = \alpha_{tabTc} \cdot \exp(-\beta_1(t - T)) \cdot (1 - \exp(-\beta_2(t - T))) \]  

(3)

If we add an error term to this expression, as in equation (4) below, then we can estimate it using a nonlinear least squares approach.

\[ P(citation_{tabTc}) = \alpha_t \cdot \alpha_a \cdot \alpha_b \cdot \alpha_T \cdot \alpha_c \cdot \exp(-\beta_1(t - T)) \cdot (1 - \exp(-\beta_2(t - T))) + \varepsilon_{tabTc} \]  

(4)

When estimating the empirical version of equation (4), we have to also adjust for heteroskedasticity by weighting the observations by the square root of the product of potentially cited patents and potentially citing patents corresponding to a particular cell, namely

\[ w = \sqrt{(n_{tab})(n_{Tc})} \]  

(5)

**Data**

In this analysis, we use utility patents granted by the United States Patent and Trademark Office (USPTO) between 1985 and 2005. To identify firms active in each of the chosen industries, we used the Compustat database and the North American Industry Classification System (NAICS). First, we selected the top 100 publicly traded firms in each industry measured by the amount of sales as identified in Compustat. Since Compustat is skewed toward North American firms, we used other data sources in order to ensure coverage of important firms outside the United States, including Amadeus, the Development Bank of Japan's Corporate Finance Database, and the U.K. R&D Scorecard.

---

4 We used sales in 2010. The firms selected using the sales in 2000 and 2005 are very similar to our target firms.
In the next step, we connected the identified firms to their U.S. patent portfolios using the updated NBER patent database.\(^5\) We only retained firms whose total number of patents between 1981 and 2005 is at least 10 in order to make sure our sample includes firms that are active producers of patented inventions.\(^6\) The U.S. patent portfolios of the retained firms constitute our set of potentially citing patents. The set of potentially cited patents is the universe of patents granted by the USPTO from 1981 through 2005.

Next, we identified software related patents, which is a perennial challenge in the empirical literature. There have been several significant efforts to define software patents. Graham and Mowery (2003) defined software patents as an intersection of those falling within a narrow range of International Patent Classification (IPC) classes and those belonging to packaged software firms. This created a sample that omitted large numbers of software patents, according to Allison et al. (2007).

The second effort was that of Bessen and Hunt (2007), who defined a software invention as one in which data processing algorithms are carried out by code either stored on a magnetic storage medium or embedded in chips. They rejected the use of official patent classification systems, and used a keyword search method instead. In this approach, the authors identified a small set of patents that adhered to their definition, and then used a machine learning algorithm to identify similar patents in the patent population, using a series of keywords in the patent title and abstract. Recently, Arora, Branstetter, and Drev (2013) used an approach that combined the Graham-Mowery and Bessen-Hunt definitions.

---

\(^5\) We were forced to exclude firms that do not appear in the NBER patent database from the sample. The updated NBER database accounts for firm mergers, acquisitions, and spin-offs through 2006, albeit with some errors and omissions discussed by Lerner and Seru (2015).

\(^6\) This accounting is based on patent grant years.
In this paper, we have taken an approach similar to that of Arora, Branstetter, and Drev (2013). First, we generated a set of patents, granted after January 1\textsuperscript{st} 1985 and before December 31\textsuperscript{st} 2005 that used a set of keywords associated with software-based technologies (e.g. “computer program” or “software”), as defined in Bessen and Hunt (2007). Second, we identified patents that fell into the narrow set of IPC categories as defined in Graham and Mowery (2003). We then defined the population of software patents for the purposes of this paper as the union of these two sets of patents. This yielded 189,134 patents, 21,755 of which were assigned to firms in our sample.

As is the case in most studies that rely on patent data, our data are potentially affected by a number of biases. It is clear that not all inventions are patented, and additional issues might be raised by changes that occurred in the patentability of software over the course of our sample period.\footnote{Graham and Mowery (2003) and Bessen and Hunt (2004) provide excellent overviews of the evolution of the patentability of software inventions in the United States.} This makes it imperative that we control for the expansion in the pool of software patents over time, which is exactly what the citation function approach allows us to do. In addition, while our analysis relies on patents generated by a single authority – the USPTO – to measure invention for both the United States and foreign firms in our sample, the foreign firms in our sample tend to be reasonably large entities with significant sales in the United States. For that reason, we can expect the foreign firms in our sample to have strong incentives to protect their inventions in the U.S. market with U.S. patents. To the extent that this assumption holds, foreign firms will patent their more important inventions in the United States, providing us with data sufficiently rich to capture important changes in their technological trajectories.
**Results**

We first look at the descriptive results presented in Figure 1, which show a stark increase over time in the fraction of citations that patents of firms in our sample industries direct toward software-related prior art. As a total, the share of citations going to software increased threefold over the sample period from 5% to 15%, with a particularly striking increase in the period following the year 2000.\footnote{The NBER Patent Database ends with patents granted in 2006. We are currently working to update our data set through 2012. Preliminary analysis based on updated patent data shows that the qualitative results identified in this paper continue to obtain in the years after 2012.} There is substantial variation across industries. Specifically, we find “aerospace and defense” to be the most software-intensive industry, followed by “automobiles and auto parts,” “medical devices,” and “pharmaceuticals.” However, these differences across industries have decreased as innovation in all of our sample industries has become increasingly reliant on software as an input into the production of new patented inventions.
Estimation results for the patent citation functions are presented in Table I. The unit of analysis is an ordered pair of citing and cited patent categories. Coefficients are reported as deviations from the baseline category – thus a positive coefficient indicates an increased citation probability relative to that category, while a negative coefficient indicates a decreased citation probability relative to the baseline category.

Our results indicate that newer cohorts of patented inventions in our sample industries are increasingly likely to cite prior patented inventions, even after controlling for increases in their volume. Importantly, we see that software patents are much more likely to be cited than non-software patents. The cited software dummy in column 1 is positive, large, and statistically significant, indicating that patents belonging to our sample firms are 21% more likely to cite software patents than non-software patents, controlling for the sizes of available software and non-software patent pools.
Estimation results reported in column 3 and 4 further solidify the point that there has been a sharp increase in the likelihood of citing software patents from 1986 to 2005.\footnote{Because our data end with the 2006 grant year, we run into difficulties associated with the truncation in our citation data. Few patents applied for in 2005 are granted by 2006. The most recent citing grant year coefficient we can estimate cleanly is for 2004.} In these specifications, we restrict the population of potentially cited patents to include only software patents. The key results are provided in column 3, which shows that a sharp, striking increase in the propensity of non-software patents generated by our sample firms to cite software prior art, even controlling for the expansion of software patents that occurred over this period. This trend emerges in the late 1990s, and accelerates through the end of our sample period, displaying a timing that is almost perfectly coincident with the rising importance of software articulated by the industry experts and engineering studies cited in the previous section. We see that a non-software patent belonging to a firm in one of the industries we study in the year 2004 is more than three times more likely to cite a software patent than a similar patent granted in the year 1986, with statistical significance at the 1 percent level. In addition, a software patent belonging to a firm in our sample granted in year 2004 is almost 3.4 times more likely to cite a software patent than a patent granted in 1986. These results align closely with the descriptive trends reported in Figure 1. We see this as strong evidence that the trajectory of technological change in our sample industries has become substantially more software-intensive.

In results not shown in the paper, we also run a variant of our citation function regressions that uses “forward” citations (citations received) instead of “backward” citations, seeking to measure how often the inventions of our sample firms are themselves cited by subsequent software inventions. We find that newer cohorts of patents belonging to firms in our
sample are increasingly likely to be cited by subsequent software inventions, suggesting that our sample firms’ R&D is increasingly embedded in and relevant to software-related technologies.

IV. Comparing Firm-Level Innovation Performance

In the previous section, we showed that there has been a software-biased shift in the nature of technical change in an array of industries, especially since the mid-1990s. Can we use this underlying trend to explain the relative innovative performance of firms in these industries? We expect that firms with a higher degree of software competence will exhibit relatively better innovative and economic performance than firms with a lower degree of software competence. If software becomes more important over time, then we should expect that the performance difference between more and less software intensive firms has grown in recent years.

In order to empirically explore these connections, we use two separate (but related) approaches: the innovation (patent) production function and the market valuation of R&D (Tobin’s Q) model.

Innovation (Patent) Production Function

The premise of this empirical approach is based on Pakes and Griliches (1984) and Hausman, Hall, and Criliches (1984). We use a log-log form of the patent production function.

\[ P_{it} = r_{it}^\beta \phi_{it} e^{\alpha SW_i} \]  

(6)

Where \( \phi_{it} = e^{\sum c_d D_c} \)  

(7)

In equation (6), \( P_{it} \) are patents taken out by firm \( i \) in period \( t \), \( r_{it} \) are research and development expenditures, \( \phi_{it} \) represent measures of innovation-sector-specific technological opportunity, and \( SW_i \) indicates if the firm is software-intensive. In equation (7), \( D_c \) represents patenting
propensity differences across \( c \) different innovation sectors. We derive our estimating equation by substituting (7) into (6) and taking logs of both sides, thus yielding:

\[
\ln(P_{it}) = \beta \ln(r_{it}) + \sum_c \delta_c D_c + \omega SW_t + \mu_{it}
\]

(8)

The error term is defined below:

\[
\mu_{it} = \xi_t + u_{it}
\]

(9)

We allow the error term in (9) to contain a firm-specific component \( \xi_t \), which accounts for the intra-industry firm-specific unobserved heterogeneity, as well as an \( iid \) random disturbance \( u_{it} \). While \( SW \) would be swept out in a linear model with firm fixed effects, because it is time invariant, we can interact \( SW \) with dummy variables corresponding to subperiods of our 1981-2005 time frame and estimate the coefficients on the interaction terms.\(^\text{10}\) If we find that the coefficients on our interaction terms are statistically significant and rising over time, this would constitute evidence that the relative performance of firms that are software-intensive throughout our sample is increasing as innovation itself has become more software-intensive. Since the dependent variable is a count variable, we use the negative binomial estimator developed by Hausman, Hall, and Griliches (1984) to estimate (8).

**Market Value (Tobin’s Q) and Shadow Value of R&D**

Since the late 1960s (Brainard and Tobin, 1968; Tobin, 1969), Tobin’s Q has been widely used to measure the relationship between a firm’s market value and the replacement value of its

\(^{10}\) Note that the fixed effects negative binomial estimation routine supplied by STATA will estimate a coefficient, even on a firm-specific variable that does not change over time. This is because the fixed effects negative binomial estimator is not exactly analogous to the linear version. See Hausman, Hall, and Griliches (1984).
book equity. The value of Tobin’s Q is affected by both a firm’s tangible capital and its intangible capital. A firm’s intangible capital (stock of knowledge measured by its R&D stock) has been found to have a positive relationship with the market value of a firm (Griliches, 1981). Following Griliches’ seminal work, hundreds of academic papers in a variety of industry and national contexts have used a firm’s R&D stock as a measure of its intangible capital in order to investigate its relationship with market value.

Following previous work, we assume an additively separable linear specification (Griliches, 1981; Hall, 1993; Hall and Kim, 2000).\footnote{Our notation follows Hall and Kim (2000).} Let $V_{it}$ and $A_{it}$ be the market value and the replacement cost of tangible assets of firm $i$ at time $t$, respectively. Then the relationship between the two variables can be written as follows:

\[
V_{it}(A, K) = q_t(A_{it} + \gamma_t K_{it})^{\sigma_t}
\]

where $K_{it}$ represents the replacement cost of the firm’s stock of knowledge, typically measured by stocks of R&D expenditures, $q_t$ represents the average market valuation coefficient of the firm’s total assets, $\gamma_t$ is the shadow value of the firm’s technological knowledge measuring the firm’s private returns to R&D, and $\sigma_t$ determines returns of scale. Following standard practice in the literature, we transform the above equation by taking natural logarithms (e.g. Hall and Oriani, 2006) as follows:

\[
\ln V_{it} = \ln q_t + \sigma_t \ln A_{it} + \sigma_t \ln \left(1 + \frac{\gamma_t K_{it}}{A_{it}}\right)
\]

By assuming $\sigma_t$ equals one (constant returns to scale) and subtracting $\ln A_{it}$ on both sides, we can obtain the following equation:
\[
\ln(\frac{V_{it}}{A_{it}}) = \ln q_t + \ln (1 + \frac{\gamma_t K_{it}}{A_{it}})
\]  

(12)

Finally, we define Tobin’s Q as the ratio of the market value to the replacement cost of tangible assets and rewrite the equation as follows:

\[
ln(Q_{it}) = \ln q_t + \ln [1 + \gamma_t \left(\frac{K_{it}}{A_{it}}\right)]
\]  

(13)

Following Hall and Kim (2000) and Arora, Branstetter, and Drev (2013), we estimate equation (13) using nonlinear least squares estimators (NLS). In order to capture the difference in the market's valuation of the private returns to R&D between firms with a higher and lower degree of software intensity, we add software intensity dummies to the model. We also include time dummies to account for a secular time trend. As a robustness check, we also estimate the equation using a linearized version of the model, with firm fixed effects, and, in these models, we can interact our time dummies with our software intensity dummies, as we did in the previous patent production function analysis.

**Data and Variables**

**Sample Firms**

The procedure used to identify our sample firms was already delineated in the previous section (see p. 10). In the regressions reported below, we were forced to drop those firms for which stock market value information was not available, as well as those which lacked sufficient

---

12 Fixed effects and random effects estimators are used. For robustness checks, we estimated a linearized version of equation (13) using firm fixed effects.
information on R&D expenditures. This yielded an unbalanced panel of 231 firms from 17 countries for patent production function analysis. For the Tobin’s Q analysis, the unbalanced panel data contains 154 firms from 14 countries. While the numbers of firms is not large, the share of global output represented by our sample firms in their respective industries is quite large. By the early 21st century, many of these industries had become increasingly consolidated, with a handful of multinational incumbents constituting a large fraction of total global sales.

**Software Intensity Variable**

We construct two software intensity variables to classify our sample firms into those who exhibit high degrees of software intensity of innovation and those who exhibit low degrees of software intensity. The first software intensity variable is based on the share of software patents in a firm’s total patent portfolio. The value of this variable equals the ratio of the number of software patents generated by the firm from 1981 to 2005 and the number of total patents generated by the firm from 1981 to 2005, where the years measure application years. The second software intensity variable is constructed using the share of citations to software patents in the total citations made by a firm’s patent portfolio. More precisely, it is the ratio of the number of backward citations to software patents made by the patents generated by the firm from 1981 to 2005 to the number of backward citations to all patents made by the patents generated by the firm from 1981 to 2005, where the years are application years.

---

13 For innovation (patent) production analysis, we drop a firm if it has less than five years of R&D flow information. Again, we exclude a firm if it has less than five years of R&D stock information for Tobin’s Q analysis. Changing the thresholds with different numbers, such as 10, does not change our results significantly. American and British firms with significant R&D budgets are generally required to disclose their R&D expenditures. Accounting practices are different in Japan, but fairly good data on R&D expenditures are nevertheless widely available for publicly traded Japanese firms. It has proven more challenging to obtain such data for firms based in continental Western Europe, since European accounting practices do not generally require disclosure of R&D expenditure.

14 We are currently working to update and extend our financial data in order to expand the coverage of our Tobin’s Q analysis.

15 Total patents include both software patents and non-software patents.
variable varies across firms but not over time. For Tobin’s Q analysis, we constructed both kinds of firm-specific software intensity variables for each of the following time periods: 1981-1988, 1989-1996, and 1997-2005.

At a first glance, it might seem counterintuitive to construct software intensity variables by averaging across time periods as opposed to simply using annual software intensity measures. However, this was necessary for several reasons. First, some firms do not report patent applications in some years. Second, firm-level software intensity measures can fluctuate significantly from year to year, especially for firms with limited patent output, but it is not reasonable to assume that these short-term fluctuations always reflect real changes in a firm’s innovation process. For example, if a firm applied for a software patent and a non-software patent in 1990, the observed share of software patents in its total annual portfolio would be 50%. Suppose, however, that in 1991 this same firm applied for two non-software patents and no software patents. Then, the observed share of software patent in that year would be 0%. Finally, if the firm applied only for a single software patent in 1992, then the share of observed software patents would be 100%. As we do not believe these annual fluctuations are necessarily reflective of an underlying drastic change in the software intensity of this firm’s innovation process, so we prefer to average software intensity measures over a longer period of time.

In order to operationalize software intensity measures, we create a binary measure that classifies sample firms into two groups: (1) the above-median software intensity group and (2) the below-median software intensity group. For Tobin’s Q regression analysis, we also construct period-specific software intensity dummy variables that allow firms to switch between the two groups.
Construction of Variables

*Patents:* Patent data were obtained from the United States Patent and Trademark Office (USPTO) and the National Bureau of Economic Research (NBER). The NBER patent database allows us to match firms with their patent portfolios through the year 2006. For those firms that were not included in the database’s firm-assignee matching correspondence, we manually matched their names to patent assignee codes. As described in detail in the section on citation function estimations, we apply the Arora, Branstetter, and Drev (2013) method in order to identify software patents.

*R&D Expenditure:* Annual R&D investment data were collected from several sources. *Compustat* provides most of the U.S. firms’ R&D data as well as data for some non-U.S. firms whose shares trade in the U.S. *The R&D Scoreboard* also contains R&D data for a number of top global R&D companies and top UK R&D firms.\(^{16}\) We exploit the *EDGAR* database to collect R&D information for some firms that are not captured in Compustat or the R&D Scoreboard.\(^{17}\) Japanese firms’ R&D data comes mainly from the *Kaisha Shiki Ho Survey* database. South Korean firms’ data is collected from the *Korea Listed Companies Association*. We deflate R&D expenditure using several alternative deflators, checking for consistency and robustness.\(^{18}\) We found our results are not sensitive to the choice of deflator.

*R&D Stock:* Following Arora, Branstetter, and Drev (2013) and others, we use the perpetual inventory method to calculate R&D stocks. A fifteen percent depreciation rate was used

\(^{16}\) The Department for Business, Innovation & Skills (BIS) of the United Kingdom has published the data from 1991 to 2010. The most recent year’s publication, the 2010 R&D scoreboard, contains global top 1000 R&D firms and UK top 1000 R&D firms.

\(^{17}\) [http://www.sec.gov/search/search.htm](http://www.sec.gov/search/search.htm)

\(^{18}\) The deflators are Consumer Price Index (CPI), GDP deflator, and Producer Price Index (PPI). This paper includes the estimation results using CPI. The estimation results using other deflators are available from authors by request.
The initial R&D stock was calculated using the previous five years’ worth of R&D expenditure flows. In order to impute R&D expenditures in years for which data were unavailable, we used a linear extrapolation based on the first five years of available R&D expenditures.

Market Value: We estimate the market value of a firm by following the method proposed by Perfect and Wiles (1994). We define the market value as the sum of market values of the firm’s equity and debt. For the firms whose data is taken from Compustat, we estimated the market value of the firm’s equity as the sum of (1) year-close price of outstanding common shares multiplied by the year-close number of outstanding common shares and (2) year-close liquidating value of preferred capital. For the Japanese firms from the Development Bank of Japan (DBJ) database, we calculated the market value of the firm’s equity as the mean value of year-high and year-low stock prices multiplied by the number of outstanding stocks. The value of preferred capital was not available in DBJ database. This, however, should not cause a problem if the values of preferred capital are not systematically different across time and technology sectors (Arora, Branstetter, and Drev, 2013). We define the market value of the firm’s debt to be equal to the sum of long-term debt and short-term debt. For the firms from Compustat, we used total long-term debt and debt in current liabilities. For Japanese firms from the DBJ database, we used fixed liabilities as a proxy for the value of the firm’s debt.

19 Different depreciation rates between 10% and 30% were applied for constructing the R&D stock. This paper reports the estimation results using 15% depreciation. Applying the different rates did not alter our results significantly. The estimation results using other depreciation rates are available from authors by request.

20 For example, R&D stock in 1990 is the sum of the R&D expenditure in 1990 and depreciated R&D expenditures from 1986 to 1989.

21 For instance, assuming that R&D expenditure in 1980 is missing, we get the projected R&D expenditure in 1980 by “backcasting,” using R&D expenditure data from 1981 to 1985.

22 See Perfect and Wiles (1994) for a detailed discussion of measurement error issues when using book values.
Replacement Value of the Firm’s Assets: It is not easy to estimate the replacement value of a firm’s assets mainly because there is often no structured and active market for used capital goods. However, Perfect and Wiles (1994) show that replacement values calculated using different methods are relatively robust. In this paper, we use the book value of a firm’s total assets as a proxy for their replacement value.

Innovation Production Function Results

In the previous section, we showed that patented inventions in an array of manufacturing industries increasingly rely on software-related prior art, even after controlling for the increased pool of citable software patents over time. Now we go a step further and investigate how firm-level innovation productivity is determined by a firm’s software competence (intensity). We first look at descriptive statistics. Figure 2 plots the average share of software patents by each sample industry.\(^23\) As expected, this software intensity metric has increased considerably in all four industries. The aerospace and defense industry seems to be the most software-intensive industry followed by medical devices, automobiles and auto parts, and pharmaceuticals.\(^24\)

\(^{23}\) The share is calculated by dividing the number of software patents by the number of all (software and non-software) patents.

\(^{24}\) Figure 2’s results closely align with those presented in Figure 1.
Figure 3 further illustrates trends in the average share of software patents for the two groups of firms with different software intensities. The two lines on the top and the bottom represent the above-median software intensity firms and the below-median software intensity firms, respectively. On average, the growth rate of the number of software patents is much higher than that of non-software patents. At the beginning of our sample period, in the 1980s, above-median software intensity firms exhibited an 8% share of software patents in their total patent portfolio. This share rose dramatically and reached 20% in the late 1990s and early 2000s. Interestingly, we see evidence of a software-biased technology shift even when we look at firms with low levels of average software intensity. Below-median software intensity firms increased their share of software patents from effectively 0% to approximately 5% of their total patent
portfolios. However, the software intensity gap between above- and below-median firms has widened, especially since the mid 1990s.

**Figure 3: Share of Software Patents by Software Intensity of the Firm**

![](image)

The average number of total patents per firm is shown in Figure 4. We see that the firms with a higher degree of software intensity produce more patents than those with a lower degree of software intensity. This difference in patent production exhibited a substantial increase from the mid of 1990s onward and accelerated in the late 1990s and early 2000s.
However, this descriptive analysis can only take us so far as many factors, such as differences in R&D investment, could have influenced the observed changes in inventive output. Instead, we want to see whether firms with a higher degree of software intensity produce more patented inventions per dollar of R&D than firms with a lower degree of software intensity. Furthermore, in order to make sure differences in the quality of patented inventions between the two groups of firms are not driving our results, we also control for patent quality. We follow the literature and use quality corrections based on the number of claims found in a patent document and the number of forward citations that a patent receives.

Table II presents our first set of key patent production function estimation results, which align closely with the picture sketched out by our descriptive statistics. The magnitudes of the
key coefficients in column (2) of Table II are graphically represented in Figure 5, where the bars represent how much the above-median software intensity firms increased their innovative productivity relative to the below-median software intensity firms in each period, relative to the base period of the early 1980s (1981-1985). We observe an increasing R&D productivity gap over time in favor of more software intensive firms across all industries in our sample. Consistent with the notion that the importance of software as an input into the creation of new technology has increased in recent years, we observe the most significant relative patent productivity gains by above-median software intensity firms in the last two periods in our sample (1996-2000 and 2001-2005). These firms became 14% and 22% more productive than their below-median software intensity peers in the late 1990s and early 2000s. It would be reasonable to expect that the R&D productivity gap between highly software intensive firms and their less software intensive peers has continued to widen over the past decade, and preliminary analysis with updated data appears to confirm this hypothesis.
The results reported in Table II were estimated using a negative binomial model, though our results are robust to the exact choice of specification. The first and second columns report regression results obtained where the total number of patents applied for by firm $i$ in year $t$ is the dependent variable. Columns 3-4 report regression results using the number of claims within firm $i$’s cohort of patents applied for in year $t$ as the dependent variable. Columns 5-8 report results using the number of forward citations received by firm $i$’s cohort of patents applied for in year $t$ as the dependent variable. Columns 7-8 report results obtained when our regressions are run on an updated citation data set extended through 2010. The R&D productivity coefficients are very similar across the columns. They provide evidence that highly software intensive firms

---

25 Use of a Poisson regression model yields similar coefficients.
started to produce more patents per R&D dollar than less software intensive firms over our sample period, without sacrificing the quality of their patent portfolios.\textsuperscript{26} Most of the coefficients on our key variables are statistically significant at the 5\% level. Furthermore, most of the coefficients in the recent time periods including 1996-2000 and 2001-2005 are statistically significant at the 1\% level. Random effects and fixed effects models produce similar estimates. This suggests that our regression results are unlikely to be driven by time-invariant unobserved firm-specific differences in research productivity or propensity to patent. We conducted separate regressions using two subsamples; U.S. firms and Non-U.S. firms. The results from both regressions are qualitatively the same with the results in Table II. We re-estimated the regressions reported in Table II, where software intensity was measured by the share of patent citations made to software prior art rather than the share of software patents. The results obtained were qualitatively similar, showing a statistically significant increase in patent productivity in the later periods. Finally, we allowed both measures of software intensity to vary within firms over time and re-estimated our specifications, again obtaining results showing that the patent intensive firms become significantly more productive, and this result strengthens over time. All of these additional results are available from the authors upon request.

As an additional robustness check, we also conducted a series of falsification estimations in which we replaced our firm-level metric of software intensity with alternative firm-level characteristics that one could possibly expect would be driving our results. We report the results of one such exercise in Table V, in which we estimated our base patent production function specification, but replaced software intensity with a measure of firm size. The key variable does not show any significant results, suggesting that firm size does not impact the patent productivity

\textsuperscript{26} We regard the number of claims and the number of citations as proxies of the quality of the patent.
of our sample firms. This is significant as it invalidates a key alternative explanation for our results – namely, that larger firms are both more productive in their inventive activities and more software intensive.

**Private Returns to R&D**

While we have already shown that more software intensive firms exhibit increasingly higher R&D productivity as measured by production of patented inventions than their less software intensive peers, we would also like to investigate whether the R&D investment of these firms receives a higher valuation from equity market investors than the R&D investments of their less software intensive peers. Tobin’s Q regressions allow us to do just that: establish how the software intensity of a firm is associated with the equity market's valuation of the private returns to its R&D investment. Table III reports estimation results for our base Tobin’s Q specification shown in equation (13) using nonlinear least squares (NLS) estimators. Figure 6 graphically depicts the average difference in the estimated private returns to R&D between above- and below-median software intensity firms. In Table III and Figure 6, software intensity is inferred from the share of software patents in total patents.

---

27 It is calculated as the difference between the below-median software intensity group subtracted from above-median software intensity group.
Figure 6 shows that above-median software intensity firms exhibit a higher estimated return to R&D investment and that this trend has accelerated in more recent time periods. At the beginning of our sample period (1981-1988), the estimated (private) return to R&D investment for above-median software intensity firms was not materially different from that of below-median software intensity firms. The effects of a software biased technology shift thus did not become apparent until the mid-1990s. Above-median software intensity firms in this period (1989-1996) started exhibiting higher estimated returns to R&D investment than their below-median software intensity peers. The difference exploded in the most recent period (1997-2005). This trend, which is strikingly similar to that reported by innovation production function

---

28 Time periods are somewhat different from the patent production function analysis. We add more years for each period because of the limited number of observations.
estimations, shows that firms which started producing more software-intensive inventions have become increasingly rewarded by stock market investors with higher market valuations.

The results of Tobin’s Q estimations are robust to a variety of robustness checks. For example, we estimated a linearized version of equation (13) using ordinary least squares (OLS) with firm-level fixed effects, and found the results using OLS/FE to be qualitatively robust. This is reported in Table IV. The trends in measured private returns to R&D for above-median software intensity firms relative to below-median firms were qualitatively similar to those obtained from the NLS specifications. We also replicated Table III and Table IV by running regressions using two subsamples; U.S. firms and Non-U.S. firms. The results from both regressions are qualitatively the same as the results reported in Table III and Table IV. Finally, we reran the regressions in Tables III and IV, measuring software intensity with the share of patent citations directed to software prior art. The results are quite similar to those show in the paper. The results of all of these robustness checks are available from authors by request.

V. Discussion

Two key facts can be derived from our analysis. First, there exists robust empirical evidence indicating the growing importance of software-related technologies and skills for successful innovation in fields far beyond the traditional borders of information technology and electronics. Software is increasingly central to innovation in automobiles and auto parts, aerospace and defense, medical devices, and pharmaceuticals. Secondly, firms in these industries which rely less on software in their R&D activity are increasingly being outperformed by their more software-intensive peers in terms of their innovative activity, as measured by patents and by the stock market’s valuation of R&D investment. To the best of our knowledge, this is the first paper that documents the extent and pervasiveness of this shift.
However, these observations lead to an obvious question: if software is so important for successful innovation in the more traditional manufacturing sectors that are the subject of our study, then why are not all firms exploiting it in equal measure? One obvious response is that low R&D productivity and low software intensity are both consequences of managerial failure. Firms with progressive managers recognize the opportunity presented by the rising importance of software and create capabilities within the firm that allow it to exploit this opportunity. Firms with less adept managers neither recognize the opportunity, nor build the capabilities necessary to exploit it. A stream of the recent management literature has focused on how managerial mindsets, formed through years of inexperience, affect the (in)ability of firms to make strategic shifts when firm environments change (Bettis and Hitt, 1995). In the economics literature, Nick Bloom, John Van Reenen, and their coauthors have shown that persistent performance differences across firms based in different countries could be driven by differences in management practices (Bloom et al., 2012; Bloom and Van Reenen, 2007, 2010). The papers also show that multinationals tend to bring their management practices, both good and bad, with them when they set up subsidiaries abroad. Cole (2006) and Cole and Fushimi (2011) argue that the striking international decline of Japan’s once formidable IT industry stems from managerial failure – the “hardware-centric” managers of Japan’s IT firms simply could not recognize the software-biased shift in technological opportunity in IT, nor adapt to it.

The prior work of Arora, Branstetter, and Drev (2013) suggests an alternative explanation for the relatively poor performance of Japan’s IT industry that is rooted in resource constraints. These authors used statistics on university graduates by discipline and immigration by occupation to create a rough statistical portrait of the human resource pool available for employment in software and related disciplines for the U.S. and Japan. They show that Japan
consistently lagged the U.S. in terms of human resources in this domain, and that the gap between the two countries widened enormously in the mid-to-late 1990s as global demand for this specialized human capital intensified. This dramatic widening of the human resource gap was driven mostly by differences in immigration, especially the entry into the U.S. labor market by Indian-born software professionals under the auspices of America’s H1-B visa program.29 Data on software “offshoring” by U.S.-based and Japan-based multinationals is less comprehensive, but any consideration of offshoring would only widen the implied human resource gap.

The current paper features data on firms based in a wider range of countries, but four of most significant home bases for our sample firms are Japan, Germany, the United Kingdom, and France, in addition to the United States. Using data from the national statistical agencies on university graduates by discipline and the immigration of IT professionals, we measure the software engineering labor pools in a manner similar to that used by Arora, Branstetter, and Drev. Some of our results are provided in Figure 7, which measures the implied “flows” of IT workers in these four countries and the United States. What is immediately apparent is that the U.S. has a sizable human resource advantage in this domain, and it widens considerably over time. Immigration into the U.S., especially from India, plays an important role in enlarging and maintaining this advantage, even in more recent years, when a statutory “quota” has limited the number of H1-B visas issued. Any consideration of software offshoring expands the gap even more, and any reasonable estimate of the “stock” of software engineers implied by these flows paints an even more overwhelming picture of American dominance.30

---


30 A brief discussion of the multiple sources of these data are provided in a supplementary data appendix.
This suggests that firms headquartered in the United States have a “built-in” advantage in software-centric research. This is significant, because when we examine which firms in our sample are in the top quartile in terms of measured software intensity, these firms are disproportionately American, and that is true across all four sectors that are the target of our current study. Foreign firms rising into the top quartile are generally large multinationals. In a moment, we will present evidence suggesting that foreign firms use their U.S.-based research labs (and to some extent, India-based research labs) to exploit local abundance in software talent.
The existence of a human resource gap in favor of U.S.-based firms is not surprising. The U.S. has held a lead in software since the early days of computing. The leading schools of computer science are all located in the U.S., and America, through its H1-B visa program and strong historical ties to centers of Indian software activity, has been able to attract large numbers of foreign software workers to the U.S., even in periods when the global demand for professionals with these skills exceeded the supply in every country. More recently, U.S. multinationals have set up large software engineering centers in India, allowing them to tap this talent without relocating the workers. Multinationals based in other countries have followed suit, but with a lag, and they generally encounter greater cultural barriers.\(^\text{31}\) The managerial literature highlights the challenges that arise when firms seek to do strategically significant R&D abroad (Anchordoguy, 2000).\(^\text{32}\)

Of course, not all the software engineers who graduate or immigrate into these various national labor markets are employed in new product development, nor are all of these engineers capable of paradigm-shattering innovation. The point we are making is that a larger resource pool can ease the constraints on the productivity of the top tier of software engineering talent. Large software engineering projects are labor-intensive, and tend to require a "pyramid" of software engineering talent, with very highly trained software architects at the apex of the pyramid, and large numbers of more narrowly trained programmers at the lower levels. In the sense that trade economists use the term, the U.S. is relatively abundantly endowed in nearly all tiers of software engineering talent, relative to the other major industrial economies. We posit

---

\(^{31}\) Language barriers can also play a role in hiring foreign software engineers. These issues appear to constrain the ability of firms headquartered in some European countries (Germany, Norway and the Netherlands) to recruit highly skilled foreign workers (McLaughlan and Salt, 2002).

that firms around the world are seeking to become more software-intensive, and that firms in the U.S. face lower barriers in doing so. The highly uneven geographic distribution of key human resources helps generate the differences in software intensity across firms captured in our data. From the perspective of our sample firms, these differences are at least partly exogenous. This line of reasoning suggests an empirical test which can help us distinguish between an explanation of our results based on managerial failure and one based on geographically variant resource constraints.

If we believed our results were primarily driven by cross-firm (but geography-independent) differences in firms’ ability to identify and take advantage of the software-biased technology shift, then we would expect to find that firms vary in how software intensive their inventions are, but we would not necessarily expect to find large differences in the software intensity of R&D conducted by the same firm in different geographic regions. If, however, we believe that geographic differences in the abundance of affordable skilled software labor have been a major factor producing variation in software-intensity across firms, then we would expect to find that firms strategically allocate software intensive inventive activities to those regions where skilled software labor is most abundant.

Figure 8 below presents the results of such an exercise, where we use sample firms’ U.S. utility patents drawn from the years 1981 through 2005. Pooling across all of our sample industries, we find stark differences in software intensity of patented inventions across regions. While U.S. firms in our sample conduct significantly more software intensive innovation at home than abroad, the exact opposite is true for Japanese and European firms. When European firms invent at home, for example, the share of software patents in their patent portfolios is only about 6%. However, when these same firms conduct innovation abroad, which is primarily in the
United States, this share rises to about 19%, even surpassing the share of software patents in the patent portfolios of US firms inventing at home (13%). Further disaggregation of the data by industry and location of foreign R&D supports the view that local human resource abundance has a significant impact on the software intensity of multinational R&D. We get a similar picture if we measure software intensity by patent citations to software prior art.

**Figure 8: Software Intensity of Patented Inventions (Share of Software Patents), by Geography of Invention and Country of Ownership - US, EU and Japan**

![Graph showing software intensity of inventions by country and geography](image)

*Note: In this figure, the different shades denote patents assigned to MNCs headquartered in the U.S., Japan, and the EU, respectively. The first three columns show the relative software intensity of inventions taken out by U.S., Japanese, and EU firms where the inventor location is in the U.S. The next three columns denote inventor location in Japan. The last set of columns denote inventor location in the EU.*

We close this section with an anecdote from our own hometown. In recent years, the German auto parts giant Bosch has set up a research facility in the Pittsburgh area – principally motivated by the desire to tap into Carnegie Mellon’s software engineering expertise. In personal interviews with some of the management of this facility, we learned that Bosch’s Pittsburgh research facility is just one small part of a major effort by the firm to acquire the software engineering capability that it feels will be essential to its continued competitiveness in
auto parts and components. Bosch has set up another software-focused research facility near Stanford and has a major development center in India. Back in the early 1990s, according to our source, the “fuel injection” business unit employed about 7,000 R&D personnel worldwide, of whom only about 150 (2.1%) were software engineers. By the mid-2000s, total global R&D personnel had risen to 11,000, and about 5,000 of these were software engineers (45%). By 2011, Bosch’s fuel injection unit employed 4,500 software engineers in India alone. Bosch did not let its German home base prevent it from acquiring the necessary capabilities, but it had to venture quite far from that home base in order to do so, hiring a nontrivial number of U.S.-based and India-based researchers in the process.

Of course, software is not the only important capability required for successful product development in auto parts or in any of the other industries we examine in this paper, and the proximity of American firms to the world’s best software engineering labor pool does not guarantee the success of individual American producers. The struggles of the American auto industry in adapting their relatively fuel-inefficient product line to the oil shock of the mid-2000s provides a useful counterexample. In addition, our analysis focuses on innovation and new product development – it says nothing about the prospects for the United States as a manufacturing location or its prospects as an exporter of manufactured goods. Nevertheless, other things being equal, America’s relative abundance in software engineering, which was achieved and maintained, in large part, due to a relatively open immigration regime, has been a source of advantage for U.S.-based firms, and it has also served as a magnet for FDI by knowledge-intensive foreign firms. The trends visible in our preliminary analysis of data through 2012 suggest that this advantage will become more important, not less, in the foreseeable future.
VI. Conclusions, Implications, and Next Steps

This paper documents the existence of a software-biased shift in the direction and nature of technological change across a range of manufacturing industries far beyond the traditional boundaries of information technology and electronics. An emerging research stream in the engineering and product development literatures suggests the existence of this shift in automobiles and auto parts, aerospace and defense, medical devices, and pharmaceuticals, but much of this evidence is anecdotal, based on comparisons of small and possibly unrepresentative samples of recent products and components. Drawing upon standard patent citation analysis methods, a broad sample of important firms in these industries, and comprehensive data on the U.S. patent grants awarded to these firms, we find strong statistical evidence for the growing importance of software-related technologies for successful innovation in this diverse array of non-IT sectors. To the best of our knowledge, this is the first paper in the economics literature that provides detailed empirical evidence for the existence of this important technology trend.

Next, using a panel of the largest publicly traded firms in these industries in the period from the early 1980s to the mid-2000s, we show that firms which draw more upon software-related technologies in their inventive activity are increasingly outperforming their less-software intensive peers. This widening gap is evident both when we investigate the average patent productivity of R&D and when we examine equity market investors' valuations of the firms' R&D investments.

Finally, our paper explores the connection between the measured software intensity of our sample firms and the relative availability of specialized human resources in different national labor markets. Firms in the highest quartile of measured software intensity are disproportionately American. Using publicly available data on university graduates by discipline
and immigrants by occupational category, we document large, persistent, and growing differences in the availability of software engineering human resources across the economies that are the most important home markets of our sample firms. The U.S., which has always been relatively abundant in software related human resources, has significantly expanded its advantage over our sample period. The presence within our data set of firms conducting R&D in multiple countries allows us to further explore the connection between local software engineering human resource abundance and the nature and direction of multinational R&D. We find that U.S. multinationals do significantly less software-intensive R&D in Europe and Japan than they do at home, whereas the opposite obtains for European and Japanese multinationals. This provides additional evidence for the notion that differences in measured software intensity are at least partly driven by labor market constraints. However, more work is certainly needed to fully determine the causal mechanisms underlying our results.

Taken together, our results may provide some interesting implications for the literature on the economics of innovation, for managers, and for policymakers. First, the nature of technological change has shifted in ways that the economics literature -- and perhaps some managers -- have not yet recognized. While a full assessment of the active and sometimes acrimonious debate over the appropriateness of software patents is beyond the scope of this paper, our results suggest that patented software technology is an increasingly central input into the creation of new products across a wide range of industries. The growing centrality of software may suggest the need for caution in any movement to narrow or restrict the ability of software inventions to benefit from patent protection, much less any movement to abolish software patents altogether. Second, the rise of software as an innovation enabler across the manufacturing space raises the salience of the highly skewed distribution of software human
resources across national labor markets and highlights the importance of high levels of in-
migration of software engineers into the U.S. in maintaining the competitiveness of U.S. firms in
innovation and new product development. Arora, Branstetter, and Drev (2013) suggested that
this was a key factor in driving the competitive resurgence of the U.S. IT industry (and the
striking competitive decline of the Japanese IT industry) in recent years. The current paper’s
results suggest that the impact of software extends much farther, into industries typically thought
of as rather distant from IT. Ending legislative barriers that currently prevent even higher levels
of in-migration of foreign software engineers would likely have benefits that extend far beyond
the boundaries of the conventionally defined IT industries.

Ongoing research efforts seek to expand our data set in breadth and time. We are
currently updating our patent data to include patents granted through 2012. Preliminary analyses
suggests that the trends documented herein have continued (and strengthened) in more recent
years, but further confirmation must await a careful accounting of the mergers, acquisitions,
divestitures, and new entry that has occurred in our sample industries. Our econometric
approach requires data on R&D expenditure, which has proved challenging to obtain for firms
located outside the U.S. and the U.K., where current accounting standards require disclosure of
“material” levels of R&D expenditure. We are continuing our efforts to expand the set of firms
for which we have reasonably high quality R&D data, and we are also expanding the set of firms
for which we have the full set of financial variables required for the calculation of Tobin’s Q.
Finally, it is apparent that the general trend towards more software-intensive innovation extends
far beyond the industries we have yet studied, and we are currently investigating the possibility
of extending our analysis to additional industries. As is always the case in economics, more
work remains to be done.
References


Long, Lyle N., "Introduction to Software Engineering" (Course Syllabus), Department of Aerospace Engineering at Pennsylvania State University, Spring 2012


Data Appendix: ICT Education and Immigration Flows

The data which describe available flows and stocks of software-skilled human capital comprise of three components: (a) annual graduates with software-related degrees, (b) annual immigration flows of immigrants in software-related occupations, and (c) annual assignment of software-related offshore talent from India. The availability and degree of official disaggregation of these data vary significantly among countries and time periods. It was therefore necessary for us to make several assumptions about the structure and evolution of various data series in order to be able to construct longitudinal time series for all of the countries included in our analysis (i.e. USA, UK, Germany, France, and Japan). This Appendix describes the data construction process for each country in some detail.

The United States: Data on cohorts of software-related graduates come from NSF’s annual Survey of Earned Doctorates and the biannual National Survey of Recent College Graduates. Due to the biannual nature of the latter survey, we used linear interpolation in order to impute numbers for missing years. The annual data on flows of immigrants in software-related occupation come from USCIS annual reports on “Characteristics of Specialty Occupation Workers (H-1B)”. Data for years prior to 1999 come from a previous paper written by the authors. Data on available offshore software-skilled labor in India come from NASSCOM’s annual reports, which contain a breakdown of Indian BPO industry’s exports by destination.

The United Kingdom: Data on annual cohorts of software-related graduates come from statistical tables published by UK’s Higher Education Statistics Agency. Data on annual flows of migrants in software-related occupations come from Prof. John Salt’s periodic reports on the state of immigration in the United Kingdom. Linear interpolation was used to impute immigration flows data for years for which official statistics were not available. Data on
available offshore software-skilled labor in India come from NASSCOM’s annual reports, which contain a breakdown of Indian BPO industry’s exports by destination.

**Germany:** Data on annual cohorts of software-related graduates come from the Federal Statistical Office of Germany. Immigration data come from annual reports published by Germany’s Federal Office for Migration and Refugees. Data on available offshore software-skilled labor in India come from NASSCOM’s annual reports, which contain a breakdown of Indian BPO industry’s exports by destination.

**France:** Data on annual cohorts of software-related graduates come from OECD’s statistical tables on “tertiary-type A” education. Due to the unavailability of data for years before 1999 and after 2010, linear extrapolation was used to populate the missing data fields. Data on annual flows of immigrants come from various reports published by the French Ministry of the Interior. Due to sparse information on immigration for some years in the sample, linear extrapolation and interpolation were used to impute immigration flows data for years for which official statistics were not available. Data on available offshore software-skilled labor in India come from NASSCOM’s annual reports, which contain a breakdown of Indian BPO industry’s exports by destination.

**Japan:** Data on graduates are taken from the Japanese Ministry of Education, Sports, and Welfare’s Basic School Survey. Data on newly registered foreign workers come from the Annual Report of Statistics on Legal Migrants published by the Japanese Ministry of Justice. Data on available offshore software-skilled labor in India come from NASSCOM’s annual reports, which contain a breakdown of Indian BPO industry’s exports by destination.
From Flows to Stocks: In order to calculate stocks of available software-related labor for the above countries at various points in time, we employed a perpetual inventory method. We made the following assumptions in order to be able to convert flow measures into stocks: (1) flow of new graduates in the 10 years prior to 1995 equaled that in the year 1995; (2) 80% of new graduates go into software-related employment every year; (3) every year 10% of existing software-related employees move out of software-related employment for good; (4) 10% of existing software-related immigrants leave the country and/or software-related employment each year; (5) 40% of total H1-B issued in a year are new petitions for initial employment; (6) flow of new ICT immigrants in the 10 years prior to 1995 equaled that in the year 1995.
Table I: Citation Function Results

<table>
<thead>
<tr>
<th></th>
<th>Full Model</th>
<th>Citations from NSW to SW</th>
<th>Citations from SW to SW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Column 1</td>
<td>Column 2</td>
<td>Column 3</td>
</tr>
<tr>
<td></td>
<td>Coefficient</td>
<td>Std.Error</td>
<td>Coefficient</td>
</tr>
<tr>
<td>Citing Grant Year</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1987</td>
<td>0.187</td>
<td>0.281</td>
<td>0.658*</td>
</tr>
<tr>
<td>1988</td>
<td>0.147</td>
<td>0.249</td>
<td>0.295</td>
</tr>
<tr>
<td>1989</td>
<td>0.248</td>
<td>0.249</td>
<td>0.513*</td>
</tr>
<tr>
<td>1990</td>
<td>0.288</td>
<td>0.240</td>
<td>0.748***</td>
</tr>
<tr>
<td>1991</td>
<td>0.214</td>
<td>0.213</td>
<td>0.534**</td>
</tr>
<tr>
<td>1992</td>
<td>0.289</td>
<td>0.210</td>
<td>0.578***</td>
</tr>
<tr>
<td>1993</td>
<td>0.385</td>
<td>0.209</td>
<td>0.518**</td>
</tr>
<tr>
<td>1994</td>
<td>0.551*</td>
<td>0.215</td>
<td>0.662***</td>
</tr>
<tr>
<td>1995</td>
<td>0.831***</td>
<td>0.231</td>
<td>0.972***</td>
</tr>
<tr>
<td>1996</td>
<td>0.996***</td>
<td>0.229</td>
<td>1.225***</td>
</tr>
<tr>
<td>1997</td>
<td>1.143***</td>
<td>0.221</td>
<td>1.435***</td>
</tr>
<tr>
<td>1998</td>
<td>1.310***</td>
<td>0.212</td>
<td>1.384***</td>
</tr>
<tr>
<td>1999</td>
<td>1.583***</td>
<td>0.208</td>
<td>1.505***</td>
</tr>
<tr>
<td>2000</td>
<td>2.380***</td>
<td>0.230</td>
<td>2.016***</td>
</tr>
<tr>
<td>2001</td>
<td>2.834***</td>
<td>0.222</td>
<td>2.659***</td>
</tr>
<tr>
<td>2002</td>
<td>3.308***</td>
<td>0.209</td>
<td>3.146***</td>
</tr>
<tr>
<td>2003</td>
<td>4.093***</td>
<td>0.203</td>
<td>3.694***</td>
</tr>
<tr>
<td>2004</td>
<td>5.128***</td>
<td>0.209</td>
<td>5.109***</td>
</tr>
<tr>
<td>Cited Grant Year</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1986</td>
<td>-0.0473</td>
<td>0.048</td>
<td>-0.102**</td>
</tr>
<tr>
<td>1987</td>
<td>-0.0683</td>
<td>0.049</td>
<td>-0.120***</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2003</td>
<td>-0.933***</td>
<td>0.018</td>
<td>-0.969***</td>
</tr>
<tr>
<td>2004</td>
<td>-0.956***</td>
<td>0.021</td>
<td>-0.982***</td>
</tr>
</tbody>
</table>

Citing patent: Firm industry

| Aerospace and Defense | 0.248*** | 0.030 | 0.115*** | 0.020 | 0.0916* | 0.037 | 0.0882** | 0.029 |
| Medical Devices       | 1.322*** | 0.047 | 0.476*** | 0.026 | 0.806*** | 0.052 | 0.235*** | 0.035 |
| Pharmaceutical        | -0.128***| 0.025 | -0.412***| 0.018 | -0.504***| 0.028 | -0.516***| 0.026 |

Software Patent

| Citing from Software Patent | -0.156*** | 0.031 |
| Cited Software Patent      | 0.209***  | 0.020 | -0.252***| 0.034 |
| Citing from Software Patent X | 5.870*** | 0.106 |
| Cited Software Patent      | 5.870*** | 0.106 |

| Obsolescence | 0.297*** | 0.012 | 0.340*** | 0.008 | 0.310*** | 0.013 | 0.340*** | 0.012 |
| Diffusion     | 4.35E-6***| 1.03E-06| 8.29E-6***| 1.34E-06| 6.15E-6***| 1.50E-06| 6.89E-5***| 1.57E-05|

Adj R-Squared           | 0.913 | 0.859 | 0.893 | 0.908 |
Number of Obs           | 1680  | 3360  | 840  | 840  |

The data for regression estimations presented in this table are drawn from the CASSIS patent database maintained by the United States Patent and Trademark Office and from the NBER Patent Data Project database. Regression specifications are estimated in STATA using the nonlinear least squares algorithm. The dependent variable is an empirical measure of the probability a citing patent with given attributes cites a cited patent with a particular set of attributes. All presented coefficients are relative to base categories, which are the following: citing patent grant year = 1986, cited patent grant year = 1985, citing firm industry = “Automobiles.” The rest of the base categories are model specific.
<table>
<thead>
<tr>
<th>Dependent Variable</th>
<th>Number of Patents</th>
<th>Number of Claims</th>
<th>Number of Citations (81-06)</th>
<th>Number of Citations (81-10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log R&amp;D</td>
<td>0.0851***</td>
<td>0.0582***</td>
<td>0.183***</td>
<td>0.165***</td>
</tr>
<tr>
<td></td>
<td>(0.0132)</td>
<td>(0.0134)</td>
<td>(0.0107)</td>
<td>(0.0110)</td>
</tr>
<tr>
<td>Software Intensity Dummy</td>
<td>-0.315***</td>
<td>-0.302***</td>
<td>-0.245**</td>
<td>-0.255**</td>
</tr>
<tr>
<td></td>
<td>(0.114)</td>
<td>(0.116)</td>
<td>(0.119)</td>
<td>(0.120)</td>
</tr>
<tr>
<td>Software Intensity Dummy * 1986-1990</td>
<td>0.306**</td>
<td>0.320**</td>
<td>0.366**</td>
<td>0.379***</td>
</tr>
<tr>
<td></td>
<td>(0.128)</td>
<td>(0.129)</td>
<td>(0.143)</td>
<td>(0.143)</td>
</tr>
<tr>
<td>Software Intensity Dummy * 1991-1995</td>
<td>0.350***</td>
<td>0.369***</td>
<td>0.268**</td>
<td>0.274**</td>
</tr>
<tr>
<td></td>
<td>(0.122)</td>
<td>(0.123)</td>
<td>(0.135)</td>
<td>(0.135)</td>
</tr>
<tr>
<td>Software Intensity Dummy * 1996-2000</td>
<td>0.431***</td>
<td>0.446***</td>
<td>0.374***</td>
<td>0.374***</td>
</tr>
<tr>
<td></td>
<td>(0.116)</td>
<td>(0.117)</td>
<td>(0.129)</td>
<td>(0.129)</td>
</tr>
<tr>
<td>Software Intensity Dummy * 2001-2005</td>
<td>0.506***</td>
<td>0.524***</td>
<td>0.450***</td>
<td>0.461***</td>
</tr>
<tr>
<td></td>
<td>(0.121)</td>
<td>(0.121)</td>
<td>(0.132)</td>
<td>(0.133)</td>
</tr>
<tr>
<td></td>
<td>-0.116</td>
<td>-0.104</td>
<td>-0.197*</td>
<td>-0.191*</td>
</tr>
<tr>
<td></td>
<td>(0.0989)</td>
<td>(0.0995)</td>
<td>(0.112)</td>
<td>(0.112)</td>
</tr>
<tr>
<td></td>
<td>-0.00353</td>
<td>0.0192</td>
<td>-0.00211</td>
<td>0.0169</td>
</tr>
<tr>
<td></td>
<td>(0.0941)</td>
<td>(0.0946)</td>
<td>(0.105)</td>
<td>(0.105)</td>
</tr>
<tr>
<td></td>
<td>0.375***</td>
<td>0.410***</td>
<td>0.462***</td>
<td>0.494***</td>
</tr>
<tr>
<td></td>
<td>(0.0999)</td>
<td>(0.0915)</td>
<td>(0.101)</td>
<td>(0.101)</td>
</tr>
<tr>
<td></td>
<td>-0.240**</td>
<td>-0.197**</td>
<td>-0.269***</td>
<td>-0.245**</td>
</tr>
<tr>
<td></td>
<td>(0.0956)</td>
<td>(0.0963)</td>
<td>(0.104)</td>
<td>(0.104)</td>
</tr>
<tr>
<td>Industry Dummies</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Number of Obs</td>
<td>3935</td>
<td>3924</td>
<td>3935</td>
<td>3924</td>
</tr>
</tbody>
</table>

The software intensity is based on the share of software patents. The patent-related data for regression estimations presented in this table are drawn from the CASSIS patent database maintained by the United States Patent and Trademark Office and from the NBER Patent Data Project database. Firm-level R&D data are collected from Compustat database, Edgar database, Amadeus database, the Kaisha Shiki Ho Survey database, R&D scoreboard, TS 2000 database (the Korea Listed Companies Association), and firm annual reports.
Table III: Tobin’s Q regressions, Nonlinear Least Squares, 1981-2005

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NLS</td>
<td>NLS</td>
<td>NLS</td>
<td>NLS</td>
<td>NLS</td>
</tr>
<tr>
<td>RD/Assets</td>
<td>-0.0938*</td>
<td>-0.0257</td>
<td>-0.134*</td>
<td>0.0969</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0562)</td>
<td>(0.187)</td>
<td>(0.0789)</td>
<td>(0.0768)</td>
<td></td>
</tr>
<tr>
<td>RD/Assets *</td>
<td>0.822***</td>
<td>0.152</td>
<td>0.392**</td>
<td>1.777***</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.142)</td>
<td>(0.298)</td>
<td>(0.163)</td>
<td>(0.317)</td>
<td></td>
</tr>
<tr>
<td>Software Intensity</td>
<td>-0.259***</td>
<td>-0.606***</td>
<td>-0.162***</td>
<td>-0.256***</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0435)</td>
<td>(0.0801)</td>
<td>(0.0572)</td>
<td>(0.0667)</td>
<td></td>
</tr>
<tr>
<td>Industry Dummies</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Year Dummies</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Number of Obs</td>
<td>2363</td>
<td>371</td>
<td>732</td>
<td>1260</td>
<td></td>
</tr>
<tr>
<td>Adj R-Squared</td>
<td>0.417</td>
<td>0.488</td>
<td>0.420</td>
<td>0.507</td>
<td></td>
</tr>
</tbody>
</table>

The software intensity is based on the share of software patents. The patent-related data for regression estimations presented in this table are drawn from the CASSIS patent database maintained by the United States Patent and Trademark Office and from the NBER Patent Data Project database. Firm-level R&D data are collected from Compustat database, Edgar database, Amadeus database, the Kaisha Shiki Ho Survey database, R&D scoreboard, TS 2000 database (the Korea Listed Companies Association), and firm annual reports. Other firm-level financial data (such as assets, long-term debt, short-term debt, the number of stocks and the price of stocks) are drawn from Compustat database, the Development Bank of Japan (BDJ) database, and the TS 2000 database (the Korea Listed Companies Association).
Table IV: Tobin’s Q regressions, OLS with Firm Fixed Effects, 1981-2005

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OLS/FE</td>
<td>OLS/FE</td>
<td>OLS/FE</td>
<td>OLS/FE</td>
</tr>
<tr>
<td>RD/Assets</td>
<td>-0.297*</td>
<td>1.633*</td>
<td>-0.222</td>
<td>-0.147</td>
</tr>
<tr>
<td></td>
<td>(0.178)</td>
<td>(0.951)</td>
<td>(0.288)</td>
<td>(0.108)</td>
</tr>
<tr>
<td>RD/Assets *</td>
<td>0.250</td>
<td>-1.218</td>
<td>0.634</td>
<td>0.611***</td>
</tr>
<tr>
<td>Software Intensity</td>
<td>(0.253)</td>
<td>(1.016)</td>
<td>(0.398)</td>
<td>(0.213)</td>
</tr>
<tr>
<td>Year Dummies</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Number of Obs</td>
<td>2363</td>
<td>371</td>
<td>732</td>
<td>1260</td>
</tr>
<tr>
<td>Adj R-Squared</td>
<td>0.096</td>
<td>0.093</td>
<td>0.026</td>
<td>0.012</td>
</tr>
</tbody>
</table>

The software intensity is based on the share of software patents. The data for the estimations presented in this table are drawn from the CASSIS patent database maintained by the United States Patent and Trademark Office and from the NBER Patent Data Project database. Firm-level R&D data are collected from the Compustat database, EDGAR database, the Kaisha Shiki Ho Survey database, R&D scoreboard, TS 2000 database (the Korea Listed Companies Association), and firm annual reports. Other firm-level financial data (such as assets, long-term debt, short-term debt, the number of stocks and the price of stocks) are drawn from Compustat database, the Development Bank of Japan (BDJ) database, and the TS 2000 database (the Korea Listed Companies Association).
Table V: Falsification Regressions, Negative Binomial, Random Effects and Fixed Effects

<table>
<thead>
<tr>
<th>Dependent Variable</th>
<th>Number of Patents</th>
<th>Number of Claims</th>
<th>Number of Citations (81-06)</th>
<th>Number of Citations (81-10)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1)</td>
<td>(2)</td>
<td>(3)</td>
<td>(4)</td>
</tr>
<tr>
<td></td>
<td>(5)</td>
<td>(6)</td>
<td>(7)</td>
<td>(8)</td>
</tr>
<tr>
<td>Log R&amp;D</td>
<td>0.0546***</td>
<td>0.0355**</td>
<td>0.0661***</td>
<td>0.0511***</td>
</tr>
<tr>
<td></td>
<td>(0.0142)</td>
<td>(0.0145)</td>
<td>(0.0123)</td>
<td>(0.0126)</td>
</tr>
<tr>
<td>Firm Size Dummy</td>
<td>-0.123</td>
<td>-0.258**</td>
<td>0.0835</td>
<td>0.0178</td>
</tr>
<tr>
<td></td>
<td>(0.121)</td>
<td>(0.121)</td>
<td>(0.119)</td>
<td>(0.119)</td>
</tr>
<tr>
<td>Firm Size Dummy * 1986-1990</td>
<td>-0.0869</td>
<td>-0.0924</td>
<td>-0.0625</td>
<td>-0.0638</td>
</tr>
<tr>
<td></td>
<td>(0.124)</td>
<td>(0.122)</td>
<td>(0.131)</td>
<td>(0.131)</td>
</tr>
<tr>
<td>Firm Size Dummy * 1991-1995</td>
<td>-0.175</td>
<td>-0.195*</td>
<td>-0.149</td>
<td>-0.169</td>
</tr>
<tr>
<td></td>
<td>(0.119)</td>
<td>(0.118)</td>
<td>(0.125)</td>
<td>(0.125)</td>
</tr>
<tr>
<td>Firm Size Dummy * 1996-2000</td>
<td>-0.121</td>
<td>-0.158</td>
<td>-0.0660</td>
<td>-0.103</td>
</tr>
<tr>
<td></td>
<td>(0.116)</td>
<td>(0.114)</td>
<td>(0.121)</td>
<td>(0.121)</td>
</tr>
<tr>
<td>Firm Size Dummy * 2001-2005</td>
<td>-0.356***</td>
<td>-0.379***</td>
<td>-0.256**</td>
<td>-0.274**</td>
</tr>
<tr>
<td></td>
<td>(0.119)</td>
<td>(0.118)</td>
<td>(0.123)</td>
<td>(0.123)</td>
</tr>
<tr>
<td>1986-1990</td>
<td>0.217**</td>
<td>0.230**</td>
<td>0.271**</td>
<td>0.278**</td>
</tr>
<tr>
<td></td>
<td>(0.106)</td>
<td>(0.104)</td>
<td>(0.111)</td>
<td>(0.110)</td>
</tr>
<tr>
<td>1991-1995</td>
<td>0.316***</td>
<td>0.347***</td>
<td>0.412***</td>
<td>0.437***</td>
</tr>
<tr>
<td></td>
<td>(0.102)</td>
<td>(0.0996)</td>
<td>(0.105)</td>
<td>(0.104)</td>
</tr>
<tr>
<td>1996-2000</td>
<td>0.554***</td>
<td>0.606**</td>
<td>0.745***</td>
<td>0.784***</td>
</tr>
<tr>
<td></td>
<td>(0.100)</td>
<td>(0.0979)</td>
<td>(0.103)</td>
<td>(0.102)</td>
</tr>
<tr>
<td>2001-2005</td>
<td>0.307***</td>
<td>0.355***</td>
<td>0.476***</td>
<td>0.509***</td>
</tr>
<tr>
<td></td>
<td>(0.103)</td>
<td>(0.101)</td>
<td>(0.105)</td>
<td>(0.104)</td>
</tr>
<tr>
<td>Industry Dummies</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Number of Obs</td>
<td>3206</td>
<td>3205</td>
<td>3206</td>
<td>3205</td>
</tr>
</tbody>
</table>

Sales data are used to define firm’s size. Firm size dummy is defined as one if the firm’s sale is above median. The average value of sales from 1996 to 2005 is calculated because of the following reasons: (1) some firms have missing sales value in the 1980s and (2) sales tend to increase over time. The regression results using the average value of sales from 1981 to 2005 are qualitatively identical. The results are available from the authors by request.