GET WITH THE PROGRAM:
SOFTWARE-DRIVEN INNOVATION IN TRADITIONAL MANUFACTURING

Lee G. Branstetter
Matej Drev
Namho Kwon

Get With the Program: Software-Driven Innovation in Traditional Manufacturing
Lee G. Branstetter, Matej Drev, and Namho Kwon
NBER Working Paper No. 21752
November 2015, Revised January 2016
JEL No. O14,O31,O32,O33

<u>ABSTRACT</u>

This paper documents 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 229 publicly listed firms from 18 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 patents per R&D dollar, and their investment in R&D is more highly valued by equity markets. We present evidence that geographic differences in the abundance of skilled software labor are an important factor in determining sample firms' software intensity and performance.

Lee G. Branstetter
Heinz College
School of Public Policy and Management
Department of Social and Decision Sciences
Carnegie Mellon University
Pittsburgh, PA 15213
and the Peterson Institute for International Economics
and also NBER
branstet@andrew.cmu.edu

Matej Drev
School of Public Policy
Georgia Institute of Technology
685 Cherry Street
Atlanta, GA 30332
matej.drev@pubpolicy.gatech.edu

Namho Kwon
Heinz College
Carnegie Mellon University
Pittsburgh, PA 15213
nkwon@andrew.cmu.edu

A online appendix is available at http://www.nber.org/data-appendix/w21752

I.   Introduction

"We believe that every industrial company will become a software company."

- GE CEO Jeffrey Immelt, in his annual letter to GE shareholders, 2014.[1]

This paper documents a significant change in the nature and direction of invention in four "traditional" manufacturing industries. In autos and auto parts, aerospace and defense, medical devices, and pharmaceuticals, inventors and product developers are increasingly using software, rather than more traditional mechanical or chemical engineering, to differentiate products, enhance product performance, and increase user utility. Engineers and industry experts in these sectors have provided anecdotal evidence of a software-biased shift in the trajectory of innovation, but this evidence has generally rested on a relatively small number of possibly unrepresentative firms and products. Using much more comprehensive patent and patent citation data, we present new statistical evidence showing that this software-biased shift is persistent, systematic, and increasingly pervasive. We also point to other indicators suggesting that this shift extends far beyond the boundaries of our four target industries.

If software has indeed become an increasingly important input into the creation of new inventions, then firms that take advantage of this software-biased shift should become better innovators than their industry peers who do not. Our empirical analysis suggests this is exactly what has happened. Using an unbalanced panel of publicly traded firms in our four target industries over the period 1981-2005, we show that the firms in these industries that have taken a more software-intensive approach to innovation have increasingly outperformed their less software-intensive peers in terms of patent productivity (patents per R&D dollar) and the market value of their R&D investment.[2] We also show that the relative performance of software-intensive firms improves at the same time that the software-intensity of

---

[1] GE also recently announced that it is moving its company headquarters to Boston in order to be a part of an "ecosystem that shares its aspirations" to become a "top 10 software company" by 2020. For details, see Lohr (2016).
[2] 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), Bloom, Sadun, and Van Reenen (2012), Huang, Ceccagnoli, Forman, and Wu (2015), Brynjolfsson and Saunders (2010), Aral and Weill (2007), and McElheran (2015) 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).

innovation in these sectors grows. Using a variety of robustness checks, we demonstrate that we can exclude several competing explanations for the observed outperformance of the software-intensive firms.

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 that geographic differences in the abundance of skilled software engineers have been an important factor in determining sample firms' software intensity and innovation performance. Drawing upon multiple data sources, we present new estimates of the numbers of IT / software engineers available for hire in the major national labor markets where our sample firms are based. We supplement these estimates with data on flows of immigrant IT professionals and the magnitude of offshoring of software engineering. We find large and growing asymmetries across countries in the endowments of software engineering human resources, providing a partial explanation for the differences in the software intensity of R&D that we observe in firms headquartered in different countries.

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 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

A survey of the engineering literature suggests a pronounced increase in the importance of software for product development and innovation across a range of manufacturing industries. In this paper, we focus on automobiles and auto parts, aerospace and defense, medical devices, and pharmaceuticals. However, there is evidence indicating this trend extends to construction equipment, farm implements, IT

hardware, and power generation and transmission, among other industries.[3]

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). 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.

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 and auto parts 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 with software firms.[4] Software design teams have become increasingly prominent decision-makers at the product design stage (Mustapic et al, 2004). 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

---

[3] For a qualitative overview of the impact of these trends in farm implements and electric power, see Biba (2014) and Clancy (2014), respectively. The rising importance of software in construction and mining equipment was verified by direct conversations with Carnegie Mellon-affiliated robotics experts and profiled in Green (2012). See Arora, Branstetter, and Drev (2012) for an econometric examination of the rising role of software in IT hardware product development. Our decision to focus on the four sectors examined in this paper was driven, in part, by data availability. As we discuss in the paper, the limited availability of R&D expenditure data was a major constraint.

[4] 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.

difficult than designing the 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, Jetley, and Abraham, 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. However, the software intensity of medical devices has been accelerating particularly quickly in the past decade (Wasden, 2011).[5]

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).

---

[5] We confirmed these trends through interviews with engineering professors who have closely followed technological trends across a range of medical device technologies.

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).[6] 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 Zedwitz, 2010). 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.[7] However, the engineering literature documenting this shift has tended to be largely anecdotal, relying heavily on a small sample of probably unrepresentative products and firms. In the next section, we use far more comprehensive patent and 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

A. 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

---

[6] It is important to note that, in most of our target industries, software has become increasingly embedded in the products themselves. However, in the pharmaceutical industry, software has significantly affected the innovation process but not the products. This significant difference led us to rerun our main empirical specifications without including pharmaceutical firms – results confirm that our main results are not qualitatively affected by the inclusion or exclusion of this industry.

[7] See Bharadwaj et al (2013) to study how digital technologies have been changing firms' business strategies, capabilities, products and services. Iansiti and Lakhani (2014) provide excellent examples of wide digital transformation and connection in traditional industries.

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. 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 α(p, P) below, as well as the time lag between the grant years of the two patents, (t-T):

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

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:

(2) $$E(citation_{tabTc}) = n_{tab} n_{Tc} \alpha_{tabTc} \exp\left(-\beta_1(t - T)\right) \cdot \left(1 - \exp\left(-\beta_2(t - T)\right)\right)$$

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 $ntab$ and nTc are the counts of patents in the respective categories. Rewriting equation (2) gives us the Jaffe – Trajtenberg (2002) version of the citation function, expressing the average number of citations from one category of patents to another:

(3) $$P(citation_{tabTc}) = \frac{E(citation_{tabTc})}{n_{tab} \cdot n_{Tc}} = \alpha_{tabTc} \cdot \exp\left(-\beta_1(t - T)\right) \cdot \left(1 - \exp\left(-\beta_2(t - T)\right)\right)$$

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.

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

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

(5) $$w = \sqrt{(n_{tab})(n_{Tc})}$$

B. 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.[8] 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, the TS2000 and FS2000 Databases from the Korean Listed Companies Association (KCLA), and the U.K. R&D Scorecard.[9]

In the next step, we connected the identified firms to their U.S. patent portfolios using the updated NBER patent database.[10] 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.[11] 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.

---

[8] We used sales in 2010. The choice of base year had little impact on our sample. Using sales data from 2000 or 2005 produced a list very similar to our target firms.

[9] The lack of R&D data for firms based outside the United States was a major constraint. Our measures of R&D productivity require data on R&D expenditure. U.S. accounting practices require publicly traded firms to fully disclose their R&D expenditures, but this principle is not widely followed outside the United States. A number of non-U.S. firms have been excluded from our data set due to a lack of publicly available information on R&D expenditures.

[10] 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).

[11] This accounting is based on patent grant years.

Next, we identified software related patents, which is a perennial challenge in the empirical literature. 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 1st 1985 and before December 31st 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 318,105 patents, 17,956 of which were assigned to firms in our sample.

As is the case in most studies that rely on patent data, our inference will need to be guided by an awareness of our data limitations. In the early years of our sample, the U.S. patent system operated under legal guidelines that held "pure" software inventions to be unpatentable. As that legal doctrine changed, the number of patented software inventions grew sharply.[12] This makes it imperative that we directly control for the expansion in the pool of software patents over time, which is exactly what the citation function approach allows us to do. Our analysis relies on patents granted by a single authority – the USPTO – to measure invention for both the United States and foreign firms in our sample. However, 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.[13] 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.

C. Results

We first look at the descriptive results presented in Figure 1, which show a stark increase over time in the software intensity of innovation for firms in our sample industries. As a total, the share of software patents in their patent portfolios increased from 3.5% to 14.4% over the sample period, a fourfold increase. Similarly, the share of citations going to software increased threefold over the sample period from 4.5% to 13.4%, with a particularly striking increase in the period following the year 2000.[14] Importantly, we observe the same trend when we look at the share of software citations coming from non-

---

[12] 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, as well as the various approaches to defining software patents.

[13] U.S. patents have been used to measure inventive output in Britain (Griffith, Harrison, and Van Reenen, 2006), Japan (Branstetter and Sakakibara, 2002), Israel (Trajtenberg, 2001), and a number of other countries.

[14] 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.

software (e.g. chemical, mechanical, and electrical engineering) patents. This suggests that that the observed sharp increase in the software intensity of innovation by our sample firms over the sample period is not entirely driven by the proliferation of software patents.

[Insert Figure 1 Here]

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 the first column is positive, large, and statistically significant, indicating that patents belonging to our sample firms are 24% 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 the third and fourth columns further solidify the point that there has been a sharp increase in the likelihood of citing software patents from 1986 to 2005.[15] In these specifications, we restrict the population of potentially cited patents to include only software patents. The key result is illustrated in Figure 2 below, which plots the coefficients from the third column of Table I, along with their 95% confidence intervals. This figure shows a 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.[16] 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 a high degree of statistical significance. These results align closely with the descriptive trends reported in Figure 1. We see this as strong evidence that the trajectory of technological

---

[15] 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.

[16] Figure 2 has a vertical light grey line in 1995 when the coefficient and its lower bound become greater than zero.

9

change in our sample industries has become substantially more software-intensive.

[Insert Figure 2 Here]

In results not shown in the paper, we estimated the citation function by excluding patents granted to pharmaceutical firms from our sample patents. Because software in this industry should be mainly related to process innovation, while software in other industries (auto and auto parts, aerospace and defense, and medical devices) would be related to both process innovation and product innovation.[17] The regression results using this restricted patent data are qualitatively similar to the results in Table I. 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.

A. 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.

$$(6) \qquad\qquad P_{it} = r_{it}^{\beta} \phi_{it}\, e^{\omega SW_i}$$

---

[17] Software itself can be embedded in the products of three industries (auto and auto parts, aerospace and defense, and medical devices). But software can hardly be embedded in the main products (drugs) of pharmaceutical industry.

(7) $$\text{where } \phi_{it} = e^{\sum_c \delta_c D_c}$$

In equation (6), P_{it} are patents taken out by firm i in period t, r_{it} are research and development expenditures, $\emptyset_{it}$ 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:

(8) $$\ln(P_{it}) = \beta \ln(r_{it}) + \sum_c \delta_c D_c + \omega SW_i + \mu_{it}$$

The error term is defined below:

(9) $$\mu_{it} = \xi_i + u_{it}$$

We allow the error term in (9) to contain a firm-specific component ξ_i, which accounts for the intra-industry firm-specific unobserved heterogeneity, as well as an *iid* random disturbance u_{it}. While SW_i would be swept out in a linear model with firm fixed effects, because it is time invariant, we can interact SW_i with dummy variables corresponding to subperiods of our 1981-2005 time frame and estimate the coefficients on the interaction terms.[18] 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).

B. 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 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;

---

[18] 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).

Hall and Kim, 2000; Bharadwaj et al, 1999).[19] 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:

$$(10) \qquad V_{it}(A, K) = q_t(A_{it} + \gamma_t K_{it})^{\sigma_t}$$

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, γ_t is the shadow value of the firm's technological knowledge measuring the firm's private returns to R&D, and σ_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:

$$(11) \qquad ln\, V_{it} = ln\, q_t + \sigma_t\, ln\, A_{it} + \sigma_t ln\, (1 + \tfrac{\gamma_t K_{it}}{A_{it}})$$

By assuming σ_t equals one (constant returns to scale) and subtracting $ln\, A_{it}$ on both sides, we can obtain the following equation:

$$(12) \qquad ln(V_{it}/A_{it}) = ln\, q_t + ln\, (1 + \tfrac{\gamma_t K_{it}}{A_{it}})$$

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:

$$(13) \qquad ln(Q_{it}) = ln\, q_t + ln\, [1 + \gamma_t \left(\tfrac{K_{it}}{A_{it}}\right)]$$

Following Hall and Kim (2000) and Arora, Branstetter, and Drev (2013), we estimate equation (13) using nonlinear least squares estimators (NLS).[20] 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.

---

[19] Our notation follows Hall and Kim (2000).

[20] Fixed effects and random effects estimators are used. For robustness checks, we estimated a linearized version of equation (13) using firm fixed effects.

C. Data and Variables

Sample Firms.—The procedure used to identify our sample firms was already delineated in the previous section (see p. 8). 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 information on R&D expenditures.[21] This yielded an unbalanced panel of 229 firms from 18 countries for patent production function analysis. For the Tobin's Q analysis, the unbalanced panel data contains 148 firms from 13 countries.[22] While the number of firms is not large, the share of global output represented by our sample firms in their respective industries is substantial. 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 to the total number of patents generated by the firm. The second software intensity variable is constructed using the share of citations to software patents in 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. Therefore, the value of this 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

---

[21] For innovation (patent) production analysis, we drop a firm if it has less than five years of R&D flow information. We exclude a firm if it has less than five years of R&D stock information for Tobin's Q analysis. Changing these thresholds does not change our results significantly.

[22] We note that the results of our patent production function regressions do not qualitatively change if we restrict our set of observations to those used in the Tobin's Q analysis.

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.

Sample Firms.— **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.

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.[23] We exploit the *EDGAR* database to collect R&D information for some firms that are not captured in Compustat or the R&D Scoreboard. 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.[24] 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 (Griliches, 1984; Hall,

---

[23] 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.

[24] 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.

1990).[25] The initial R&D stock was calculated using the previous five years' worth of R&D expenditure flows.[26] 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.[27]

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.[28]

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.

D. 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 have to go beyond a descriptive analysis as many factors, such as differences in R&D investment, can influence any observed

---

[25] 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 the online appendix.
[26] 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.
[27] 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.
[28] See Perfect and Wiles (1994) for a detailed discussion of measurement error issues when using book values.

changes in inventive output. As a consequence, 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. The magnitudes of the key coefficients from the second column of Table II are graphically represented in Figure 3, 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. The R&D productivity gap began to emerge since the early 1990s. 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).[29] These firms became about 40% 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.

[Insert Figure 3 Here]

The results reported in Table II were estimated using a negative binomial model, though our results are robust to the exact choice of specification.[30] 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. The third and fourth columns report regression results using the number of claims within firm i's cohort of patents applied for in year t as the dependent variable. The fifth and sixth columns report results using the number of forward citations received by firm i's cohort of patents applied for in year t as the dependent

---

[29] Figure 3 doesn't necessary tell us that the productivity gap in the late 1990s is slightly greater than the one in the early 2000s. The coefficient of the interaction term of the last period (2001-2005) is generally greater than the one of the previous period (1996-2000) when estimating with the other software intensity (share of citations directed to prior software patents) and a Poisson model. The size of the coefficients of these two periods also marginally changes with different dependent variables (number of claims and number of citations).

[30] Use of a Poisson regression model yields similar coefficients.

variable. The 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 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.[31] Most of the coefficients on our key variables 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 estimated the regressions by excluding pharmaceutical firms from our sample. The regression results excluding pharmaceutical firms are qualitatively similar to Table II. 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 also re-estimated the regressions reported in Table II, but 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 are 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 online appendix.

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.[32] 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 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.

E. 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

---

[31] We regard the number of claims and the number of citations as proxies of the quality of the patent.
[32] Aral, Brynjolfsson, and Wu (2006) show the existence of the "virtuous cycle" in the relationship between IT investment and productivity.

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 4 graphically depicts the average difference in the estimated private returns to R&D between above- and below-median software intensity firms.[33] In Table III and Figure 4, software intensity is inferred from the share of software patents in a firm's total patents.

[Insert Figure 4 Here]

Figure 4 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.[34] 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 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. The regression results are consistent with our main results when pharmaceutical firms are excluded from our sample. 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 re-ran 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

---

[33] It is calculated as the difference between the below-median software intensity group subtracted from above-median software intensity group.

[34] Time periods are somewhat different from the patent production function analysis. We added additional years for each period because of the limited number of observations.

the paper. The results of all of these robustness checks are available from the online appendix.

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 mind-sets, 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.[35] 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 five of the most significant home bases for our sample firms are Japan, Germany, the United Kingdom, France, and 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 Figure 5, which portrays 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.[36]

[Insert Figure 5 Here]

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.[37] 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 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

---

[35] See Hunt and Gauthier-Loiselle (2010) and Kerr and Lincoln (2010) for studies highlighting the important role of immigrants in American innovation.
[36] A discussion of the multiple sources of these data is provided in the supplementary online appendix.
[37] Arora, Forman, and Yoon (2008) provide evidence that the United States has been a leading exporter of software products.

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.[38] The managerial literature highlights the challenges that arise when firms seek to do strategically significant R&D abroad (Anchordoguy, 2000).[39]

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 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

---

[38] 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).

[39] Jaffe, Trajtenberg, and Henderson (1993) find that national boundaries limit knowledge spillovers. Branstetter (2006) finds that Japanese FDI facilitates knowledge spillovers between U.S. and Japanese inventors, but the impact is limited. Froman, Goldfarb, and Greenstein (2015) suggest that a potential mechanism of geographic concentration of invention measured by patents is the low cost-distance digital communication.

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 6 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 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 7%. However, when these same firms conduct innovation abroad, which is primarily in the United States, this share rises to about 16%, reaching the share of software patents in the patent portfolios of US firms inventing at home (16%). 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.

[Insert Figure 6 Here]

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 managers 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 further. As is always the case in economics, more work remains to be done.

REFERENCES

Anchordoguy, Marie. 2000. "Japan's Software Industry: A Failure of Institutions," Research Policy, Vol. 29(3), pp. 391-408.

Aral, Sinan, and Erik Brynjolfsson, and D. J. Wu. 2006. "Which Came First, IT or Productivity? Virtuous Cycle of Investment and Use in Enterprise Systems," IT Business Value, International Conference on Information Systems

Aral, Sinan, and Peter Weill. 2007. "IT Assets, Organizational Capabilities, and Firm Performance: How Resource Allocations and Organizational Differences Explain Performance Variation," Organization Science, Vol. 18(5), pp. 763-780.

Arora, Ashish, Lee G. Branstetter, and Matej Drev. 2013. "Going Soft: How the Rise of Software Based Innovation Led to the Decline of Japan's IT Industry and the Resurgence of Silicon Valley," The Review of Economics and Statistics, Vol. 95(3), pp. 757-775.

Arora, Ashish, Chris Forman, and Ji Woong Yoon. 2008. "Software," In Innovation in Global Industries: U.S. Firms Competing in a New World, edited by David C. Mowery and Jeff Macher, Washington, DC: National Academies Press, pp. 53-100.

Bakal, Martin, "Challenges and Opportunities for the Medical Device Industry: Meeting the New IEC 62304 Standard," IBM Life Sciences White Paper, 2011.

Bessen, James, and Robert M. Hunt. 2007. "An Empirical Look at Software Patents," Journal of Economics & Management Strategy, Vol. 6(1), pp. 157-189.

Bettis, Richard A., and Michael A. Hitt. 1995. "The New Competitive Landscape," Strategic Management Journal, Vol. 16(S1), pp. 7-19.

Bharadwaj, Anandhi S., Sundar G. Bharadwaj, and Benn R. Konsynski. 1999. "Information Technology Effects on Firm Performance as Measured by Tobin's q," Management Science, Vol. 45(7), pp. 1008-1024.

Bharadwaj, Anandhi S., Omar El Sawy, Paul Pavlou, and N. Venkatraman. 2013. "Digital Business Strategy: Toward a Next Generation of Insights," MIS Quarterly, Vol. 37(2), pp. 471-482.

Bloom, Nicholas, and John Van Reenen. 2007. "Measuring and Explaining Management Practices across Firms and Countries," Quarterly Journal of Economics, Vol. 122 (4), pp. 1351-1408.

Bloom, Nicholas and John Van Rennen. 2010. "Why Do Management Practices Differ Across Firms and Countries?" Journal of Economic Perspectives, Vol. 24(1), pp. 203-224.

Bloom, Nicholas, Rafaela Sadun, and John Van Reenen. 2012. "Americans Do I.T. Better: U.S. Multinationals and the Productivity Miracle," The American Economic Review, Vol. 102(1), pp. 167-201.

Brainard, William C., and James Tobin. 1968, "Pitfalls in Financial Model-Building," The American Economic Review, Vol. 58(2), pp. 99-122.

Branstetter, Lee G. 2006. "Is Foreign Direct Investment a Channel of Knowledge Spillovers? Evidence from Japan's FDI in the United States," Journal of International Economics, Vol. 68(2), pp. 325-344.

Branstetter, Lee G., and Mariko Sakakibara. 2002. "When Do Research Consortia Work Well and Why? Evidence from Japanese Panel Data," The American Economic Review, Vol. 92(1), pp. 143-159.

Brynjolfsson, Erik, and Lorin Hitt. 1995. "Information Technology as a Factor of Production: The Role of Differences across Firms," Economics of Innovation and New Technology, Vol. 3 (3-4), pp. 183-200.

Brynjolfsson, Erik, and Kristina McElheran. 2015. "Data in Action: Data Driven Decision Making in U.S. Manufacturing," Working Paper.

Brynjolfsson, Erik, and Adam Saunders. 2010. "Wired for Innovation: How Information Technology is Reshaping the Economy," Cambridge, MA: MIT Press

Caballero, Richard, and Adam Jaffe. 1993. "How High are the Giants' Shoulders: An Empirical Assessment of Knowledge Spillovers and Creative Destruction in a Model of Economic Growth," NBER Macroeconomics Annual, Vol. 8, pp.15-74.

Cole, Robert E. 2006. "Software's Hidden Challenges." In D. Whittaker and R. Cole, Eds., Recovering from Success: Innovation and Technology Management in Japan. London: Oxford University Press.

Cole, Robert E., and Shinya Fushimi. 2011. "The Japanese Enterprise Software Industry." In H Miyoshi and Y. Nakata, Eds., Have Japanese Firms Changed? London: Palgrave Macmillan.

De Roze, Barry C., and Thomas N. Nyman. 1978. "The Software Life Cycle - A Management and Technological Challenge in the Department of Defense," IEEE Transactions on Software Engineering, Vol. 4(4), pp. 309-318.

Denger, Christian, et al. 2007. "A Snapshot of the State of Practice in Software Development in Medical Devices," Proceedings of the First International Symposium on Empirical Software Engineering and Measurement, pp. 485-487.

Drews, Jurgen. 2000. "Drug Discovery: A Historical Perspective," Science, 287(5460), pp. 1960-1964.

Duardo-Sanchez, Aliuska, Grace Patlewicz, and Antonio Lopez-Diaz. 2008. "Current Topics on Software Use in Medicinal Chemistry: Intellectual Property, Taxes, and Regulatory Issues," Current Topics in Medicinal Chemistry, Vol. 8(18), pp.1666-1675.

Forman, Chris, Avi Goldfarb, and Shane Greenstein. 2015. "Information Technology and the Distribution of Inventive Activity," In The Changing Frontier: Rethinking Science and Innovation Policy, edited by Adam Jaffe and Benjamin Jones, University of Chicago Press, pp. 169–196.

Frischkorn, Hans-Georg, "Automotive Software - The Silent Revolution," Talk at Automotive SW Workshop San Diego, January 10-12 2004.

Gassmann, Oliver, Gerrit Reepmeyer, and Maximilian Von Zedwitz, "Leading Pharmaceutical Innovation: Trends and Drivers for Growth in the Pharmaceutical Industry," New York, NY: Springer, 2010.

Graham, Stuart, and David Mowery. 2003. "Intellectual Property Protection in the U.S. Software Industry" in Cohen, W. and Merrill, S.A.: Patents in the Knowledge-Based Economy.

Griffith, Rachel, Rupert Harrison, and John Van Reenen. 2006. "How Special is the Special Relationship? Using the Impact of US R&D Spillovers on British Firms as a Test for Technology Sourcing," American Economic Review, Vol. 96(5), pp. 1859-1875.

Griliches, Zvi. 1981. "Market Value, R&D, and Patents", Economics Letters 7.

Griliches, Zvi, and Ariel Pakes. 1984. "Patents and R&D at the Firm Level: A First Look," R&D, Patents, and Productivity, University of Chicago Press

Grimm, Klaus. 2003. "Software Technology in an Automotive Company - Major Challenges," ICSE '03 Proceedings of the 25th International Conference on Software Engineering.

Hausman, Jerry, Hall, Bronwyn H., and Griliches, Zvi. 1984. "Econometric Models for Count Data with an Application to the Patents-R&D Relationship," Econometrica, Vol.52(4), pp. 909-938.

Hall, Bronwyn H. 1990. "The Manufacturing Sector Master File: 1959-1987, "NBER Working Paper No. 3366

Hall, Bronwyn H. and Kim, Daehwon. 2000. "Valuing Intangible Assets: The Stock Market Value of R&D Revisited," Working Paper.

Hall, Bronwyn H., Adam Jaffe, and Manuel Trajtenberg. 2001. "The NBER Patent Citations Data File: Lessons, Insights, and Methodological Tools," NBER Working Paper No. 8498.

Hall, Bronwyn H., and Raffaele Oriani. 2006. "Does the Market Value R&D Investment by European Firms? Evidence from a Panel of Manufacturing Firms in France, Germany," International Journal of Industrial Organization, Vol. 24(5), pp. 971-993.

Holden, Constance. 1986. "Regulating Software for Medical Devices," Science 234(4772), pp.20.

Holloway, C. Michael, and Kelly J. Hayhurst. 2003. "Software System Safety & the NASA Aeronautics Blueprint," Proceedings of the 21st International System Safety Conference.

Huang, Peng, Marco Ceccagnoli, Chris Forman, and D.J. Wu. 2015. "IT Knowledge Spillovers and Productivity: Evidence from Enterprise Software," Manuscript submitted for publication.

Hughes, David, "Informational Technology: This Changes Everything," Aviation Week & Space Technology. December 21/28, 1998.

Hunt, Jennifer, and Marjolaine Gauthier-Loiselle. 2010. "How Much Does Immigration Boost Innovation?" American Economic Journal: Macroeconomics, Vol. 2(2), pp. 31-56.

Iansiti, Marco, and Karim R. Lakhani. 2014. "Digital Ubiquity: How Connections, Sensors, and Data Are Revolutionizing Business," Harvard Business Review, Vol. 92(11), pp. 90–99.

Jaffe, Adam, and Manuel Trajtenberg. 1996. "Flows of Knowledge from Universities and Federal Labs: Modeling the Flow of Patent Citations over Time and across Institutional and Geographic Boundaries," NBER Working Paper No. 5712.

Jaffe, Adam, and Manuel Trajtenberg. 2002. "Patents, Citations, and Innovations: A Window on the Knowledge Economy," Boston: MIT Press.

Jaffe, Adam, Manuel Trajtenberg, and Rebecca Henderson. 1993. "Geographic Localization of Knowledge Spillovers as Evidenced by Patent Citations," Quarterly Journal of Economics, Vol. 108(3), pp. 577-598.

Joglekar, Nitin R., and Stephen R. Rosenthal, "Coordination of Design Supply Chains for Bundling Physical and Software Products," Journal of Product Innovation Management, Vol. 20(5), 2003, pp. 374-390.

Jones, Paul, Raoul Jetley, and Jay Abraham, "A Formal Methods-based Verification Approach to Medical Device Software Analysis," Electrical Engineering Times, February 9 2010.

Kahn, Alan. 1991. "The Changing Economics of Medical Technology," Washington, DC: Institute of Medicine.

Kerr, William R., and William F. Lincoln. 2010. "The Supply Side of Innovation: H1-B Visa Reforms and U.S. Ethnic Innovation," Journal of Labor Economics, Vol. 28(3), pp. 473-508.

Lohr, Steve. 2016. "G.E. Is Moving Headquarters to Boston and Itself Into the Digital Era," The New York Times, January 13, 2016. Retrieved from: http://www.nytimes.com/2016/01/14/technology/ge-boston-headquarters.html

Long, Lyle N. 2008. "The Critical Need for Software Engineering Education," Crosstalk - The Journal of Defense Software Engineering, January 2008, pp. 6-10.

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

Mai-Duc, Christine, "Medical Device Software Criticized as Under-regulated," Los Angeles Times, September 1 2011.

McElheran, Kristina. 2015. "Do Market Leaders Lead in Business Process Innovation? The Case(s) of E-Business Adoption," Management Science, Vol. 61(6), pp. 1197-1216.

McLaughlan, Gail and Salt, John, "Migration policies towards highly skilled foreign workers," Report to the Home Office, University College London, 2002.

Mustapic, Goran et al. 2004. "Real World Influences on Software Architecture – Interviews with Industrial System Experts," Proceedings of Fourth Working IEEE/IFIP Conference on Software Architecture, pp. 101-111.

Nelson, Ed, "Defining interfaces as Services in Embedded Vehicle Software," Talk at Automotive SW Workshop San Diego, January 10-12 2004.

Perfect, Steven B., and Kenneth W. Wiles. 1994. "Alternative Constructions of Tobin's Q: An Empirical Comparison," Journal of Empirical Finance, Vol. 1(3-4), pp. 313-341.

Potocki de Montalk, J.P., "Computer Software in Civil Aircraft," Microprocessors and Microsystems, Vol. 17(1), 1993, pp. 17-23.

Sandler, Karen, Lysandra Ohrstrom, Laura Moy, and Robert McVay. 2010. "Killed by Code: Software Transparency in Implantable Medical Devices," Software Freedom Law Center Working Paper.

Searls, David B., "Using Bioinformatics in Gene and Drug Discovery," Drug Discovery Today, Vol. 5(4), 2000, pp. 135-143.

Shorey, Rajeev, "Emerging Trends in Vehicular Communications," IEEE New York Presentation, June 8 2011.

"Software Takes on More Tasks in Today's Cars" - Wired, April 25 2011

Tobin, James. 1969, "A General Equilibrium Approach to Monetary Theory," Journal of Money, Credit and Banking, Vol. 1(1), pp. 15-29.

Trajtenberg, Manuel. 2001. "Innovation in Israel, 1968-1997: a Comparative Analysis Using Patent Data," Research Policy, Vol. 30(3), pp. 363-389.

Wang, May Dongmei, "In the Spotlight: Bioinformatics," IEEE Reviews in Biomedical Engineering, Vol. 4, 2011, pp. 3-5.

Wasden, Christopher, "Navigating Through the Mobile Healthcare Revolution," Medical Device and Diagnostic Industry, April 2011.

Waterman, Jason, "Detroit is the Latest Hotspot for Software Engineers," Waterman International Services Website Article, October 27 2011.
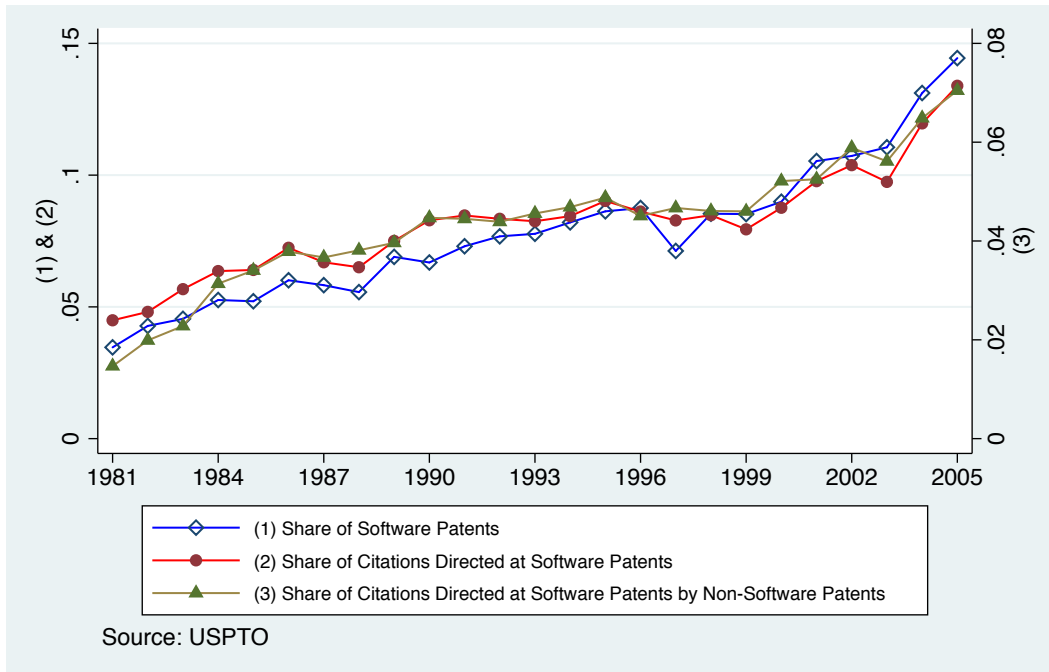
FIGURE 1: SOFTWARE INTENSITY OF PATENT PORTFOLIOS OF FIRMS ACTIVE IN FOUR MANUFACTURING INDUSTRIES – (1) SHARE OF SOFTWARE PATENTS, (2) TOTAL SHARE OF CITATIONS DIRECTED AT SOFTWARE PATENTS, AND (3) SHARE OF CITATIONS DIRECTED AT SOFTWARE PATENTS BY NON-SOFTWARE PATENTS
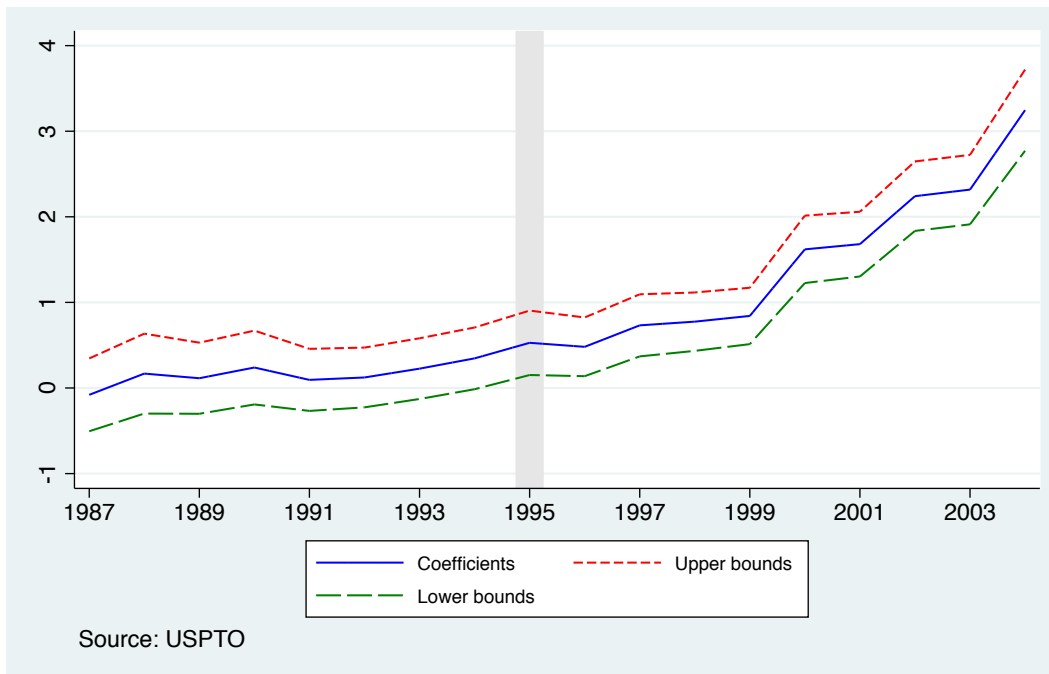

FIGURE 2: PROPENSITY OF NON-SOFTWARE PATENTS IN FOUR MANUFACTURING INDUSTRIES TO CITE PRIOR PATENTED SOFTWARE ART
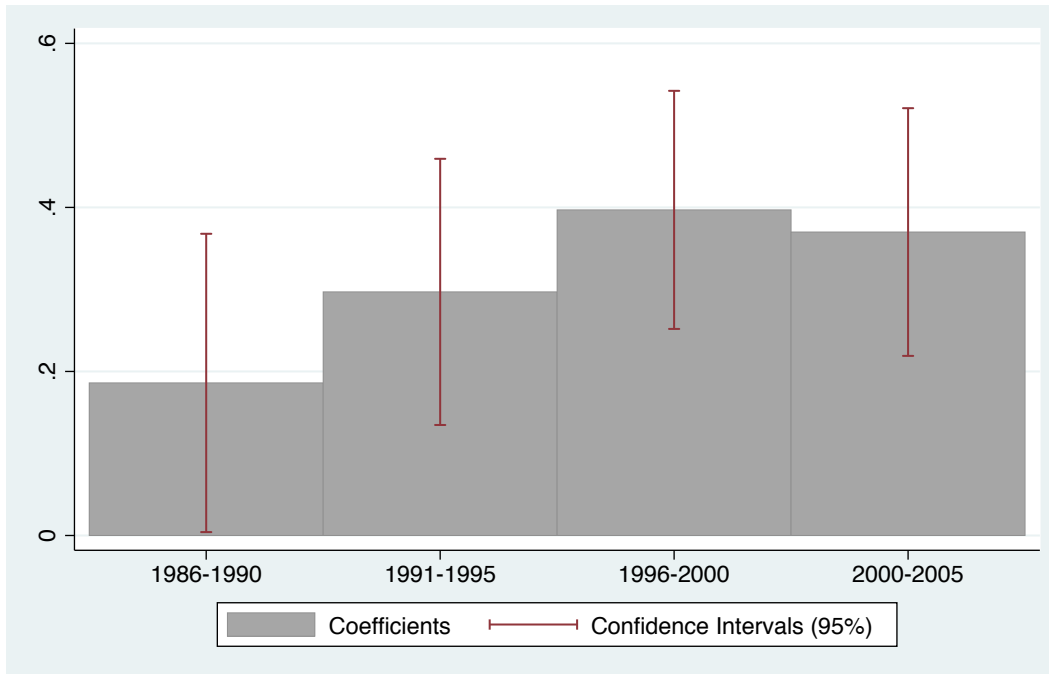
30

FIGURE 3: AVERAGE PATENT PRODUCTIVITY DIFFERENCE BETWEEN ABOVE-MEDIAN AND BELOW-MEDIAN SOFTWARE INTENSIVE FIRMS
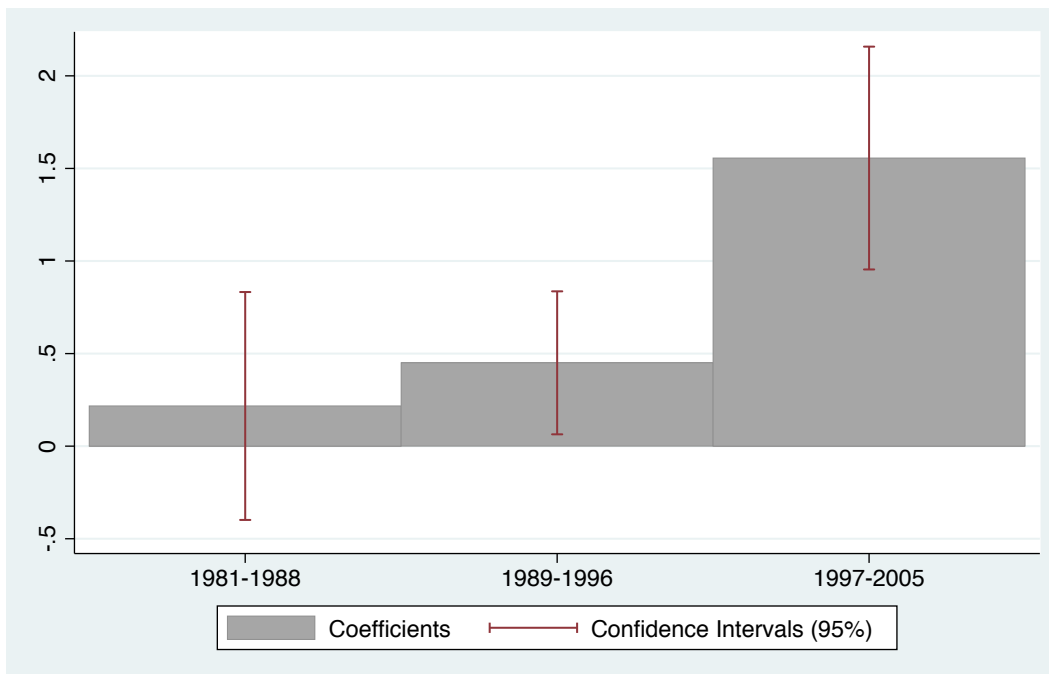
FIGURE 4: AVERAGE DIFFERENCE IN STOCK MARKET'S VALUATION OF R&D INVESTMENT BETWEEN ABOVE- AND BELOW-MEDIAN SOFTWARE INTENSITY FIRMS
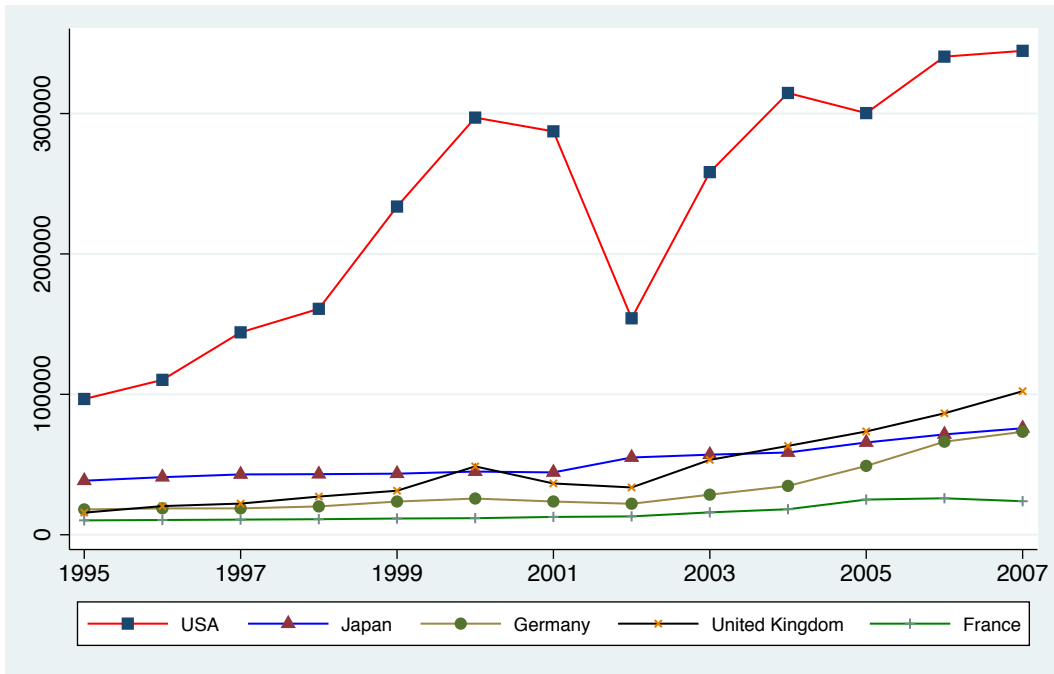
FIGURE 5: ICT HUMAN RESOURCES IN THE U.S., JAPAN, GERMANY, THE U.K., AND FRANCE / INFLOWS OF NEW WORKERS BY YEAR
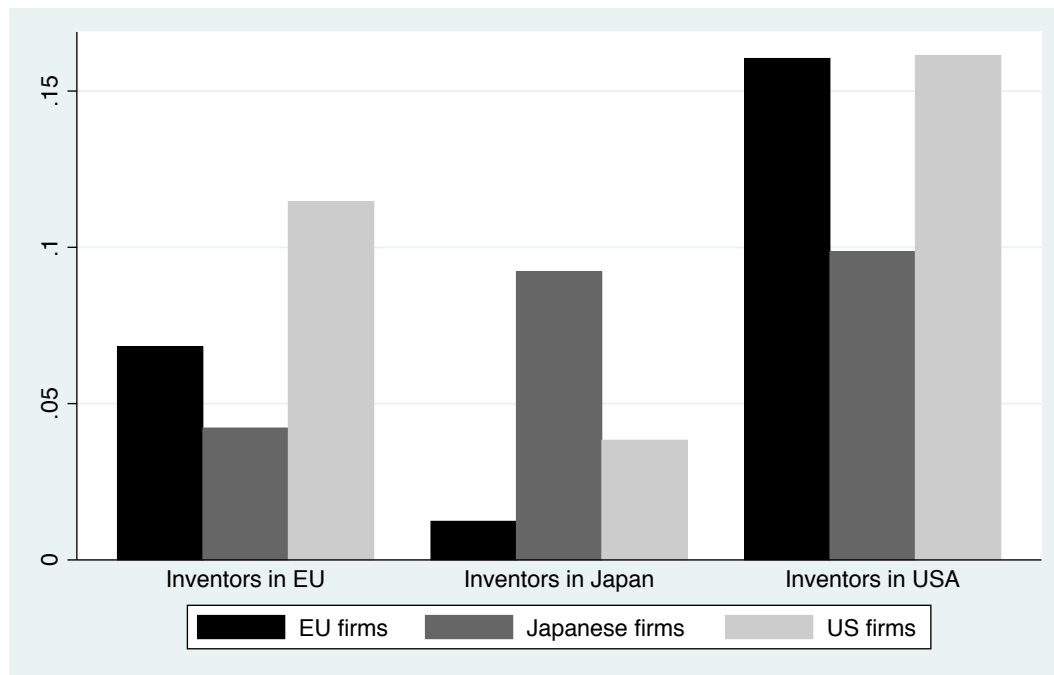
FIGURE 6: SOFTWARE INTENSITY OF PATENTED INVENTIONS (SHARE OF SOFTWARE PATENTS), BY GEOGRAPHY OF INVENTION AND COUNTRY OF OWNERSHIP - US, EU AND JAPAN

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 average 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 patents invented in the EU.

Table I: Citation Function Results

	Full Model				Citations from NSW to SW		Citations from SW to SW	
	Column 1		Column 2		Column 3		Column 4	
	Coefficient	Std.Error	Coefficient	Std.Error	Coefficient	Std.Error	Coefficient	Std.Error
Citing Grant Year								
1987	0.157	0.281	0.629*	0.261	-0.0798	0.217	0.770	0.418
1988	0.116	0.248	0.265	0.194	0.168	0.238	0.326	0.301
1989	0.193	0.243	0.462*	0.205	0.114	0.212	0.574	0.326
1990	0.219	0.233	0.649**	0.216	0.239	0.220	0.832*	0.355
1991	0.134	0.203	0.432*	0.177	0.0947	0.185	0.573*	0.288
1992	0.183	0.197	0.434**	0.166	0.123	0.178	0.570*	0.269
1993	0.260	0.194	0.392**	0.150	0.226	0.181	0.515*	0.243
1994	0.381	0.196	0.476**	0.147	0.346	0.184	0.589*	0.238
1995	0.611**	0.208	0.717***	0.157	0.528**	0.192	0.864***	0.256
1996	0.732***	0.203	0.905***	0.159	0.481**	0.175	1.111***	0.265
1997	0.843***	0.195	1.060***	0.157	0.732***	0.185	1.323***	0.269
1998	0.962***	0.184	0.987***	0.138	0.775***	0.174	1.201***	0.235
1999	1.128***	0.176	0.977***	0.128	0.842***	0.168	1.114***	0.218
2000	1.670***	0.188	1.348***	0.134	1.620***	0.201	1.508***	0.233
2001	2.000***	0.180	1.771***	0.139	1.681***	0.193	2.035***	0.252
2002	2.414***	0.172	2.211***	0.145	2.241***	0.207	2.511***	0.270
2003	2.867***	0.163	2.555***	0.149	2.318***	0.207	2.854***	0.288
2004	3.680***	0.168	3.588***	0.171	3.246***	0.242	4.264***	0.346
2005	4.420	.	4.379	.	4.329	.	5.315	.
Cited Grant Year								
1986	-0.0391	0.0506	-0.0950**	0.0347	-0.145**	0.0560	-0.126*	0.0511
1987	-0.0408	0.0517	-0.0744*	0.0362	-0.199***	0.0530	-0.109*	0.0530
…	…	…	…	…	…	…	…	…
2003	-0.913***	0.0237	-0.956***	0.00969	-0.954***	0.0143	-0.970***	0.0106
2004	-0.942***	0.0283	-0.973***	0.0125	-0.970***	0.0189	-0.982***	0.0138
Citing patent: Firm industry								
Aerospace and Defense	0.254***	0.0312	0.120***	0.0204	0.0547	0.0329	0.0938**	0.0304
Medical Devices	1.237***	0.0470	0.427***	0.0253	0.592***	0.0433	0.191***	0.0356
Pharmaceutical	-0.175***	0.0251	-0.499***	0.0171	-0.675***	0.0242	-0.628***	0.0267
Software Patent								
Citing from Software Patent			-0.136***	0.0329				
Cited Software Patent	0.239***	0.0212	-0.247***	0.0356				
Citing from Software Patent X Cited Software Patent			6.125***	0.115				
Obsolescence	0.284***	0.012	0.324***	0.009	0.321***	0.012	0.349***	0.013
Diffusion	4.38E-6***	1.06E-06	7.96E-6***	1.34E-06	7.01E-6***	1.54E-06	6.95E-5***	1.73E-05
Adj R-Squared	0.904		0.853		0.898		0.897	
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. * $p<0.10$, ** $p<0.05$, *** $p<0.01$

Table II: Innovation (Patent) Production Function Regressions, Negative Binomial, Random Effects and Fixed Effects

Dependent Variable	Number of Patents		Number of Claims		Number of Citations	
	NB: RE	NB: FE	NB: RE	NB: FE	NB: RE	NB: FE
	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
Log R&D	**0.073*****	**0.046*****	**0.178*****	**0.162*****	**0.166*****	**0.150*****
	(0.013)	(0.013)	(0.011)	(0.011)	(0.011)	(0.012)
Software Intensity Dummy	**-0.232****	**-0.212***	**-0.289****	**-0.304****	**-0.262****	**-0.260****
	(0.113)	(0.115)	(0.118)	(0.119)	(0.115)	(0.116)
Software Intensity Dummy * 1986-1990	**0.384*****	**0.398*****	**0.482*****	**0.510*****	**0.538*****	**0.555*****
	(0.127)	(0.128)	(0.142)	(0.142)	(0.134)	(0.134)
Software Intensity Dummy * 1991-1995	**0.489*****	**0.509*****	**0.465*****	**0.487*****	**0.561*****	**0.574*****
	(0.121)	(0.122)	(0.133)	(0.134)	(0.127)	(0.128)
Software Intensity Dummy * 1996-2000	**0.589*****	**0.609*****	**0.572*****	**0.595*****	**0.601*****	**0.618*****
	(0.115)	(0.116)	(0.127)	(0.128)	(0.123)	(0.123)
Software Intensity Dummy * 2001-2005	**0.557*****	**0.582*****	**0.580*****	**0.617*****	**0.447*****	**0.462*****
	(0.120)	(0.120)	(0.130)	(0.131)	(0.129)	(0.130)
1986-1990	**-0.164***	**-0.152**	**-0.279****	**-0.287*****	**-0.329*****	**-0.330*****
	(0.099)	(0.100)	(0.111)	(0.111)	(0.106)	(0.106)
1991-1995	**-0.080**	**-0.057**	**-0.113**	**-0.109**	**-0.245****	**-0.234****
	(0.094)	(0.095)	(0.104)	(0.104)	(0.100)	(0.100)
1996-2000	**0.274*****	**0.304*****	**0.333*****	**0.345*****	**-0.103**	**-0.085**
	(0.091)	(0.091)	(0.099)	(0.100)	(0.097)	(0.097)
2001-2005	**-0.261*****	**-0.224****	**-0.337*****	**-0.335*****	**-1.093*****	**-1.067*****
	(0.095)	(0.096)	(0.102)	(0.103)	(0.102)	(0.102)
Industry Dummies	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Number of Obs	3884	3873	3884	3873	3884	3873

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. * p<0.10, ** p<0.05, *** p<0.01

Table III: Tobin's Q regressions, Nonlinear Least Squares, 1981-2005

lnQ	Full Sample	1981-1988	1989-1996	1997-2005
	NLS	NLS	NLS	NLS
RD/Assets	-0.081	-0.083	-0.077	0.213**
	(0.059)	(0.183)	(0.079)	(0.096)
RD/Assets * Software Intensity	0.834***	0.217	0.450**	1.556***
	(0.152)	(0.314)	(0.197)	(0.307)
Software Intensity	-0.282***	-0.601***	-0.233***	-0.247***
	(0.044)	(0.090)	(0.064)	(0.065)
Industry Dummies	Yes	Yes	Yes	Yes
Year Dummies	Yes	Yes	Yes	Yes
Number of Obs	2288	365	719	1204
Adj R-Squared	0.415	0.459	0.418	0.502

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, Edgar, Amadeus, 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, the Development Bank of Japan (BDJ) database, and the TS 2000 (the Korea Listed Companies Association).
* $p<0.10$, ** $p<0.05$, *** $p<0.01$

Table IV: Tobin's Q regressions, OLS with Firm Fixed Effects, 1981-2005

lnQ	Full Sample	1981-1988	1989-1996	1997-2005
	OLS/FE	OLS/FE	OLS/FE	OLS/FE
RD/Assets	-0.312	2.329***	0.042	-0.165
	(0.214)	(0.767)	(0.241)	(0.212)
RD/Assets * Software Intensity	0.237	-1.990**	0.410	0.699**
	(0.318)	(0.835)	(0.517)	(0.281)
Year Dummies	Yes	Yes	Yes	Yes
Number of Obs	2288	365	719	1204
Adj R-Squared	0.100	0.131	0.019	0.016

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, EDGAR, 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).
* $p<0.10$, ** $p<0.05$, *** $p<0.01$

Table V: Falsification Regressions, Negative Binomial, Random Effects and Fixed Effects

Dependent Variable	Number of Patents		Number of Claims		Number of Citations	
	NB: RE	NB: FE	NB: RE	NB: FE	NB: RE	NB: FE
	(1)	(2)	(3)	(4)	(6)	(7)
Log R&D	0.055***	0.037**	0.068***	0.087***	0.081***	0.067***
	(0.014)	(0.015)	(0.012)	(0.015)	(0.013)	(0.013)
Firm Size Dummy	-0.156	-0.301**	0.070	-0.007	0.093	0.010
	(0.120)	(0.119)	(0.117)	(0.112)	(0.111)	(0.112)
Firm Size Dummy * 1986-1990	-0.121	-0.125	-0.095	-0.030	-0.012	-0.007
	(0.122)	(0.121)	(0.130)	(0.108)	(0.114)	(0.114)
Firm Size Dummy * 1991-1995	-0.172	-0.190	-0.150	-0.118	-0.101	-0.112
	(0.118)	(0.117)	(0.124)	(0.107)	(0.111)	(0.111)
Firm Size Dummy * 1996-2000	-0.113	-0.145	-0.074	0.164	0.126	0.104
	(0.114)	(0.113)	(0.119)	(0.109)	(0.110)	(0.110)
Firm Size Dummy * 2001-2005	-0.322***	-0.341***	-0.248**	-0.049	-0.072	-0.055
	(0.118)	(0.117)	(0.122)	(0.131)	(0.119)	(0.119)
1986-1990	0.245**	0.255**	0.300***	0.168*	0.180*	0.177*
	(0.104)	(0.101)	(0.108)	(0.091)	(0.097)	(0.096)
1991-1995	0.318***	0.346***	0.426***	0.073	0.185**	0.201**
	(0.100)	(0.098)	(0.103)	(0.090)	(0.094)	(0.093)
1996-2000	0.547***	0.592***	0.755***	-0.474***	-0.042	-0.010
	(0.098)	(0.096)	(0.100)	(0.093)	(0.094)	(0.093)
2001-2005	0.279***	0.324***	0.471***	-2.108***	-1.003***	-0.986***
	(0.101)	(0.100)	(0.103)	(0.109)	(0.100)	(0.100)
Industry Dummies	Yes	Yes	Yes	Yes	Yes	Yes
Number of Obs	3111	3110	3111	3104	3111	3110

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. * $p<0.10$, ** $p<0.05$, *** $p<0.01$