MMRC

DISCUSSION PAPER SERIES

No. 424

Killing Innovation Softly: Japanese Software Challenges

Robert E. Cole

Research Associate, ITEC, Doshisha University

Emeritus Professor, UC Berkeley, Haas School of Business

March 2013

Revised June 2013

東京大学ものづくり経営研究センター

Manufacturing Management Research Center (MMRC)

http://merc.e.u-tokyo.ac.jp/mmrc/dp/index.html

Killing Innovation Softly: Japanese Software Challenges*

Robert E. Cole

Research Associate, ITEC, Doshisha University
Emeritus Professor, UC Berkeley, Haas School of Business

Revised June 1, 2013

*The author gratefully acknowledges the funding of this research by grants from: the Institute for Technology, Enterprise and Competitiveness, Doshisha University, Kyoto, Japan; the California Management Review; and the Garwood Center for Corporate Innovation, Institute for Business Innovation, at the Haas School of Business, UC Berkeley. I am indebted to the feedback , help, and advice I received from the following individuals: Takahiro Fujimoto, Shinya Fushimi , Kenji Hiranabe, David Hodges, Toshiro Kita, David Levine, James Mok, David Mowery, Yoshifumi Nakata, Katsutoshi Shintani, Shinji Takai, Tetsuo Tamai, Hirofumi Tatsumoto and Masato Takeichi.

This paper aims at solving a puzzle. How do we explain why the world's third largest economy, Japan, second only to the U.S. in software sales, nevertheless, lags so significantly in other key measures of software performance. One can measure these lags in its miniscule software and information services exports, large software and information services imports, absence of path breaking software firms, and absence of a large independent software sector. [1] This paper focuses primarily but not exclusively on the role of software in the IT sector. This sector is seen commonly as encompassing the following industries: computer hardware, software, electronics, semiconductors, internet, telecom equipment, e-commerce and computer services.

My analysis will document Japan's software deficits relative to the U.S. and explore competing explanations for them. Specifically, I evaluate recent research on U.S. and Japanese IT patents regarding software innovation performance in the IT sector and the consequences of Japan's lagging software innovation capabilities. This analysis involves an examination of embedded, enterprise, and application software. I explore the role of path dependency as a factor limiting Japanese firm transitions from hardware centric to a more software focused approach. This leads into an examination of the structure of the American and Japanese software industries along with their differential propensity to customize and what that tells us about management priorities. These priorities impact the status and reputation of software jobs in firms and society at large and their consequences for attracting the best and the brightest to software careers.

In July 2010, IBM quietly restructured its Systems and Technology Group (its hardware group) to report to its software Group-an extraordinary development seen historically. It was IBM's hardware group which initially gave birth to its software group. Software as IBM has now come full circle. It is as if the grown up child is now controlling its aging parent. IBM's announcement of this restructuring explained that IBM wanted to bring to market more tightly integrated packages of hardware/software solutions.[2]

These developments at IBM capture the broader movement of hardware companies transforming themselves into software companies or hybrids. Many Japanese observers see this as a uniquely American phenomenon based on American weakness in manufacturing; American managers and workers, they believe, don't have that "monozukuri gene." It is commonly believed Japanese among manufacturing personnel that monozukuri is about Japan's unique ability and practices for building high quality continually improving precision hardware products. In this world view, software is simply a facilitator of hardware capability, an assistant and a controller, not in itself a driver of customer value. Prof. Takahiro Fujimoto has been a major promoter of monozukuri consciousness from the late 1990s. He adopts an information processing definition of monozukuri, a short version of which is: "A firm's products and processes are artifacts that have been designed. Firm's manufacturing (monozukuri) capability is its distinctive ability to effectively and efficiently transform the flow of design information from and to customers." This is accomplished through achieving a tight relationship between organizational capabilities and product architecture.[3] There is nothing in that definition which prohibits seeing software as a designed artifact nor as a driver of monozukuri capability. Yet, the promoters of monozukuri almost never frame the matter in this fashion. How did monozukuri get to be seen as primarily a hardware capability?

It is first of all an old concept arising from efforts to capture the virtues of Japanese craftsmanship, associated with material processing and/or mechanical production activities (typically carried out by smaller firms) in which Japan has excelled. Naturally, software did not figure into this initial formulation. Secondly, we can learn how the term both evolved and came back into fashion from observing how U.S. management responded to a major crisis. Neil Fligstein, in analyzing the response of U.S. business leaders to the crisis of the great depression, notes that initially managers were confused, their old formulas were no longer producing the expected results. As a response, managers turned to an intensified use of tools in their existing repertoire. When hit with a threat the magnitude and character of which is not well understand, managers initially will often fall back on familiar solutions.[4] What Fligstein found in the case of the U.S. was that when this approach failed to yield positive results, eventually some firms took the lead in developing new capabilities and products rather than relying on executing better its past strong capabilities and skillsets. It was this that gradually led these firms to a renewed prosperity. As Japanese managers began to stumble in IT manufacturing (e.g., loss of DRAM market to Koreans in 1990s), one can imagine them initially reasoning:

> We are having problems because we are not executing very well our standard practices and approaches that produced success in the past. We need to better execute what are our core capabilities.

Monozukuri is a concept, and indeed has become a management ideology, which characterizes Japanese manufacturing firm's core capabilities as they evolved in postwar Japan. In the late 1990s, Japanese IT firms were faced with a hollowing out of manufacturing capabilities, loss of key global markets and challenged by the rise of modular production and Asian competitors. It is in this environment that the monozukuri concept was resuscitated and elevated into a national strategy. Various government agencies began promoting monozukuri as the path to renewed corporate health and Japanese prosperity.

Not having yielded the desired results to date, some are calling for reevaluating monozukuri's value. In 2009, Dr. Hidenori Kimura published a book entitled, "Monozukuri Defeat". He argued that Japanese firms have excessive confidence in their manufacturing ability (their monozukuri) and that their belief in the superiority of the quality of their manufactured goods was no longer justified. In the digital age, the quality of manufactured goods doesn't differ among firms as it once did. He argued that what really counts today are software, the ability to make quick strategic decisions and to go global. It is an argument that stresses conditions have changed and new capabilities are required. Seen in this light, organizational memory which prioritizes monozukuri concepts, practices and successes are no longer assets but a barrier to innovation and competitive success.[5]

The shift to a heightened role for software symbolized by IBM's restructuring may be happening early in the U.S. partly because of its competitive weakness in IT manufacturing as some Japanese say but regardless of the causes, there is reason to believe that it will happen everywhere. This is because hardware is increasingly a commodity product with low margins, as so many Japanese manufacturing firms are painfully aware. The reason for the diminishing value assigned to hardware, is that more and more of the value in high tech and other products and services is contributed by software.

Software contributes directly to competitive advantage through its use of an ever expanding array of value-added applications.[6] Software provides more and more functionality for hardware (through embedded software) and it enables "smart" enterprises (enterprise software). It does this through its expanding role in product design, enabling data aggregation, integration and analysis, providing data on detailed customer behavior and needs and, facilitating delivery of services. The locus where truly innovative high tech products are being developed is at the nexus of hardware, software, and services.

At its best, embedded and application software provide a stream of services throughout the life cycle of manufactured products. The traditional approach to product development focuses on the value of the core tangible product at the time it is delivered to the customer. When you bought a car in the past, you got access to all its capabilities at once. But products, using software, can release expanded value over the life cycle of the product. This can ultimately be much more important for total customer value. That is what has happened to the mobile phone. It is starting to happen to TVersus, and to automobiles; this list will only grow.

Software's role in delivering services and software as a service (SaaS) has seen huge increases in recent years. IBM's service revenues have grown from less than 30 percent of revenues to more than 50 percent from 1998 to 2008. Sun Microsystems, Hewlett-Packard, Cisco, and even Dell have all greatly increased their delivery of services, as they build on and modify their hardware past.[7] A Dell spokesmen reported that its software increasingly makes its storage technologies stand apart from competitors. He stated in early 2012 "hardware is becoming very commoditized. In [Dell] Storage we have more than quadrupled the size of our engineering team since our recent [software] acquisitions, so now 90% of our engineers are software developers." [8]

Japanese IT Firms' Decline

It is hard to even imagine that just some twenty odd years ago, Americans were being warned that Japan was positioned to become the next software superpower, following upon its success in manufacturing. Small fragmented Silicon Valley firms were seen as no match for the commercialization capabilities and financial resources of large integrated Japanese hi tech manufacturers. Moreover, the factory approach to software development, favored by large Japanese firms, was seen as superior to the craft approach dominant in the U.S.[9]

Notwithstanding these predictions, the last two decades has not been kind to Japanese IT firms. They have been steadily losing global share in major products. Richard Katz has documented these developments as follows: Japan's electronic products output produced in Japan, typically containing embedded software, declined 50% from 2000 to 2011. Electronic exports are down 37% in that same period. While the global image of Japanese electronics has been on consumer products, they are increasingly engaged in intermediate goods transactions producing parts and components for othe manufacturers. Consumer electronic components and devices accounted for 45% of production in 2000 and rose to 56% by 2011; in the same period, industrial components rose from 8 to 12%.[10] The problem is that the bulk of profits for these products typically go to sellers of final products, the Apples and Samsung's of the world, while parts and components suppliers, excepting those with strong IP, are most likely to be left with the crumbs. The combined market capitalization of Japanese electronics manufacturers was more than halved by the end of Dec. 2011 from its year 2000 level. Nor is this decline simply a reflection of the overall decline in market capitalization during this period; the first section (large firms) of the Tokyo Stock Exchange declined just 16% over the same period.[11]

The IT sector is more than electronics. IT products are included in the Information and Communication Technology (ICT) sector. Services accounted for 5% of total ICT value in 1996 but they rose to 22% of the total by 2009. Most of the value of ICT services is contributed by software. Yet, services accounted for a little over 2% of Japan's ICT exports in 1996, and fell to a little less than 2% by 2009.[12] Clearly, Japan is lagging in the hardware to services transition and a large part of the weakness can be attributed to software.

Finally, with respect to creating and employing state of the art enterprise software to create smart enterprises ready to take quick advantage of strategic opportunities, Japan is a "third mover" with its firms belatedly taking many of its cues from American firms. We see this in their slow and limited adoption of enterprise software (ERP) and even when adopted, fewer modules are implemented. In particular, in keeping with their commitment to maintaining their perceived unique monozukuri capabilities, the manufacturing module is seldom adopted.[13] The manufacturing module typically consists of: engineering, bill of materials, work orders, scheduling, capacity, workflow management, quality control, manufacturing process, manufacturing projects, manufacturing flow, and product life cycle management. Enterprise software systems integrate internal and external management of information across an entire organization, thereby enabling the firm not only to be managed more efficiently but also enabling information to be better used for strategic purposes. Analysis of comparative survey data from 2003 suggests that American firms are much more likely to use IT investment for strategic benefits (e.g., winning new customers, increase in sales, development of high added value of products/services) than Japanese firms which are more focused on extracting its operational benefits (e.g., cost reduction).[14]

Another area where Japanese risk averseness is apparent, is in corporate adoption of agile development methods. These methods are attractive because of their potential to provide continuous customer/client collaboration and feedback throughout the software development process and thereby deliver products more closely attuned to customer needs. They are characterized by an iterative approach with frequent "builds" which take changing requirements into account from the start and potentially enables higher quality. Forrester Research, based on surveys of U.S. developers in large and small organizations, reports that some 45% of U.S. software developers in 2010 could be classified as using agile development methods, broadly defined. The most frequently used aspects of agile were: short iterations, constant feedback, daily scrum meeting (stand up meetings which the team uses to share progress and make decisions), product owner, and continuous integration and build. Kenji Hiranabe, a leader of the agile movement in Japan, estimates that the acceptance rate is less than 20% among Japanese developers.[15] Particular concerns include a fear that time to market will increase and that quality will deteriorate. Promoters like the government agency, IPA, find it useful to use the euphemism "non-waterfall methods of software development" in order not to scare off possible adopters. Neither of these lags bode well for Japan taking a leadership role in a dynamic area where electronics/software/services are drivers of competitive success.

Japanese firms in IT and software services have almost no presence in global markets.[16] Even where Japanese IT firms have strong sales globally, their profit margins, already low compared to competitors, have fallen even lower in recent years, many now in the 3-4% range.[17] That means that less funds are available to them for future capital investment than is available to many of their foreign competitors with higher margins. To locate the internal sources of these problems, I turn to an examination of software innovation. There are both exogenous and endogenous sources of Japan's growing competitive problems in IT software. The focus in this article is on those endogenous sources arising from an assessment of Japan's software innovation capabilities.

U.S and Japanese IT Firms' Software Innovation

One can grasp the status of Japanese IT firms by examining the recent results from Arora, Branstetter and Drev's recent study of US and Japanese IT patents granted by the U.S. Patent Office (USPTO) from 1983-2004.[18] The authors examined U.S. and Japanese embedded, application, system and enterprise application software patents in the high tech sector. They found first that IT Innovation has become increasingly software intensive. Researchers first looked at hardware patents in the high tech sector from 1990-2003. They found that these hardware patents increasingly cite software patents. This is true in patents emanating from all countries including Japan. How big a change has there been? For the total dataset, an IT patent granted in 2003 was almost 3.2 times more likely to cite a software patent than one granted in 1990. This represents a remarkable shift over just 13 years suggesting the rapidly growing importance of software.[19]

Even though this trend is apparent in both Japanese and U.S. patents, the researchers found that Japanese IT hardware patents were systematically less likely to cite software patents over time than non-Japanese firms. Consistent with this finding, they found that Japanese firms file fewer software patents than their U.S. counterparts and that the difference has grown steadily since the late 1980s and especially after the mid-90s. Their findings indicate first that IT inventions have become much more software intensive over time. Second, U.S. firms have incorporated more software inventions into their products and services than have their Japanese counterparts. Even more telling, they find that U.S. firms have improved their innovation performance vis a vis Japanese firms in just those IT segments which are most software intensive (computers and peripherals). The researchers also find that within IT, Japanese firms are now disproportionately located in less software intensive sectors.[20]

Innovation performance was measured in a number of different ways , including patent productivity per dollar invested in R&D. Using these and other measures, they found that the software innovation performance of Japanese IT firms increasingly lags behind that of its U.S. counterparts, particularly in IT sectors which are more software intensive. They conclude that their findings are consistent with, and provide a new explanation for the "precipitous global decline" of Japan's IT industry in the 1990s.[21]

These findings are consistent with the data noted earlier that Japanese manufacturers focus more and more on the components business and are moving away from markets which demand total solutions and therefore, are likely to be more software intensive. This critical parts strategy represents a gamble that Japanese IT firms can dominate the global market for their hardware centric parts, thereby commanding a premium price. Yet, while some will undoubtedly succeed, most will likely end up as commodity suppliers in fierce competition with other Asian producers.

Sources of the Innovation Gap

The research reported above is, in itself, important. Moreover, it leads to a second question which is the primary focus of our analysis. Arora and colleagues discuss two possible sources for Japan's shortfall in software innovation. First they point to a constrained supply of software knowledge and skills (human capital) in Japan and second to the slowness of Japanese IT firms to recognize the transformational nature of software. They set up a test to differentiate between the two. They reasoned that if Japanese firms are constrained by their software human resources at home, then they will have the incentive to tap into foreign knowledge and expertise by setting up software intensive R&D facilities abroad (e.g., Silicon Valley) to further software innovation. The researchers further reasoned if the Japanese deficit in innovation performance was because the Japanese managers downplay the importance of software, then the research output of their overseas subsidiaries, like that of their domestic operations, ought also to be less software intensive than their American counterparts. Using the same dataset, they find support for their first hypothesis that Japanese firms, constrained domestically by their software human resources, further their software innovation in their U.S. subsidiaries; to wit, the share of software patents of total patents invented and patented in the U.S. by U.S. based Japanese firms is much higher than the share of software patents of total patents invented in Japan and patented in the U.S. by Japanese parent firms.

There are, however, problems with this test. The authors assume homogeneity in the patents Japanese file in the U.S. and Japan by relying on filings with the USPTO to measure both U.S. and Japanese software inventive activities.[22] Yet, it is more expensive to file in foreign jurisdictions, reflecting the costs of translation, retaining offshore legal talent, etc. Moreover, while standards for review, appeal of reviews, and issue of patents have formally been harmonized across high-income economies, differences in all of these persist, further raising the costs and uncertainty of offshore patent prosecution. Finally, software innovations, unlike in the United States, are not by themselves patentable in Japan. The Japan Patent Office states that information processing software is concretely realized by using hardware resources and when the software and hardware resources are working together so as to realize arithmetic operations or manipulations of information. Therefore, software can be patentable only if the information processing by the software is concretely realized working in concert with the hardware. The original specification needs to fully disclose the hardware arrangement of the apparatus to which the software is applied, the specific functions of the hardware, and the relationship between software and the hardware.[23] These requirements further distinguish Japanese corporate approaches to patenting software in the U.S. from their practices in Japan.

For all of these reasons, those conducting empirical analyses of patent data rarely assume that "home-base" and "offshore" patent applications are homogeneous." In line with this thinking, it would be wise to control narrowly for product characteristics. There appears to be a “home country” bias in that U.S. inventors have a home court advantage so that they file more patents in the USPTO compared to foreign applicants for a given set of inventions.[24] The authors do not consider whether there are alternative explanations for their findings. One might expect, for example, that Japanese firms likely select the IT patents they choose to develop in their U.S. subsidiaries and file, from among those with the most promising export/sales potential to the U.S and other global markets. This, not implausible scenario, may lead to the outcomes they found (Japan’s innovation shortfall), an outcome which may have very little to do with software human resource deficits in Japan.

There are still other problems with the software human resource deficit argument. In building their case, Arora and colleagues documented Japanese software innovation resource constraint deficits vis á vis the U.S. in terms of the differences between Japan and the U.S. in annual inflow of domestic IT graduates and immigrant inflows. They found that by 2001 the annual inflows of graduating IT professionals and H-1B visa entrants (including some hardware engineers) into U.S. firms had grown to almost 3 times those in the comparable categories in Japan. To be specific, the inflow in the U.S. was running about 165,000 a year versus roughly 58,000 in Japan. They also point out that if one took into further account offshoring, the resource gap between the two countries would be even larger. These differences, they believe, go a long ways to explaining Japan’s software innovation deficit vis á vis the U.S.

If one assumes that that human resource constraints are important factors limiting Japanese IT software innovation and that Japanese executives were aware of IT software innovation’s importance for subsequent competitive performance, economic theory would lead us to expect that they would address these resource constraints over time. Indeed, if past Japanese reactions to important factors contributing to deficits in competitive performance were any guide, we would see firms and government agencies like the Ministry of Economy, Trade and Industry (METI) and the Ministry of Education, Culture, Sports, Science and Technology (MEXT) making strong efforts to eliminate these deficits. As we shall see, their efforts have been less than impressive.

Possible avenues to closing the gap would be for firms to raise wages of Japanese software professionals relative to other technical specialties, and rapidly expand offshoring to match U.S. firms’ exploitation of this resource. Another option would be for firms to step up their own training activities. Such efforts would be especially expected in Japan where large firms have a strong history of developing their own core technologies and doing in-house training.

There has been a rapid increase in offshoring by the Japanese since 2001. Kojima and Kojima estimate their offshoring in 2004-2005 was roughly 1-2 billon dollars or roughly 6% that of the United States.[25] Official government estimates suggest that they doubled the value of their offshore investment from the period 2003-2004 to 2010.[26] This would still leave them well short of U.S. levels of offshoring, even if U.S. offshoring levels had not grown over this period which they surely did. Moreover, whatever success Japanese firms had in increasing their offshoring by 2010, they were still left with the huge gap in home base software human resources.

I used the stock of national software personnel, versus the annual inflows of IT university software and hardware related graduates plus H-1B visa entrants used by Arora et. al, to estimate this gap. Labor stock is arguably a better indicator of software resource capabilities, than labor inflow from domestic university graduates and H-1B visa since in both Japan and the U.S., IT positions are often filled by those without IT degrees.[27] Unfortunately I was unable to find labor stock data for 2001. Examining labor stock totals for 2010, however, putting aside offshoring deficits, the gap in human resource capabilities between the two IT sectors using this different measure remains relatively unchanged. The U.S continues to record roughly 3 times more software professionals than Japan.[28] See Table 3 below.

I found no evidence that software salaries rose sharply relative to other technical occupations or that universities rapidly expanded their capabilities to produce much larger numbers of high quality software engineers. Given the failure of the Japanese to diminish the large gap in domestic software resources relative to the U.S. suggests first that perhaps human resource constraints were not in themselves critical to the Japanese IT sectors' deficit in software innovation. .

Second, these findings are consistent with the alternate hypothesis rejected by Arora and colleagues. To wit, that Japanese executives and government officials have demonstrated over a long period a lack of full understanding of the growing importance of software for their firm's future competitive performance. The findings are also consistent with the view that as understandings of the importance of software did gradually develop, as it certainly has in recent years among the major Japanese IT companies, they found themselves in a bind. They don't know how to proceed without jeopardizing their existing hardware businesses. Many of these hardware businesses now manufacture in offshore locations, especially China. They generate low but consistent profits. They are built on manufacturing strengths associated with high quality precision manufacturing. What these products require from their embedded software is high quality (as measured in minimal if not zero defects) and customized software.

High quality is needed so as not to jeopardize the hard won high quality reputation of their manufactured products. Customized software is required to effectively and efficiently transfer the flow of design information from and to customers (the essence of their monozukuri). From this perspective, software is simply an assistant and aid to selling superior hardware. High quality customized software married to high quality precision manufacturing is seen as essential to maintaining current markets and revenue streams, however meager they often may be. [29]

Japanese manufacturing firms are having trouble reconciling these traditional objectives of high quality, customized software and short time to market as new software products start to invade their traditional hardware/embedded software products as occurred recently with car navigation systems. In this case, Android smart phones are being integrated into existing navigation systems based on hardware and embedded software. At the same time, the Japanese firms lack confidence that they will be able to replace these old advantages with new value creation activities arising from innovative software solutions. Rigid quality assurance practices are not conducive to software innovation. Thus, they have often chosen not to heavily invest in these solutions. It is in this context that many manufacturing firms don't see themselves as experiencing significant software human resource constraints. These themes will be explored further below.

One can pursue the human capital constraints explanation from another angle by examining the software labor stock relative to the total labor stock available to both U.S. and Japanese firms. As discussed earlier, the U.S. has approximately three times the stock of home base software engineers than does Japan. One can adjust these numbers to take into account labor force size. By doing so, we see that in 2010, the number of software engineers per 100,000 labor force members in Japan was roughly 1,500 versus the U.S. with 2,100. These adjustments reflect that Japan's total labor force is only 43% that of the U.S. and its GDP is only 37.5% that of the U.S. [30] Thus, adjusting for size of GDP and total labor stock, makes Japan look far less of an outlier relative to the U.S. with respect to its software human resources than suggested by Arora and colleagues. Nevertheless, large differences remain.

Still a numbers gap, while smaller, does persist. Arora and colleagues' analysis rests on the assumption that more software talent yields more innovation. Yet, more software engineers do not, in themselves, equal more productivity or innovation whether in a firm or an industry sector. [31] Large aggregations of software engineers whether in a firm, region or nation cannot be expected in itself to yield greater innovation. Innovation in the highest patent yielding regions in the U.S. (San Jose, San Francisco, New York and Los Angeles) benefit from powerful supporting ecosystems, from high quality engineers and from the embeddedness of these engineers in specific kinds of organizations whose objective is to create and market these innovations in horizontal markets. Put differently, Japan's innovation deficit vis a vis the U.S. may arise less from smaller numbers of engineers but from different organizational and managerial capabilities and especially from different strategic choices. It matters for software innovation whether most software engineers are in firms making one product for one customer or in firms aiming to sell millions of their product to many customers.

University IT Education:

Given the criticality of human capital to software development, we can partially evaluate a nation's recognition of the importance of software in terms of its timing and quality of investment in software education.[32] Steinmuller argues that the U.S. software industry's advantage over Japan lay in its first mover advantage, one that was enabled by government R&D policy and the early development of computer science (CS) education at the university level.[33] The latter advantage over Japan and many other nations, as measured in terms of the quality of U.S. CS education and research, persists to this day.

An examination of the formal education of software engineers in Japan and the U.S. reveals somewhat similar distributions when comparing high school, junior college, and university graduates. For example, 53% of Japanese software engineers (2007) are reported to be University graduates versus. 48% in the U.S. (2008).[34] The problem with these data is that they don't reveal what percentage of these graduates actually had any kind of formal IT education. It is a widely-held view among leading Japanese software specialists that well over half of software engineers have had no formal education in the field but acquired their skills through company training. This is particularly likely to be the case in the lower tiered IT service firms providing customized software.[35] Unfortunately, I have been unable to find comprehensive data documenting these common views. The number employed in software jobs in the U.S with no formal education in software, has also been large in the past.

Over time, however, those with formal software education have increased. As we shall see below, those with undergraduate CS degrees have increased rapidly in recent years in the U.S. but much less so in Japan and they have been in higher demand from industry than has been the case in Japan. Analysis of NSF data show that as of 2003, 85% of computer software engineers had at least one Science and Engineering degree, defined broadly as those with degrees in computer/math sciences, biological, environmental, and life sciences, physical sciences and social science and engineering. Of these, 41% had at least one degree in computer/information sciences and those whose highest degree was in computer/information sciences constituted 38%.[36]

It is especially revealing that some 20% of software engineers in the U.S. have a graduate school degree compared to 10% in Japan.[37] The gap in PhDs is even larger than suggested by this 10% difference. Until recently, the majority of Japanese PhDs in engineering were earned by industrial researchers in Japanese companies (ronbun hakase). These degrees are awarded by the employees' former university, typically after many years of R&D research. No coursework is necessary, only submission of a dissertation with some articles published in well-regarded journals. These PhD degree holders are very unlikely to be on the forefront of software innovation. These type PhDs are in sharp decline.

Putting aside the ronbun hakase PhDs, most CS (and most Japanese engineering PhDs for that matter) are earned by those aiming for an academic career. By contrast, most CS PhDs. from U.S. universities are hired by industry or government.[38] A great deal of new value in the private sector has been created in the U.S. by PhD and Masters of Science students and engineers. Some 22,000 PhDs have been granted in CS and Information Technology in the United States between 1978 and 2008.[39] Phd students at the best U.S. universities experience rigorous coursework and practicums and not a few go on to envision and then create and sustain a whole new world, e.g. UNIX, relational databases. Startups founded by those with PhDs include Adobe, Qualcom, Google, Sun, Cadence, Synopsys, VMware, and Symantec etc. Still other PhDs like Edgar Codd, with a PhD in CS from the University of Michigan, worked at IBM but nevertheless inspired startup firms like Oracle to develop commercial products building on his path breaking research on relational databases. This is where the Americans have excelled.

One is hard put to compile a comparable list of path breaking equivalents in Japan. It is not uncommon at major U.S. IT companies to see Chief Technology Officiers (CTOs) with PhDs. Moreover, a comparable list of companies founded by CS graduates with Masters of Science degrees could easily be compiled. To be sure, the Japanese system is changing: by the period 2007-2009, the dissertation only degrees declined to some 27 % of all information science and software dissertations but the pace of change is slow and the benefits of the new distribution yet to be revealed. Behind the limited numbers of Japanese PhDs in software related private sector jobs is the long term lack of demand by Japanese industry for engineering PhDs and the paucity of software startups.

There is further evidence of Japan’s lagging status from the 2012 Academic Rankings of World Universities conducted at Shanghai Jiao Tong University. In 2012, some 430 universities reported their data on students, academic staff and resources. The participating universities cover 61% of the Top 100 universities and 50% of other Top 500 universities on the Academic Ranking of World Universities 2012. The data were augmented by data for 1,200 universities including alumni and staff winning awards and medals, highly cited researchers, and papers in leading journals. The rankings for CS show that U.S. universities captured 20 of the top 24 rankings including the top 9. The only Japanese university in the top 200 universities in CS was the University of Tokyo which ranked between 150-200. [40] The foregoing rankings are hardly definitive. In part, they capture globalization of programs which is not equivalent to excellence per se. Weak English language capabilities and limited numbers of foreign students studying in Japan hold down the Japanese rankings. Nevertheless, the rankings do suggest that Japan is far from the forefront of CS and information technology education.

More telling has been the long slow process of incorporating software into the technical curricula of Japanese universities. It was well recognized by CS professors in Japan in the 1990s that CS faculties in Japanese universities were weak.[41] They relied heavily on U.S. created standards of IEEE-CS (Institute of Electrical and Electronics Engineers-CS) and ACM (Association of Computing Machinery), but with long delays.

The 1997 (J 97) standards published by the Information Processing Society of Japan (IPSJ) were greatly influenced by the U.S. "Computing Curricula 1991" of IEEE-CS and ACM.[42] Though these curricula were already outdated by the time they are were adopted in the U.S., not to speak of when they were adopted in Japan in 1997, they nevertheless lifted the modest profile of software in Japanese IT curricula.

Still later, a completely revised set of standards was completed in 2007 (J07); they were heavily influenced by the CC2001 of IEEE-CS and ACM. The timing of both Japan revisions reveals a significant lag from the initial American revisions. To be sure, the J07 standards, when published, incorporated CC2005, but the fact stands that the Japanese educational establishment has been a slow follower when adopting software curricula.[43] Given the rapidly changing hardware and software technology during these periods, this lag suggests a lack of appreciation by the Japanese educational establishment of the speed at which the technology was changing and the importance of quickly accommodating to these changes. Along similar lines, the IPSJ's accreditation committee criticized Japanese CS and other related curricula for giving priority to desk study while the American university curricula emphasized software design (exposing students to experiments and practicums). [44]

Critical to the incorporation and communication to students of up-to-date curricula is faculty with up-to-date knowledge. Here, Japanese universities were at a disadvantage. Much new IT knowledge was being developed in the U.S, especially with the advent of the PC era in the early 1980s. At the same time, Japanese firms which had achieved significant competitive success through modeling IBM's mainframe architecture, remained committed to building new generations of still larger mainframe computers.[45] This misplaced focus of the incumbents put them still further behind the U.S. in developing, integrating and deploying new software knowledge.

The expansion of IT departments in Japanese universities in the 1980s was associated with a shortage of CS-trained faculty. To fill positions, many universities hired “retired” IT executives from major companies, like NEC, Fujitsu, Hitachi, and Toshiba. It was and still is common for the Ministry of Education to push new departments in applied fields like technology and business to hire a portion of their faculty from among those with practical experience. These new academics helped fill out the faculty staffing in IT departments in the 1980s and 1990s. While they had practical experience in IT business issues and some technical matters, most of their corporate training and experience was in mainframe hardware and software. In the U.S. these were quickly being displaced by PCs, workstations, newer programming languages, networked systems, the Internet, and the like. Most of the Japanese faculty with these corporate backgrounds, were not competent to teach university students, much less the state of the art or the state of best practice. Moreover, most of those with corporate backgrounds were not capable of making major research contribution. This was at a time in which mostly American research scientists were transforming the field of CS. All this contributed further to the Japanese falling further behind the US and Europe in software during these formative years.[46]

The Education Ministry (MEXT) nominally plays a strong role in the development of university curricula for both private and public universities. Typically, they approve the plans for the curriculum of a new department and then monitor performance for some four years until the new students’ cohort graduate. In practice, the Ministry has relatively little content knowledge (despite having faculty committee advising the Ministry on proposals for new schools and departments and monitoring new schools and departments). It left devising content to university departments’ discretion. The Ministry’s monitoring role consists of insuring that the new department executes the curriculum plan according to what they promised they would do in their original proposal. In short, they perform typical bureaucratic scrutiny which insures consistency but limits flexibility and adaptation. Accordingly, while they were involved in the development of the 1997 standards, they were absent from the development of the new 2007 software curricula.

Professor Nobumasa Takahashi, a key player in the activities of the Information Processing Society of Japan, analyzed the national universities which typically set the direction on new technical curricul. He observed that the Japanese postwar university departmental structures were shaped by continual expansion in the fields of civil engineering, machinery, electrical engineering (EE) and chemistry. In many cases, new departments were formed from the Ministry of Education’s (MEXT’s predecessor) budgetary appropriations demanded by these existing departments. “As a result, the new departments were created with the strong coloration of colonies of the old departments.”[47] In 1998 at a university/industry meeting, industry people complained that the old disciplines were still casting a long shadow over the new disciplines of CS, software engineering and information systems.[48] The clear implication here is that entrepreneurial initiatives by the new software oriented departments were constrained by the old established engineering departments.

Other constraints on the new departments of informatics (roughly equivalent to what Americans call information science) and CS arose from an unanticipated interaction of two developments. First, in 1991, a de facto Ministry of Education deregulation of universities took place which no longer required science and engineering schools as well as other schools to offer liberal arts for the first two years of undergraduate education. As a result, many technical and science universities substituted engineering and science courses in their stead. As a consequence, a large number of redundant faculty were created at these institutions. Since they could not be fired, they needed to be relocated from their original departments. Second, at roughly, the same time, the Ministry, concerned about declining national birth rates, imposed limits on the expansion of university departments and faculties. They made a few exceptions, however, for growing fields of national importance, one of which was for information technology. The universities took advantage of this exception to "offload" some of their redundant faculty to these growing fields. Many of the transferred faculty continued to teach their existing specialties despite now being in CS departments.

The net result is that Prof. Masato Takeichi found that the total faculty (2,615) of the Japanese information science departments in Schools of Science and Engineering in 1998 were constituted roughly as follow: 30% were core CS faculty whose academic specialties were in CS and mathematical informatics. Viewing information science more broadly to include system informatics, electrical informatics and intelligent informatics added another 45%. A remarkable 25%, however, had educational specializations with little or nothing to do with CS.[49]The Ministry did nothing to combat this blatant misuse in the filling of the new faculty slots. Ideally, the new dynamic fields of informatics and CS were expected to make strong contributions to software education and to the economy as a whole. Instead, the Ministry actions and then inactions, at a formative stage in development of these new fields, led to the new departments being hamstrung and weighed down by excess and irrelevant faculty baggage contributed by redundant faculty from other fields and corporate retirees limited by obsolete mainframe software expertise. In turn, this meant that students had far less access to up-to- date knowledge in these rapidly evolving core fields. More subtly, it made it more difficult for CS to be recognized as a distinctive discipline.[50]

The latter problem continues to this day. Prof. Fumihiko Kimura, a member of the Science Council of Japan, at a public forum held at the University of Tokyo in March 2013, stated that CS (called johoukagaku in Engineering Schools) is still not recognized by most interested parties in Japan as a clearly defined discipline.[51] One measure of a new field's coalescence into a recognized discipline is the willingness of those in established disciplines to contribute to the emerging discipline. This both reflects existing assessments of the emerging discipline and bestows further legitimacy on the new field. In the case of the U.S., both theoretical and applied mathematicians played a key role in building the new CS discipline. Mathematical logic, the theorems of Turing and Godel, Boolean algebra for circuit design, and algorithms for solving equations and other classes of problems in mathematics played strong roles in the early development of the CS as a discipline.[52] Prof. Masato Takeichi, Dean of the Graduate School of Information Science and Technology at the University of Tokyo from 2004-2007, reports that mathematicians have not been as drawn to the new IT fields in Japan as was the case in the U.S. He attributes this to CS not being seen a distinct discipline in Japan.[53] Contributing to this weakness, Japanese CS researchers tend to focus on applied rather than basic research.

Indicative of the early recognition of CS in the U.S. was the formation of CS departments and sometimes equal partnership in joint EE&CS departments as at MIT and UC Berkeley. Stanford and UC Berkeley established their respective current CS departments and partnerships with EE in 1965. MIT recognized the growing CS activity within its EE Department and in 1975 changed its name to EECS. However, the Laboratory for CS (LCS) was founded in 1963. Originally known as Project MAC (Multiple Access Computing and Machine-Aided Cognition), its mission was to develop a computer system accessible to a large number of people, and to exploit the computer as an aid to research and education. LCS members and alumni have since been important contributors to the development of the ARPANET, the Internet, Ethernet, the World Wide Web, time-shared computers, RSA encryption and dozens of other technologies.

The contrast with Japan is striking. At the University of Tokyo, The Information Engineering Department was established in the Graduate School of Engineering was established in 1972 but the Department of Information and Communication Engineering for undergraduates was not established until 1991.[54] The Department of Information Science in the Faculty of Science (most comparable to CS departments in the U.S.) started in 1975 as an augmented successor to the Information Science Laboratory, and has since offered undergraduate and graduate degree programs.

CS still has relatively low status in the hierarchy of engineering related fields in Japan and their graduating students are not in high demand in industry. The most common explanation given for the reluctance of Japanese firms to hire CS graduates is that they do not have sufficient domain knowledge.[55] Given the history of Japanese manufacturing firms in providing domain specific training, it seems an odd explanation. Still another problem reported by knowledgeable software specialists is that the quality and quantity of Japanese high school students choosing to study software fields at the university level is declining. Students appear less interested in IT than in the past and the reason appears to arise from the structure of the industry with its heavy domestic outsourcing of customized software development and limited opportunities for employees to develop and capitalize on specialized skills.[56]

This description greatly contrasts with trends in the U.S. CS enrollments hit their peak in the year 2000 and then suffered a sharp decline after the dot.com collapse in 2001-2002, with the decline continuing through 2007. Most recently however, CS has exploded in popularity.[57] This growth is driven by perceived entrepreneurial opportunities and also by growth in overall corporate demand for the services of graduates of such programs.

The acceleration of enrollment growth in CS can be seen at Stanford University, one of the U.S.'s premier engineering schools. Those undergraduates graduating in EE (EE) total 50 a year and the number has been stable for some time but the number of those graduating with CS degrees has been on a sharp upward trajectory in recent years. In 2012, some 250 students graduated with CS majors, five times more than EE majors. Even more telling, CS is now the largest undergraduate major at Stanford across the whole university.[58] The same trend is evident at UC Berkeley. CS in the College of Letters and Science is the second fastest-growing large major. Its numbers more than doubled, from 138 to 357 (159 percent) from 2010 to 2013.[59] Moreover, its combined EE and CS Department in the Engineering College projects that its undergraduate upper division courses will have a ratio of roughly 3:1 in favor of CS for 2013-2014. [60] This acceleration of growth is in line with national data showing undergraduate enrollment in CS recovering from a low point in 2007 and then growing sharply for five years in a row from 2008 through 2012.[61]

At the leading U.S. engineering schools, those departments like CS, which experience sharp growth, drawing on a highly qualified applicant pool, have the discretion and strong incentives to grow their department. Faculty can be motivated to "put their brand on" a new generation of smart young people and they also use growth to claim greater departmental resources. The incentives for faculty growth are on display at MIT. In their combined EE (EE) and CS (CS) department in early 2012, the percentage split between EE and CS faculty was 54% to 46% in favor of EE. In Spring 2013, the combined faculty of this department voted to hire faculty at a ratio of 2:1 CS:EE for the next 3-4 years to address the shift toward CS in undergraduate enrollments while balancing this shift with the still greater number of graduate EE than CS students.[62] Central administrators typically are pleased with such growth and as seen in the MIT case, even in combined departments such as at MIT and Berkeley, there is seldom any successful effort by non-CS faculty to hinder adjustments to CS's growing student numbers. The historically decentralized education system in the United States means that the Department of Education, unlike its Japan counterpart, has been and is a non-actor on such matters. As expressed by David Hodges, former Dean of Engineering at UC Berkeley, "deference tends to be given to those faculty and departments which show leadership."[63] This captures the entrepreneurial spirit which underlies such rapid adaptation to new growth opportunities. This is not to say that such a decision making model doesn't have its downsides. In situations of declining job opportunities for graduates, departments and schools can be quite reluctant to cut back on admissions. The glut of recent law school graduates in the U.S is a case in point.

The contrast with Japanese universities is stark. We may use the University of Tokyo, Japan's leading university in CS and IT programs, as cases in point. In the engineering faculty, each undergraduate major, like information and communication engineering (not entirely equivalent to CS at an American university), has a prescribed student quota. Until the University deregulation of 2006, these quotas were rigidly prescribed by the education ministry, MEXT. Since the 2006 deregulation, enrollment limits became the formal responsibility of faculty. The faculty in question, however, is not that of specific engineering departments but rather the entire engineering faculty. Thus when discussion of allocation occurs, each department tries to protect its turf and not suffer losses at the expense of other departments. This makes it difficult to reach a consensus on changing the distribution of allotted student quotas. The quota for the departments within the Faculty of Engineering at the University of Tokyo has been changed only once since the enrollment limits became the responsibility of the faculty in 2006. In 2008, they changed in conjunction with a restructuring and consolidation of the various departments. Because of the restructuring, it is hard to say how the enrollment of information and communication technology students has fared. It is the case that the national universities, having been so long dependent on MEXT, find it hard to exert their independent will. They will often ask MEXT officials for advice before making changes.[64] Solving the problem by expanding the pie through increasing overall enrollments is a hard case to make in view of Japan's declining college population and inertial forces. In the case of the University of Tokyo, the admission quota as of May 2012 for electrical and electronics engineers (a combined total for juniors and seniors) was 150 while it was only 80 for information and communications technology.[65] This disparity persisted even after the adjustments of 2008 noted above.

In other words, the allocation of student quotas between EE and information and communications technology appears more suitable for two decades ago when IT and software, in particular, were not such global competitive forces . Instead, the current distribution supports traditional manufacturing (monozukuri) education in which software is subordinated to hardware. We also can see this distribution as again showing how the established disciplines continue to influence the field of information technology. To be sure, the EE education which Japanese students receive today, as is the case in the U.S., incorporates far more CS into their coursework than was the case in the past. Anecdotal evidence from UC Berkeley and the University of Tokyo, however, reveals a much stronger dose of required software courses in the UC Berkeley undergraduate EE curriculum than is the case with the University of Tokyo.

Most importantly, Japanese universities find themselves in a bind in that even if farsighted faculty believe in the need to rapidly shift toward CS and were able to overcome resistance from other departments, industry shows no great enthusiasm for hiring large numbers of their graduates at this time. Lastly, one must be careful in making too sweeping generalizations here since CS related programs and curricula appears in multiple schools and departments in many Japanese universities including the University of Tokyo. This is also the case in the U.S. Further research on this matter is needed.

In summary, there is ample evidence that the Japanese educational establishment was slow to incorporate strong software education into the curricula and that it still lags. They also failed to insure an up-to-date curricula was developed and executed by a large and capable faculty. Furthermore, growing the undergraduate CS student and related enrollments faces multiple challenges from declining student interest, and modest industry interest, to non- CS faculty resistance. Arguably, this situation contributed to human resource constraints as regards to developing software engineers with up-to-date knowledge who could make strong IT contributions to private and public sector firms. Yet, these human resource constraints, to the extent that they exist, can be better seen as a proximate cause of Japan's lagging software innovation. Our analysis suggests they operated more as an intervening variable. Some of the deeper causes can be found in the structure of the software industry –to be addressed below. Of critical importance was the Japanese educational establishment's failure to recognize and act on the importance of software. This, in turn, led to a failure to prioritize software education, and faculty software competence and to address weak student interest. Relevant also is the failure of leadership in the private sector to push for stronger IT education and to hire and train talented IT students.

METI Support for IT

We can also assess the slow recognition of software's importance through the actions of METI and its predecessor MITI. When MITI had large resources to co-invest in big projects in the 1980s like the Fifth Generation Computer Project and the Sigma Project, the big IT companies collaborated with them. Since then, however, METI has had no big resources to invest in IT projects. As a result, the big IT firms don't pay a great deal of attention to their activities and there is a widespread view among IT people that METI officials are poorly informed as to the real issues which need to be addressed. At the same time, it is widely acknowledged that the Japanese government and MITI and METI in particular relied heavily on the advice of Fujitsu, NEC and Hitachi (known among software people as FNH) in forming government IT policy over the last 30 years. They formed a kind of informal club where IT matters were discussed. It is highly probable under such circumstances that government policy was more oriented toward preserving the dominance of Japanese incumbents than to fully adapting to the new world of IT.

It is also notable that an outsider, IBM Japan, was not invited to join METI's Information Economy Committee until 2004.[66]

Yasunari Baba, Shinji Takai and Yuji Mizuta writing in the mid-1990s concluded that fundamental change was needed in current policies. They argued that this would require that MITI and other government organizations abandon their current premise that hardware production is more important than software production. They further point out that most Japanese government policies in the computer industry prioritize expansion of hardware production and sales while doing little to establish software as an independent sector. [67] METI did pursue an initiative in the 90's to support software startups but it was widely seen as a failure. As late as 2004, software related items accounted for only 5% of METI's and the Ministry of Internal Affairs and Communications' (MIC) combined 150 billion yen IT budget. By contrast, hardware related R&D (especially device related) took up 75% of their budgets.[68] To be sure, hardware projects require a great deal more capital than software projects; nevertheless the one-sided emphasis on hardware is hard to ignore.

There is robust evidence, in short, that Japanese government agencies woefully underestimated the importance of software throughout the 1990s and beyond for the emerging competitive environment. Eguchi Junichi, Director of Information Services Industry Division at METI acknowledged as much in an March 2013 interview.[69] Even as METI came to recognize the growing importance of software from the early 2000's to the present, it was, and is, often seen in the context of (embedded) software as an attachment or assistant to Japan's great physical (monozukuri) products rather than embedded and application software as potential drivers of competitive success.[70] METI's primary instrument for implementing its policy goals in IT is its subsidiary, the Information-Technology Promotion Agency (IPA). Its 2013 plan for IPA's Software Engineering Center (SEC) focuses on collecting defect data from industry and establishing a mechanism for suppliers to be accountable for their products. Again, we see a play to Japan's strengths rather than working on addressing the bigger issues confronting Japanese IT. The rise of "Big Data" has received a great deal of attention in Japan and METI published a 2008 report on "How to Use Big Data." One would think that the era of big data would finally be the vehicle for convincing government, educational and corporate leaders that software has a vital role to play in promoting competiveness. But thus far, its greatest use among Japanese manufacturing firms appears to have been using masses of internal data to optimize operational performance. As such, it reinforces their hardware mentality. As yet, there are far fewer success stories of manufacturing firms collecting and analyzing customer "Big Data" to bring about competitive breakthroughs.

An Institutional and Structural Perspective

The foregoing analysis suggests there are deeper structural causes of Japan's software innovation shortfall both in the period investigated by Arora and colleagues as well as subsequently. As one Japan software veteran reported to me,

> We did have labor shortage around the 2000 year period. The problem currently, is not an overall labor shortage but rather: Not enough Japanese products are focused on software, and not enough Japanese software professionals are focused on product. Both of these lead to a lack of demand for excellent software architects/designers. Taken together, this situation results in not enough innovation.

In other words, there continues to be some human resource constraints relating to specific skills. The Japanese IT sector's lag in software innovation, however, may arise less from human resource constraints per se than it does from a weak corporate demand and incentives for creating innovative software products. If firms aren't pushing to create such products, then they don't believe they have a need to hire, train, and use high talent software personnel.

One piece of evidence in support of this hypothesis can be gleaned from comparative data on IT investment. As shown in Table 1. These data show the percentage of IT investment to annual revenue in private sector firms. They reveal that the annualized percentage for Japanese industry overall in 2007 was 1.03. The comparable percentages in 2008 for the U.S. was 4.3, for the EU 3.0 and for the Asia/Pacific region (excluding Japan) 2.9. If U.S. firms are spending more than 4 times of their annual revenue on IT than the Japanese, and they have a strong ecosystem supporting such innovation, it would likely be the case that they would be generating more software innovation than their Japanese counterparts. Unfortunately, there is no data broken out for the IT sector but for manufacturing assembly industries U.S. firms are spending 6 times more than their Japanese counterparts and for manufacturing process industries 5 times more. Only in banking, insurance and the securities sector in the 2007-8 period were Japanese firms investing a higher percentage of their annual revenue on IT than the Americans. Notable also is that Japanese industries overall fall well behind the EU countries and the rest of the Asia/Pacific region on this measure. These data are not drawn from a representative sample so they should be taken as only a rough guide.

Nevertheless, Japanese private sector firms appear to be spending less of their annual revenue on IT. Therefore, they require fewer IT human resources and their lower level of IT human resources might be reasonably well matched with these lower levels of IT investment. Therefore, contrary to Arora et.al., they are likely to suffer only modest IT human resources constraints.

Table 1.
Percentage IT Investment of Annual Revenue by Industry Sector and by Country and Region: 2007-2008*

Country & Region	Japan	U.S.	EU	Asia Pacific
Percentage IT Invest. of Annual Revenue: All Industry Sectors	1.03	4.31	3.04	2.92
(N)	556	284	405	463
Industry Sector				
Mfg. (Process)	0.69	3.55	2.10	3.09
(N)	144	7	35	30
Mfg. (Assembly)	0.75	4.60	2.68	1.77
(N)	91	33	33	29
Retail/Hotels/Real Estate/Wholesale	0.52	2.73	2.76	1.54
(N)	133	26	47	53
Banks/Insurance/ Securities	5.89	5.08	3.79	2.15
(N)	29	42	42	43
Info and Comm. Services	2.92	4.54	3.10	4.62
(N)	11	45	49	59
Services	1.05	3.33	3.36	3.12
(N)	87	48	78	92
Agric./Mining/ Construction	0.65	1.26	2.08	2.16
(N)	52	8	19	38
Other Non-mfg.	0.15	6.25	1.82	1.42
(N)	3	9	30	24
Non-profit/Edu./ Medical Care	0.95	5.13	4.12	3.88
(N)	6	61	63	88

Source: Gartner, Japan; Approval no. GJ 13141
*** Survey dates: Japan: 5/2008; other regions: 11-12/2007.**

We can dig deeper into IT investment differences by examining the composition of IT investment. Table 2 compares the distribution of software investment in the U.S. and Japan by type of investment. The Japanese figures from the year 2000 show a heavy reliance on outsourcing (70%) but it is outsourcing primarily to the firms' domestic network of IT subsidiaries and their tiered set of supplier firms, sequentially involved in the software development process. The Japanese refer to this as "multiple step ordering" (tadankai hacchuu). Thus, Arora and colleagues are correct that Japanese product firms do far less foreign outsourcing to meet their software development needs than do U.S. firms. Yet, as just noted, large Japanese firms, with their thin employment of software professionals, do an enormous amount of outsourcing to a deep well of domestic subsidiaries, vendors and subcontractors.

As also shown in Table 2, in-house development for a firm's own use is just a modest 20%. Together these two customized development types account for some 90% of all Japanese software investment with packaged software left with just 10%. Nobuyuki Yajima, a long term software analyst, writing in 2012, believes that the Japanese data have not changed all that much since the year 2000, with the exception that there has been a rise in the use of packaged software . He estimates it is closer to 15% in recent years.

Table 2

U.S. and Japanese Software Investment by Type

(in percent)

	Packaged	Outsourced (customized)	In-House for own use (customized)
Japan*	10	70	20
United States**	29	34	37

*N . Yajima, 2012. "Management's Common Sense: Japan has Others Make Software, U.S. Makes it Themselves," (Keiei joushiki: sofuto o tanin tsukuraseru Nihon, jibun de tsukuru beikoku), *Nikkei Business Online*, (Oct. 12):1-7.

**Bureau of Economic Analysis, Dept. of Commerce, 2012. Table 11 Business (Private) Software Investment and Prices, http://search.bea.gov/search?query=software+investment+2011&commit=Go&utf8=%E2%9C%93&affiliate=u.s.bureauofeconomicanalysis. Accessed Nov. 10. 2012

The U.S. figures for 2010 reveal a strong contrast with the Japanese distribution. They show a relatively even balance between investments in packaged software, outsourcing and investment in in-house software intended for the firm's own use. Notable also is that in-house investment for own use software development accounts for almost twice as much of U.S. private sector investment compared to Japan (37% versus 20%). Important also is the gap in the importance of software investment in packaged software (29% in the U.S. versus just 10-15% in Japan). An examination of U.S. software investment in the early 2000s show the distribution among packaged, outsourced and in-house investment is virtually the same as that shown in the 2010 figures. These differences between the two economies reported here are non-trivial.

Given these differences in composition of software investment, we would expect that the distribution of software engineers would be strikingly different in the two economies as well. More specifically, we would expect that U.S. private sector firms would be more heavily staffed with software development personnel, many of whom are likely engaged in developing innovative software to enhance the strategic interests of the firm.

Table 3 shows the respective distributions. Some 75% of Japanese IT engineers are located at software vendor firms (e.g., large independent IT firms like Fujitsu and NTT DATA), system integrators-some of which are subsidiaries of large corporations, and tiered subcontract firms. As we shall show in later discussion, at most of these firms, the opportunities for creating innovative software products for sale to large numbers of customers is very limited. The priority of software vendors is meeting cost and, delivery and quality specifications. Vendors have few incentives to provide solutions with the latest system architecture and product innovation overall is a low priority. Indeed, it is not uncommon that projects led by top vendors are actually carried out by many layers of subcontractors with little involvement by the prime vendor.[71]

Table 3

Employment Location of IT Engineers

	U.S. %	Japan %
No. of IT Engineers at User Firms*	2,362,300 (72)	254,700 (25)
No. of IT Engineers at Software Vendors**	941,400 (28)	771,400 (75)
Total IT Engineers Employed	3,303,700 (100)	1,026,100 (100)

*User firms are those firms using internally or selling IT systems which they develop or have had vendors develop for them,.

**Software vendors are often called Internet service firms in Japan.

Source: Information Technology Promotion Agency (IPA), 2012. "Survey of Factors for Dissemination and Expansion of the Areas of Adoption of Non-Waterfall Model Development," (Hi Wotaforu Kata Kaihatsu no Fukyuu Yoin to Tekiyou Ryouiki no Kakudai ni Kansuru Chosa), IPA :14, 21.

Only 25% of Japanese IT engineers are estimated to be located at user firms. This means that most Japanese corporations are thinly staffed with high level software engineers. They are highly dependent on Internet service provider firms (vendors of various kinds and sizes) for meeting their IT needs. By contrast, Table 3 reports that in the U.S. some 72% of IT engineers are located in user firms and thus suggest that those firms employing these IT personnel are more likely to engage in developing their own software for internal use.

When discussing the large number of U.S. IT engineers employed in user firms developing their own software. It is important to capture differences in software startups. For truly innovative software, large U.S. technology firms still rely on software startups which attract the most creative talent. Large publicly listed firms are typically unwilling to tolerate the massive costs arising from learning failures associated with developing breakthrough software technologies. Instead, it is left to U.S. venture capital to fund these risky new technologies. Software firms get a very large portion of total venture capital investments in the U.S.

In 1989, .04 percent of total venture capital investment in Japan went to software startups; this compared to 11% in the U.S. at the same time.[72] By 2011, total venture capital investment in the U.S. totaled some 29 billion dollars while the sums invested in Japan were roughly 12% of that total. Computer Software startups investment accounted for some 24% of venture capital funded startups in the U.S. and IT investment, more broadly, accounted for 57% of all US venture capital investment. For Japan, software startups accounted for just 9% of venture capital startups.[73] So not only has there been a much lower percentage of venture capital funded investments devoted to developing Japanese software but total venture capital investment was far less in Japan as well. Most importantly, this much larger number of software personnel in the U.S. venture capital software startups were employed seeking to create new commercial products. Given the relatively low costs of software startups, one might hypothesize that the percentage of software startups among all startups would be higher than among only venture capital funded startups. Teikoku Data Bank reports on all annual startups which are 100% independent, have operating revenues or employees, and which have entered their database by requesting credit or requesting a credit check on someone. Of the 5,292 firms which qualified from 2005 through 2011, only 7% were software firms.[74] In short, the much smaller number of software startup firms in Japan both contributes to the weakness of Japan's software innovation capabilities and deprives large Japanese technology firms of their potential products.

Moreover, even the existing software startups face a large barrier. Large Japanese IT firms have a history of showing little interest in domestic startups, whether hardware or software. They have been accustomed to look for innovation in their existing network of alliances. Large IT firms tend to see partnering with unproven startups as too risky and often are only willing to purchase from or partner with existing firms with which they have exclusive subcontracting ties. Startups often are not seen as reliable and legitimate partners and their technology therefore suspect. Further, established firms are often not willing to do the due diligence necessary to separate the most promising from the less promising. While this reaction of established firms is not unique to Japan, it appears to be unusually rigid.

We can see the difference from the U.S. in the fact that many successful Japanese startups in the past have had to turn to foreign markets (typically the U.S.) to find their first customers to prove themselves before being taken seriously by large Japanese firms. Four prominent IT examples from Kyoto are Samco, Kyocera, and Nidec and Horiba. The founder of Horiba Ltd. recalls that Japanese potential customers were more interested in the amount of his new firm's capitalization and its number of employees than they were in its technology.[75]

In the U.S., the large technology firms access new software technology often by acquiring the successful startups or learning how to partner with them. In 2009, Intel acquired Wind River for almost 1 billion dollars. It is a global embedded software specialist in software device optimization. Wind River's list of clients includes Apple, HP, Boeing, Motorola, NASA, Mitsubishi, Siemens, and Samsung. So partnering or M&A is a possible way forward given the continuing difficulty large firms in Japan & U.S. have in recruiting & managing the most creative software talents.

In summary, software startups occur with far less frequency in Japan than in the U.S. The net results is that the U.S. has a vibrant independent software industry compared to the relative weakness of this sector in Japan. Software engineers working in software product firms, are by definition more likely to be oriented towards software innovation as a product and creating new business opportunities. Those in software service firms, where an overwhelming majority of Japanese IT professionals work, are much more likely to be engaged in installing, customizing, maintaining and upgrading existing enterprise software systems. The CEO of Infosys Technologies, the Indian outsourcing giant, estimated in 2010 that up to 80% of annual IT spending in Japan goes toward maintaining and operating existing systems compared with no more than 60% in the U.S.[76] Mr. Shintani at IPA , in leading corporate IT projects, reports similar findings from his work with corporate IT planning managers.[77] The upshot is that there is far less left in the average Japanese IT budget than in the average U.S. corporate IT budget for funding innovative software projects. These observations apply to enterprise and application software budgets of Management Information Systems (MIS) departments; embedded software budgets and specialists are more typically located in product development departments.

One Japanese industry veteran talks about the extreme pressure for high quality from business customers on system integrators, contractors and subcontractors. This leads to a quest for perfectionism as manifested in those lower down in the supply chain devoting meticulous efforts to the cosmetic detail of system screens, minor improvements in usability adding cost and a push for zero bugs instead of business process innovation.[78] The end result may be higher quality as measured by higher reliability and fewer bugs but it often comes with increased costs. Prior research does suggest that Japanese software developers score high on quality relative to their global competitors.[79] Yet, in the rapidly changing world of IT, high reliability and few bugs are an advantage only when combined with low costs, speed and strong innovation. But a rigid emphasis on delivering no bugs does not appear compatible with innovation. As Michael Cusumano points out, software innovation requires experimentation and trial and error.[80] User firms demanding software perfection from their suppliers display strong risk aversion. One can contrast this Japanese approach with Facebook's slogan, "Done is Better than Perfect" or "the good enough" solutions adopted by many of their foreign competitors. Of course, when the project involves mission critical systems, then the Japanese approach may well be the preferred approach.

Further disconnects take place because large business customers, with their typically thin supply of software engineers, are heavily dependent on their system integrators to tell them what they need, with the system integrators responsible for conceiving, carrying out software projects, and delivering a final business solution.[81]

The high dependency of large firms on their system integrators, as well as the lack of strategic importance many firms give to IT, can be seen in the low proportion of large Japanese firms with full time Chief Information Officers (CIOs). The Mitsubishi Research Institute's software specialists estimate that only 30-40% of large Japanese firms had full time CIOs in 2008.[82] It would be rare to find US firms with annual revenues over $500 million which did not have a fulltime CIO in 2008. A later survey by the Japan Users Association of Information Systems reports that over 50% of large Japanese firms have someone with the title of CIOs but the amount of time they spend on IT work is little more than 10%.[83] Just having a CIO work full time on IT, of course, is no guarantee that they are performing strategic roles for the firm.[84] The absence of the CIO role, however, suggests that large Japanese firms are making far fewer efforts in this regard and that CIO work itself is regarded as non-strategic low level. This situation appears to reflects top management's long term lack of recognition of IT's importance One can reason that if top management understood the importance of IT better, their policies would insure a full time CIO and this position and it would carry considerable weight. Indeed, this appears to be the view of many in the Japanese IT community. With this background in mind, we see that weak software innovation in Japan relative to the U.S. arises less from a weakness in software human resources per se than it does from structural differences, which institutionalize a maldistribution of software engineers.

This maldistribution grows, to a large degree, out of the long standing practice among Japanese firms to heavily customize their software purchases. As shown in Table 2, some 70% of Japanese software development investment is outsourced to other firms, compared to 34% for the U.S. If we include in-house development of software, just over 15% of the value of the Japanese software industry is accounted for by software products while 83% is accounted for my customized development of software.[85] This is important because of the high costs associated with customization, maintenance and upgrades, much of which is in the form of huge human capital requirements.

The Mechanics of Customization

To understand what is driving the structural differences discussed in the previous section, we turn to examining this strong propensity of Japanese firms to heavily customize their software solutions. This leads us to analyze the nature of work and product flow associated with customization and how they are distributed organizationally. This, in turn, will tell us a great deal about their consequences for patentable software innovations and overall competitive activities. What follows is a prototypical example of the organizational structure, practices and rationale of the "Big Five" Japanese electronics manufacturers which have led the enterprise software business in Japan. It is based on our field observations.[86] The players are: Hitachi, NEC, Fujitsu, Mitsubishi Electric and Toshiba and their system integrator subsidiaries. This example will not resemble exactly any one of the five firms but rather presents a composite picture. This is less a problem than it might otherwise be because major Japanese software firms tend to resemble one another in structure and practices. As in many Japanese industries, the leading software firms have a full range of matched product offerings. The extensive use of the term “sogo” (general or full line) in Japanese captures these strong tendencies.[87] That is to say, there is relatively modest specialization at the firm level of software product offerings among the Big Five. Such comparable coverage by competitors is not absent in the U.S. in such fields as human resources and financial software where regulatory requirements induce a similarity in offerings but the phenomenon is more pervasive in Japanese industry. It is important to note that most recently Hitachi, Toshiba and Mitsubishi Denki have announced that their primary business focus going forward will be large social infrastructure projects (energy, transportation and utilities). For these projects, these firms see software more a glue holding different project components together rather than as primary drivers.

We can begin our treatment of the prototypical Company X with a discussion of its Product Group A. Group A is a software "product center" for the system integration business. Although it is often formally organized and called a business unit, its “products” (such as database management software, application development tools, and job management software) are supposed to be used only in Company X’s system integration projects. Group A is ostensibly a profit center, but in reality it is managed as a cost center of the system integration business unit to which it reports.

The rationale for the above mentioned reporting relationship is that the enterprise software developed by Company X is plugged into large complex custom solutions for specific customers. These inputs are coordinated and combined by the system integration business unit. The solutions in question are typically vertically integrated solutions (e.g., Operating System + Database + Applications + Hardware). In this sense, the business application “product” developed by Group A is just a component (independent module) in this larger complex solution. Nevertheless, they are referred to as products because “products” reference critical components of the system to be developed that have been already “productized.” To say that they have already been “productized” means that they have already been operationalized and used elsewhere and consequently there will be shorter development time and lower cost for building that module. In addition, these components are referred to as products because the firm wants to have the same set of “products” that IBM and other competitors provide. They also want to follow current fashion by saying, for example, that “our version 1.xx supports the Service Oriented Architecture concept.”

Put differently, to make an ideal product at low cost in this “software factory” approach, firms avail themselves of commonly used software parts as independent modules which they call “products.” These modules are then deployed for use in multiple end user products to provide economies of scope. It is critical, that these components are smoothly integrated with the larger custom solution.

The contrast with IBM is quite striking. IBM's software group has brought to market many innovative packaged software products, tools and middleware that it passes over to its Global Business Services (GBS) group to be used as a factor input to which GBS adds a range of value added services associated with their customization work for end users. These products, like its DB2 database, compete directly with other enterprise software firm products such as those from Oracle and Microsoft. Yet, the GBS group also partners with these same enterprise software companies as part of its public commitment to provide customers the best possible solution to meet their needs. That is to say, IBM's GBS unit offers competitor, as well as IBM products, to end users in their effort to deliver to customers the best solution. As a result, the software group cannot be sure of sales from global services unless they have cutting edge products.

Company X's Group A, on the other hand, is primarily oriented toward meeting the needs of its System Integration Business Unit which, in turn, is responsible for integrating the total complex custom solution for a specific customer. The System Integration Business Unit is their primary and sometimes exclusive customer. As a result, business applications, which in the U.S. may be tradable stand-alone products such as an order entry module in an ERP system or application development software tools or even whole ERP and Database systems, tend to be seen quite differently in Japan. For the Japanese firms producing them, they typically are non-tradable components aimed at contributing to the whole solution package which will be sold to the end user. The system integration business unit provides a vertically integrated solution. As a consequence, the "products" created by Group A, customized as they are for a specific end user, typically are not marketed to other customers. Indeed, Group A doesn't have a strong marketing arm nor an aggressive sales force aimed at the external market.

Software groups, operating in this fashion, have few incentives to patent innovative software to be sold on the open market. This is because many of the enterprise software related innovations are unique to a particular Japanese customer's product, thereby reducing the incentive to patent in the US or Japan. As implicit in our previous discussion of Table 3, the bulk of young person entering into software jobs are assigned to work on customizing enterprise software. There is not a lot of room here for creative activities and thus these young people may lose their enthusiasm for IT and worse yet, their experiences. as we shall see, get conveyed to the next generation of students.

There is also a software Group B at Company X. Group B's revenues are (typically) much smaller than Group A's. Group B is a true software product business unit, and operates as a profit center with relative autonomy. In some system integration firms, Groups A and B are in the same business unit. While a profit center, Group B does not, however, market its products to the external market. Rather it markets its products, enterprise software, to the Group As of other major system integrators. The unstated assumption is that it is futile to try to sell directly to end user corporations. This is in striking contrast to much software sales activity in the U.S. Group B does have some marketing and sales capabilities but they are not strong. By contrast, U.S. companies like Oracle making innovative business applications, have succeeded in the U.S. and Japan by prioritizing the sales function.

The huge challenge facing Group B is that they are trying to sell their products to system integrators at other companies which typically offer a full range of their own Company X products. So unless Group B can develop a truly breakthrough product or greatly improve an existing one, it is very hard to successfully sell product to other system integrators. These other system integrators are their potential customers but these customers have every incentive to use their own products for their own end user customers. Moreover, even if Group B can develop a successful product, the other system integrators will quickly act to "cover its hit." They also will do everything they can to persuade the end user to buy a related product from them, even if its functionality is more limited, or to wait until their own equivalent product becomes available.

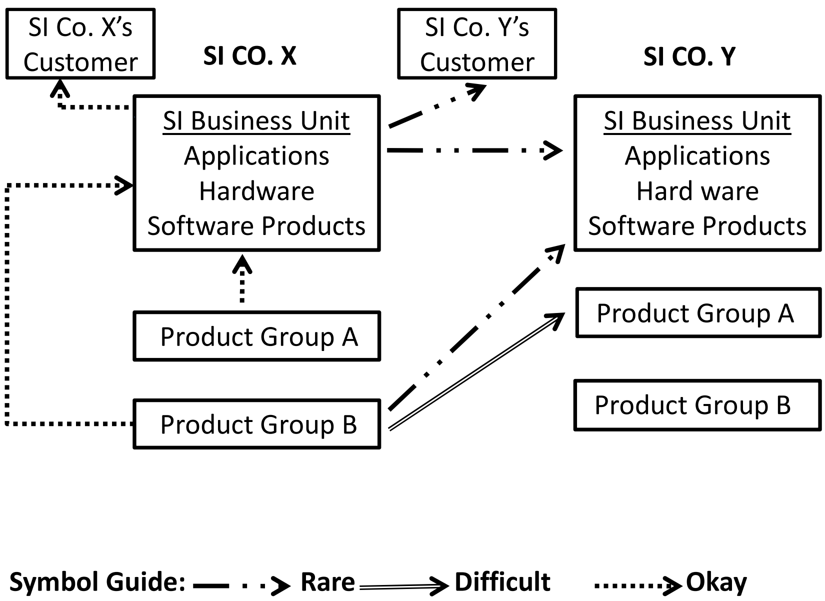
This brings us to an interesting feature of this system from a Western perspective. The pivotal role of system integrators in Japan, is such that software vendors rarely succeed, as discussed above, in marketing enterprise software directly to large corporate end users. While successful cases do occur, the end user corporations are usually heavily dependent on their long term system integrator partners to guide them in their IT selection choices for reasons discussed above.

Because most Japanese top managers don't have confidence in their ability to evaluate IT choices, they rely on their system integrator. The system integrators know the details of their business processes through their long term relationship, and can guarantee a smooth implementation process without disruptions.

In summary, we can say that in Japan, customization work is the central activity of system integration firms done in collaboration with in-house teams and in some cases an external software vendor. More than in the U.S., there is an overwhelming bias among corporate managers that large scale business applications like ERP will require extensive customization, despite the added costs this imposes. From a Japanese system integrator vendor perspective, the ultimate form of a product (hardware or software) is one which is fully customized for a specific customer and that is exactly what customers have come to expect and demand. Software producers, inextricably tied to hardware firms, seem to have fully internalized the user mentality epitomized by Japanese hardware managers.

Figure 1 summarizes the relationships we have described above in terms of the possibility of sales results. System Integrator Firm X represents the prototypical system integration firm. Its Product Group A is focused primarily on supplying software components to its System Integration Business Unit which in turn integrates them into complete solutions for its customers. In relatively rare cases, efforts are made to sell directly to Company Y's system integration unit. Success is not common but when successful, Company X's system integration unit proceeds as follows. Instead of making direct sales efforts to Y's system integration business unit, it rather tries to influence Y's end user customers to use enterprise application z.

Figure 1
Software Sales Between
System Integrator (SI) Firms X & Y
and Their Customers



If successful, the end users come to prefer it and tell System Integrator Y to use it in their system integration projects; the sale itself is to System Integrator Y. A much smaller Product Group B mostly sells its software products to Company X's system integration business unit but it also tries to accomplish the difficult task of selling its outstanding products to other Product Group As at other system integrators.

We may add a few additional observations regarding customization of enterprise software from an American perspective. As noted above, customizing a business package like ERP is often very expensive and complicated. Customization usually involves altering the code to meet a specific company need. Some business packages have very generic features, such that a great deal of customization occurs in most implementations. Alternatively, all U.S. enterprise products, like Oracle's ERP, now have increased configuration options which eliminates the need for customization in some areas. Rather than changing the source code, exercising configuration options is more about settings tabs or parameters to meet specific customer needs such as setting up of payment terms or the length of the customer record.

Creation of a best practice capability embedded in enterprise software systems is based on modeling specific business processes among "lead users" and then creating a standardized best practice model for such things as order entry systems. A common view among U.S. software vendors is that customization is a main cause of cost overruns in software development and leads to increased costs in long-term maintenance and upgrades. In this view, all too often customization is made to satisfy a perceived need for company-specific functionality, despite that a more than adequate method/process already exists in the standard software routines. Indeed, a number of researchers in the U.S. cite avoiding customization as one of the critical success factors in successful ERP implementation.[88] While long term relationships with system integrators certainly are in evidence in the U.S., it is also the case that customization work is often undertaken on a time and materials basis rather than as an activity subsumed under a long term partnership.

Much research over the last decade documents the declining significance of keiretsu in the Japanese economy, using as outcomes measuring the diminished role of interlocking stock ownership and board of directors.[89] In the world of enterprise software, however, vertical keiretsu networks continue to powerfully shape trading relationships among enterprises and the nature of competitive outcomes. While undoubtedly, innovation takes place under these arrangements, firms appear to have less incentive to patent their software solutions.

In the past, exogenous shocks from American entries and products, have had shaken up existing Japanese software markets. Salesforce.com's aggressive entry into the Japanese market, may have that effect on the enterprise software market.. They were able to recruit a top level Japanese executive who opened doors for them and they secured a prestigious contract from Japan Post. They also did a large project for METI and thus got a big PR boost. [90] Salesforce.com promotes software as a service as the model to replace traditional enterprise software. Software as a service (SaaS) or "on-demand software" is a software delivery model in which software and associated data are centrally hosted on the cloud. To accommodate the Japanese market, Salesforce.com cleverly has been willing to provide a fair degree of customization to their Japanese clients. Unlike in the U.S. they also targeted large companies first and have had notable successes. Salesforce.com's largest market outside the US is already

Japan (led by CRM, Enterprise Software and other products). Should they continue to take market share, the SaaS model could radically disrupt the existing enterprise software market. This might further a more rational reallocation of software resources.

Path Dependency and Hardware Centricity

We have seen the problems with the human resource constraint explanation for Japan's innovation shortfall. The alternative hypothesis posited and then rejected by Arora and colleagues is that Japanese IT managers failed to appreciate and respond to the growing importance of software. In so doing, they delayed in building up their software capabilities. In support of this hypothesis, we suggest that the origins of Japanese IT management's delayed response toward software lay in the structural issues discussed above and in their hardware centric histories.

We can take a page from Arthur Stinchcombe in his seminal 1965 contribution. He observed that organizations which are founded at a particular time and place must of necessity construct their social systems with the social resources available. This includes, of course, the human capital seen as necessary to enable organizational growth and survival. His particular insight was that once established, an organizations of a given type, tends to retain many of the basic characteristics it assumed at the time of its founding. That is to say, key organizational characteristics are imprinted so that the nature of incentives used to drive performance and the characteristics of those initially recruited are often sustained. Stinchcombe stressed the importance of understanding what enables those founding characteristics to persist over long periods of time. The answer lies in the path dependency of organizational and technological learning. A firm's prior organizational routines, norms and corporate culture (its history), often reproduced over long periods, constrains its future behavior. This is partly because opportunities and incentives for learning often will be close to previous capabilities. The more a firm uses a technology, the better it get at that technology with cumulative sunk costs in human and physical capital. If a firm succeeds over time using a given technology, dynamic increasing returns may cement its commitment to that technology.[91]

One can apply this understanding to the hardware centric origins and sustained commitment to their technology on the part of established Japanese IT manufacturing firms. Their very success reinforced these capabilities over long period of times. Electronic engineers were dominant in the founding of IT electronics firms. This occupational specialty dominated the management of IT firms throughout its history. It is plausible that this history continues to constrain their transition to software.

As noted earlier, skilled human capital remains the most crucial factor for software innovation.[92] The quantity and quality and deployment of such skilled human capital is decisive.

Sony has been one of Japan's quintessential hardware centric companies. The Walkman epitomized Sony's technical capabilities; it was a triumph of electronic engineering. Since then, a major portion of Sony's innovations have involved making products, smaller, thinner and lighter, outcomes achieved largely through creative hardware design. They, however, have been particularly weak in developing innovative application software.[93] Its Play Station is the rare exception. Howard Stringer, former President of Sony, in a 2006 interview with the Wall St. Journal provides a revealing insight into what has been one of Japan's premier firms. The following quote speaks specifically to the issue of human capital deployment.

> We did not bring software engineers into product development at the beginning. The hardware engineers would begin the product and then software would come in after the fact. And that's because in a company that has jobs for life, the older people are at the top and the younger software engineers are on the bottom, pushing up. So there is a kind of a generation gap.[94]

Stringer explains the late entry of software engineers into the product development process in terms of the Japanese age and seniority promotion system (nenkou joretsu seido). Yet, it is not just that older hardware engineers have greater seniority in Japanese manufacturing firms.

Following upon Stinchcombe's analysis referenced above, those managers, now at the upper leadership levels, rose to their positions at a time when hardware capabilities were critical factors leading to their own personal and company's success. Thus, these top people tend to associate success with hardware innovation, unconsciously downplaying the growing contributions that can be made by software and service orientations. This lag in leadership recognition of current realities is hardly unique to Japan but may be more prevalent because of the strength of the age and length of service system, the company's extraordinary manufacturing successes in the past, and the association of hardware manufacturing (monozukuri) with national culture.

Of relevance here is the tendency of large old established firms to enter new fields by using spinoffs. Along with their established vertical keiretsu and tendency to staff their spinoffs with parent firm employees, these spinoffs tend to choke off opportunities for startups and limit the innovativeness of spinoffs.[95] Researchers have documented for select new technologies and industries the tendency of extant large Japanese hi tech firms to adapt and survive, while in the U.S., established hi tech firms more often disappear and are replaced by new firms. Such was the case in the shift from mainframes to pcs and in the hard disk industry.[96] How is this relevant to our preceding discussion? All things being equal, these large enduring Japanese IT firms and their spinoffs are more likely to carry over their founding organizational culture to the new field and be slower to adopt current practices and conventions. So we might expect them to be slower to adapt to the expanding importance of software. The new hi tech firms in the

U.S., however, are more likely, all things being equal, to adopt current practices and conventions which would accelerate the recognition of the importance of software.

In the case of the origins of Japanese IT, it is reasonable to assume that the most successful firms developed practices to recruit, train and promote high talent hardware engineers. Because they succeeded, they were more likely to be emulated by other high tech firms. It is only in recent decades that failure to recognize the importance of software and to recruit, train, and promote high talent software engineers has started to pose a challenge to organizational survival. The delayed recognition of software's importance can be attributed both to the success of these firms until relatively recently and to the institutionalized power of hardware engineers and their vested interest in protecting their status. This plays out in many different ways and is reflected in what we call organizational culture.

Digging deeper into the micro practices that sustain past practices, the action proposals that often look sensible to upper management are those which follow precedents, harmonize with current actions, resemble practices of other organizations with which they regularly interact, fit with top managers' values and reinforce existing power holders.[97] In short, they have strong vested interests and sunk costs growing out of their past experiences.

We have long standing research in the U.S. which shows management feels most comfortable promoting into their ranks people who are most like them. Why? It is because they see those similar to them as easier to communicate with, to understand, and to trust? Stinchcombe observes that the critical relationship of trust is more difficult to achieve when the parties do not have sufficient history together to be able to predict what their respective colleagues will do in a given situation. So if hardware engineers dominate the top management positions, they are most likely to feel most comfortable promoting hardware engineers. They know what hardware engineers do but software engineers produce an intangible product. All these factors slow the inevitable transition to the software era. Of course, there are, progressive Japanese firms that have moved ahead to successfully navigate the software transition but relative to the U.S. the overall movement appears more halting.

If a firm only brings in software personnel later in the product development process, this leads to the self-fulfilling prophecy confirming that software doesn't have much to contribute – that it is basically just a set of tools for implementing hardware centric product design. By the time software engineers are called upon to perform their tasks, the product development process, especially for complex products, is often behind schedule and the software engineers are under intense pressure to complete their tasks. In this environment, it is less likely that software engineers will be able to fully understand and take advantage of the hardware design strengths.[98] In summary, the cost of hardware coming first is that it sacrifices the optimization of software contributions.

The problems Stringer described are mirrored at many other high tech Japanese firms.[99] In the case of Sony, Stringer and his HR personnel responded to the problem by devising a new set of promotion practices designed to accelerate the rise of young software talent. In keeping with the constraints described above, however, the new system produced only modest results as middle management, often described as the true strength of Japanese corporations, blocked the full implementation of the new policies.

One reason for not enough collaboration between hardware and software engineers is that the hardware voice is often stronger than the software voice. In an organizational culture dominated by hardware people, there tends to be an unconscious assumption that hardware people should take the lead in product development. To be sure, the answer of who takes the lead should be determined by the product, whether it is integral or modular, by the dynamic nature of customer expectations, etc. That said, the growing potential contributions of software suggest the need for more collaboration among these engineering specialists and that it needs to be conducted on a balanced basis.

There is also an issue of how deployment of software personnel is related to positions of power and control. In my field research, I found much stronger recognition of the importance of prioritizing software among some of Japan's Korean competitors than among Japanese managers whom were interviewed. A Samsung manager recounted the aftermath of the firm's recent development of a hit product. A few years after the product was released, it was found that the hardware engineers, who worked on it, were being promoted faster than the software engineers who worked on it. A internal study of why this happened revealed that those higher level managers who were making promotion decisions were hardware personnel. The contributions of the software engineers weren't as visible to them as were the contributions of the hardware engineers and therefore the software engineers were promoted at a slower rate.[100] In short, organizational history, culture and power shapes which and how performance metrics get applied, often with participants being unaware of the biases they exhibit. In the case of Samsung's higher level management, this case showed their priorities. They demonstrated leadership by becoming aware of the situation, defining it as a problem and setting about trying to understand why this was happening and then developing countermeasures. Overall, I found Korean IT managers and government officials more aggressively talking about the importance of software than was the case with their Japanese counterparts.

Software Engineering as an Attractive Career Route

How are software/IT jobs seen in society at large? The results of a CNN/Money Magazine study of the best jobs in America were published in 2010. The survey methodology combined objective (e.g., growth rate of jobs, salaries, educational requirements) and subjective quality of life measures derived from a PayScale survey (e.g., amount of reported job stress, job flexibility) to arrive at their final list.[101]

These U.S. job rankings were included in a June 2012 IPA report aimed at Japanese firms with strong interests in software development.[102] The subject of the report was the comparative standing of the Japanese software industry, especially relative to the U.S. The CNN/Money Magazine survey results show that five of the top ranked twenty jobs in the U.S. are filled by software professionals. Kenji Hiranabe, the aforementioned leader in the Japanese agile software development movement, successfully pushed for the inclusion of the survey results in the IPA report. He explained to me why:

> To Japanese engineers, it is amazing that software developer and other related occupations are so highly recognized in the U.S. It seems that industry, investment, employment, innovation, education and others factors surrounding IT are mutually reinforcing one another in the U.S. to create positive business outcomes.[103]

Katsutoshi Shintani, Chief Advisor to the Software Engineering Center at the Information-Technology Promotion Agency (IPA), commenting on this same data, stated:

> Whenever I mention this data, I stress that Japanese software engineers need to be socially respected and rewarded with high pay as in the case of the U.S. I mean to wake up Japanese management to understand that if their companies are relying on software engineers for growth, then their software engineers need to be rewarded with high pay. Japanese firms also need to display publicly to outsiders the corporate contributions of their software engineers. In so doing, young and talented people will be motivated to work in software development.[104]

These comments suggest that software engineers are not treated all that well in Japanese corporations, nor seen as all that important in society. Consequently there are recruitment problems, especially with regard to getting high school students to consider majoring in IT fields. This would certainly be consistent with the structure of the industry as described above; many software professionals work in subsidiaries and tiered subcontractors with lower pay and more severe working conditions than would be the case than if they were employed by larger corporations. As already documented, they are less likely to be employed in software product oriented firms, whether start-ups or large corporations as is the case in the U.S. It is in the latter kinds of firms where higher status would more likely be accorded.

Japanese IT engineers hold what are often called the new 3k jobs. This labeling is widely used in the media. The term originated in manufacturing many years ago to describe low status manufacturing jobs characterized by dirty, dangerous and harsh conditions (kitanai, kiken, kitsui). In the world of software, this has been transformed to: lots of overtime work, low wages and stressful conditions (kaerenai, kyuuyo ga hikui, kitsui). This designation is more likely to apply to those working on customization, maintenance, and upgrading of enterprise software than it is to the higher status embedded software development jobs.

The characterization of IT engineers as low status is on display in a set of three featured articles published in Dec. 2006 in the popular business magazine, Nikkei Business.[105] Their analysis is supported by survey data from some 2,200 IT engineers.[106] They found that 90% of respondents reported stress in their work with over 1/3 of those reporting strong stress. The biggest source of stress was quotas and work content. Some 25% reported that they didn't think their work was worth doing. Fifty one percent report their salary was low (though no comparative data was reported). Respondents stated their overtime work averaged 12 hours a week (versus. 10.4 hours per month for all industry employees and 17.5 per month for manufacturing employees according to the government's Monthly Labour Survey. It may well be that the perception of low wages is influenced by the extraordinarily amount of overtime employees feel compelled to work relative to other jobs. As required by Japanese labor regulations, overtime work under 60 hours a week, is compensated only at a rate of 1.25 times the regular rate. A major problem reported in the survey was poor motivation of software engineers in their late 30s. Particular sources of concern, in these and other articles and surveys, are found in the firms' lack of clarity about employee career trajectories and prospects. Overall, these findings are consistent with IPA reported survey data. Nevertheless, the data are not drawn from a random sample so it is, at best suggestive.

Even allowing for exaggeration by these media treatments, they create an image that is hardly enticing to young people contemplating their future careers. This can only contribute to the problems universities and corporations have in recruiting software students and professionals.[107] Consistent with this interpretation, the survey found that some 73% of respondents believed that IT companies are unpopular among students and young engineers. The large amount of expected overtime work is believed to be a major source of these negative views. Clearly, for many of the best and the brightest, IT is not the career route of choice. In the end, this may be more of a constraint on Japanese software innovation than the education system which is far from best in class but often good enough for many purposes.

None of this is to suggest that the situation in the U.S. approaches nirvana. There is a longstanding and ongoing debate as to whether there are shortages or looming shortages of IT talent.[108] Those denying shortage argue that those claiming shortage misrepresent or misinterpret the data to justify importing low wage docile foreign workers through the H-1B visa program. Moreover, it is argued that the import of foreign IT labor diminishes the prospects for American citizens and reduces incentives for them to enter IT studies. This occurs at the same time that talented foreign students continue to flock to the highest ranked U.S.

universities to study CS and related fields. Many of these students have stayed to work for varying periods at U.S. corporations. Whatever the merits of the respective arguments, the U.S. IT industry has had available to them ovr the last 30 years a strong supply of talented IT workers and budding entrepreneurs. Indeed, the results of a 2012 study of engineering and technology companies founded in the United States between 2006 and 2012 is quite revealing. The study found that 24.3% of these companies had at least one key founder who was foreign born. This was a slight decline from previous decades. There is no counterpart to this phenomenon in Japan.[109]

There are relatively few examples of Japanese software firms which have achieved global success and few software heroes celebrated in the popular culture. By contrast, the U.S. job rankings shown earlier are consistent with the presence in the U.S. of a large number of software start-up successes and individual software heroes, entrepreneurs like Bill Gates, Steve Jobs and Mark Zuckerberg, Jeffrey Bezos, Larry Page and Sergey Brin. The U.S. also has highly successful global software companies like Google, IBM and Oracle. The considerable rewards for their founders' and employees success are well publicized. These cannot help but contribute to the positive view of software related jobs as seen in the broader society.

We can gather more insight into this matter by looking at the research findings of Universum, a firm engaged in employer branding.[110] They conduct large multi-national surveys of student's preferences for the most attractive potential employers. Business undergraduate students were asked to choose the five employers they would most like to work for. The 2012 U.S. findings show that among the top 20 most attractive employers, software firms accounted for four of the top sixteen employers. Google ranked first, Facebook 12th, Microsoft 14th and Amazon 16th.[111] Moreover, Apple, which differentiated itself from competitors to a large extent as a result of its software capabilities and strong integration between software and hardware, ranked second. Were Apple included, five of the top sixteen would be software firms. The comparable Japan rankings shows the first software firm on the top 20 list is Nintendo at number 10 and then Google at number 16, a total of two, with one being an American firm.[112] In short, we see further evidence here for relatively poor student perception of Japanese IT jobs offering promising career opportunities compared to the U.S.

In the business community, successful CIO's in the U.S. are held up as heroes in the business community. Recognized entities such as Silicon.com and CIO magazine produce annual list of the top 50 CIOs. The more popular Computerworld has its list of Premier 100 CIOs announcing that the winners drive strategy and innovation in top tier IT departments.[113] In short, software has a far more positive reputation in the U.S. business culture as well and it attracts among the best and the brightest in its universities. There are no well-publicized counterparts in Japan and as we have seen, many Japanese corporations either have no CIO or only a part time one.

With hardware increasingly becoming a commodity, it is software which increasingly adds value to the firm and enables competitive advantage. This being the case, one would expect that software personnel would be increasingly rewarded relative to hardware personnel. There is anecdotal evidence in the U.S. which shows heated competition for software personnel as reports grow that firms increasingly acquire other firms just to gain access to their software talent. Such extreme efforts are uncommon in Japan. What about wages and salaries? If Japanese firms are lagging in the recognition of the importance of software relative to the U.S., then we might expect a larger wage advantage for software personnel relative to hardware engineers in the U.S. compared to Japan.

What is the situation in Japan? First, there is a widespread view among software engineers, and in society at large, that their salaries are low compared to other professionals, such as hardware engineers. Software specialists, whom I interviewed, tend to discount such claims. The only evidence found in our literature search, provides weak support for a modest advantage for hardware engineers. Again, it may be that the long overtime worked by software engineers relative to other professionals fuels their sense of being disadvantaged overall.

We do, however, have data comparing salaries of engineering professions in Japan and the U.S. which provide indirect evidence of a gap. Nakata and Miyazaki report that Japanese engineering professions' overall salaries (whether using yearly or hourly salaries), are significantly lower than their U.S. counterparts, as measured using either exchange rate or purchasing power parity methods. They also found that salaries of Japanese male software engineers (referred to in Japanese statistical reports as system engineers), relative to their U.S. counterparts, ranged from 48.7 to 57.1 percent depending on whether they used exchange rates or purchasing power parity methods. By contrast, male Japanese general engineers fared notably better in their pay with their pay rising to 56.6 to 62.8 of their U.S. counterparts.[114] This does suggest that Japanese software engineers are not seen as valued a resource relative to their U.S counterparts as Japanese general engineers are compared to their U.S. counterparts.

Conclusion

This analysis began with a documentation of Japanese IT firm's relative weakness in patentable software innovation. One would do well to keep in mind, however, this does not mean that Japanese IT firms are outliers. It is rather more correct to say that U.S. IT firms, from a global perspective, display exceptional performance in software innovation. Were one to compare Japanese IT firms to those in other advanced economies, Japanese performance most likely would compare more favorably.

My analysis drew upon the findings of Arora and colleagues who analyzed USPTO time series data from 1983 to 2004. Specifically they examined the numbers of software citations in hardware patent applications for both Japanese and U.S. firm originated patents. They concluded that IT innovations are become more software intensive for firms in both countries, that Japanese firms rely less on software capabilities in their hardware inventions than U.S. IT firms and they produce notably fewer software inventions. They also concluded that Japanese IT firms are increasingly lagging behind especially in software intensive sectors.

The researches go on to analyze the cause of Japanese IT firms growing software deficits and conclude that they might well arise from a deficit in software human resources. They cite reports of software labor shortages in Japan, a 3 to 1 U.S advantage in annual software professional inflows to IT firms plus a much larger use of offshoring by U.S. IT firms. They also examine the share of software patents as a ratio of total patents invented by Japanese subsidiaries in the U.S. and find it occurs at a notably higher rate than the equivalent share produced by their parent companies in Japan. Based on this analysis, the authors conclude that a constrained supply of software knowledge and skills may explain the relatively weaker innovation performance of Japanese IT firms in the 1990s.

There are, however, problems with this analysis. First the authors dismiss the literature which suggests a “home country” bias that leads U.S. inventors to have a home court advantage in filing with the USPTO compared to foreign inventors such as the Japanese. It is also quite plausible that Japanese firms select the patents they choose to file in the U.S. from among those with the most promising export/sales potential to the U.S. and other global markets. This criterion could lead them to patent more software intensive solutions in the U.S. This scenario may have nothing to do with software human resource deficits in Japan being the source of their innovation shortfall.

The authors also documented that the U.S. advantage in the annual inflow of IT domestic software graduates and foreigners (H-1B visas in the case of the U.S.) into the domestic labor pool had grown to a 3 o 1 advantage by 2001. Added to this was a large U.S. advantage in the use of offshoring. Assuming these gaps led to human resource constraints in IT innovation and that this was recognized by Japanese executives, economic theory would predict that management and relevant government agencies would act to address these constraints. Yet, data, reporting the stock of IT professionals in 2010, show the same 3 to 1 gap advantage for the U.S. While Japanese offshoring appears to have doubled from its very low levels in the 2004-2005 period, it is not even clear that this gap was reduced as U.S. offshoring during this period certainly also grew. Given the failure to make major strides in diminishing the human resource capability gap, it may well be that Japanese management either did not recognize the existence or importance of a software innovation gap or it did not see human resources constraints as the critical factor in addressing the gap in software innovation.

To understand the situation more fully, we go beyond the period examined by Arora and colleagues. While one does hear talk of some specific shortage of IT talent in the past and nowadays, there is no widespread national conversation in Japan about large human resource constraints lying behind the recent past or current Japanese problems in IT. Nor has there been such a conversation for a very long time. In 1987, MITI predicted a shortage of software human resources totaling one million by the year 2000.[115] Today, METI officials acknowledge that this report was based on false assumptions and wildly off base.[116] Indeed, it has been argued that the only clear cut time of human resource constraints impacting software innovate occurred when all application and system engineering needed to be done in the Japanese language as thus required a vast army of software engineers. This changed when in 1991 IBM/Japan, together with many of NEC's competitors, introduced to Japan a bilingual operating system for personal computers .DOS/V. The significance of DOS/V was that it allowed Japanese purchasers of DSO)S/V to run English language MS-DOS compatible software at the same time it ran DOS/V Japanese software.[117] This opened up Japanese markets to foreign software and reduced the need for large numbers of software engineers.

The failure to significantly reduce the gap over this long period can also be interpreted to provide support for the view that Japanese IT executives have been slow to recognize human resource constraints as a big factor limiting IT innovation. It is also consistent, however, with the view that as top executives in IT firms came gradually to recognize the importance of software innovation for their competitive performance, they found themselves in a bind. They didn't know how to proceed without jeopardizing their legacy hardware businesses built on high quality precision manufacturing and requiring high quality and customization for their complementary software. Seen through their eyes, sustaining the legacy businesses does not lend itself to their prioritizing software innovation. Rather, it reinforces traditional monozukuri practices.

In this connection, I explored the evidence for the difficulty Japanese IT firms have had in making the transition from hardware centric to more software focused firms. Building on Stinchcombe, one can make a plausible case that organizations of a given type, tend to retain many of the basic characteristics they developed at the time of their founding. The nature of incentives used to drive performance and the characteristics of those initially recruited are often sustained because organizational and technological learning is highly path dependent. A firm's prior organizational routines, norms and corporate culture, often reproduced over long periods, constrains its future behavior. This is partly because opportunities and incentives for learning often will be close to previous capabilities. The extraordinary success of Japanese IT firms in the 1970s, 1980s and 1990s, reinforced many of their hardware centric practices. I elaborated various factors which enhanced the credibility of these behaviors in the view of hardware executives.

There is evidence that the quality of IT capabilities produced at Japanese universities has greatly lagged and continues to lag that produced at U.S universities. The slowness in upgrading IT curricula is notable and it is hard not to label Japan a slow follower in this regard. This can be interpreted to support Arora and colleagues' thesis of the continuing human resource constraints limiting Japanese software innovation. These same facts, however, are also quite compatible with the understanding that the Japanese educational establishment didn't grasp the importance of software to Japan's competitive strength and failed to act aggressively to address software weaknesses. Indeed, our analysis suggests powerful inertial forces that slowed innovation in this area.

Arora and colleague's analysis of the causes of the innovation shortfall ultimately rest on the assumption that more talented engineers yield more innovation. Yet, more talented engineers do not, in themselves, equal more productivity or innovation. Large aggregations of talented software engineers whether in a firm, region or nation cannot in itself be expected in itself to yield greater innovation. Software innovation benefits from powerful supporting ecosystems, from high quality engineers and especially from the embeddedness of these engineers in specific kinds of organizations and societal institutions whose objective is to enable firms to create and market these innovations to large numbers of users. Put differently, Japan's patentable innovation deficit vis a vis the U.S. may arise less from smaller numbers of talented engineers than from different organizational and managerial capabilities and especially from different strategic choices.

This suggests there are deeper structural causes of Japan's software innovation shortfall both in the period investigated by Arora and colleagues as well as later. The evidence suggests not enough Japanese products are focused on software and not enough software professionals are focused on product. Both of these lead to lack of demand for excellent software architects/designers. The net result is not enough software innovation. In this sense, the human resource constraints identified by Arora and colleagues at best are intervening variables reflecting a lack of focus on software products, especially ones intended for sales to large numbers of users. Relevant here is the thin staffing of IT professionals at IT firms. Also relevant is the huge amount of software capabilities directed toward customization for sales in vertical markets. The evidence for this can be seen most starkly in the foregoing account of the prototypical work and product flow of the "Big Five" Japanese enterprise software players based on our field observations. Opportunities for software innovation are modest in these activities.

The huge resources Japanese firms devote to customization are testimony to the importance attached to preserving monozukuri capabilities. In this environment, the primary role of software is to support the design, production and sale of specific monozukuri products.

As such, software is seen as an assistant to and controller of hardware products. Its potential as a driver of customer value tends to go unfulfilled and incentives for software innovation are modest.

Of particular concern for Japan's IT future, is the seeming low esteem in which IT work is held in Japan versus the U.S. This is reflected in IT not being seen as an attractive career route relative to the U.S. The reasons for this appear to lie in the limited opportunities for innovative activities and associated rewards for most Japanese software engineers. Instead, most Japanese software professionals are employed as cogs in IT service firms rather than in product producing firms or in startups.

Notes

[1] The following reference is to Japan's software imports and exports. The other weaknesses discussed here will be documented later in the paper. D. Lippoldt and P. Stryszowski, 2009. Innovation in the Software Sector, OECD, Paris: 94.

[2] T. Morgan, 2010. "IBM Reorganization Tucks Systems Under Software," (July 26), http://www.itjungle.com/tfh/tfh072610-story05.html. Accessed Dec.28, 2012.

[3] T. Fujimoto, 2004, *Nihon Monozukuri Tetsugaku* (Japanese Philosophy of Manufacturing). Tokyo, Nihon Keizai Shinbun.

[4] N. Fligstein, 1996. "A Political-Cultural Approach to Market Institutions," *American Sociological Review*, (August): 656-673.

[5] H. Kimura,2009. *Monozukuri Haisen* (Monozukuri Defeat), Tokyo, Nikkei Premium Book.

[6] OECD, 2002, Information Technology Outlook, Paris: Organization for Economic Cooperation and Development,: 105.

[7] M. Cusumano, 2008. "The Changing Software Business: Moving from Product to Services, *IEEE Computer Society,* (January):20-27.

[8] http://www.datacenterdynamics.com/focus/archive/2012/01/dell-shows-growing-importance-software-new-storage-releases. Accessed Oct. 23, 2012.

[9] R. Florida and M. Kenny, 1990 . *The Breakthrough Illusion: America's Failure To Move From Innovation To Mass Production, New York,* Basic Books; M. Cusumano, 1991*, Japan's Software Factories: A Challenge to U.S. Management*, New York: Oxford University Press.

[10] R. Katz, 2012a. "Plunge in Japanese Electronics, Part 1," *The Oriental Economist*, (March 1): 1-9.

[11] A. Nakatyama, 2012. "Electronics Firms Desperately Seek New Engines as TV Ops Sputter," Nikkei Weekly, (Feb.12):4, http://www.tse.or.jp/english/market/data/value/index.html. Accessed Jan.19, 2013.

[12] R. Katz, 2012b. "Plunge in Japanese Electronics, Part , Part 2," *The Oriental Economist*, (March, 5): 1-11

[13] Information Technology Promotion Agency (IPA), 2012. "Survey of Factors for Dissemination and Expansion of the Areas of Adoption of Non-Waterfall Model Development," (Hi Wotaforu Kata Kaihatsu no Fukyuu Yoin to Tekiyou Ryouiki no Kakudai ni Kansuru Chosa), IPA: 1-65; Japan Users Association of Information Systems(JUAS), 2008. "More than 50% of Big Corporations Install ERP Packages, Mainly for Personnel Management and

Accounting," Packages and Organization, http://www.juas.or.jp/en/research_results/index.html. Accessed Oct. 10, 2012.

[14] R. Cole 2006. "Software's Hidden Challenges," in D. Whittaker and R.Cole (eds.) *Recovering from Success*, Oxford: Oxford University Press: 115.

[15] K. Hiranabe, 2011."What Has Happened and is Happening in Japan's Agile Movement," *InfoQ*, www.inf.com/articles/japan-agile-whats-happening. Accessed 8/12/2012; D. West and T. Grant, 2010. "Agile Development has Changed Agility, Forrester Research, Inc., 5-21

[16] I. Von Morgenstern, P. Kenevan, and U. Naeher, 2011. "Rebooting Japan's High-Tech Sector," McKinsey and Company (ed*.) Reimagining Japan*, San Francisco: Viz Media: 297-308.

[17] Ibid., 298.

[18] A. Arora, L. Branstetter, and M. Drev, 2012. "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, Accepted for publication Manuscript 14150, http://www.mitpressjournals.org/toc/rest/0/ja. Accessed June 15, 2012.

[19] Ibid, 9-10.

[20] Ibid, 16.

[21] Ibid,30.

[22] Ibid, 8-9

[23] Sugimura, 2013. An Important Market: Software Patenting in Japan, Sugimura International Patent and Trademark Attorneys, http://sugi.pat.co.jp/portfolio/an-important-market-software-patenting-in-japan/. Accessed May 31, 2013.

[24] S. Nagaoka, K. Motohashi and A. Goto, 2010. "Patent Statistics as an Innovation Indicator," (Ed.) Bronwyn H. Hall, Nathan Rosenberg), *Handbook in Economics of Innovation*, Vol. 2, Oxford, Elsevier. B.V., 1100-1101.

[25] S. Kojima and M. Kojima, 2007." Making IT Offshoring Work for the (sic) Japanese Industries," in B. Meyer,nd J. Mathai (Eds*.), SEAFOOD (Software Engineering for Offshore and Outsourced Development)* Conference Proceedings, LNCS 4716 (2007), 67-82.

[26] JISA, 2012. IT Human Resources White Paper (IT Jinzai Hakushou 2012). *Information Technology Promotion Center*, 121.

[27] National Science Foundation, 2012. Unpublished tabulations of the 2003 National Survey of College Graduates provided to the Economic Policy Institute by Dr.Nirmala Kannankutty, senior adviser, National Center for Science and Engineering Statistics at the National Science Foundation. Data are tabulated by the presence or absence of various degrees for specific computer related occupations and appear in D. Costa, 2012. Stem Labor Shortages: Microsoft Report Distorts Reality About Computing Occupations, Economic Policy Institute. http://www.epi.org/publication/pm195-stem-labor-shortages-microsoft-report-distorts/ Accessed March 2, 2013; J. Mok, 2007. My Struggles on the Front Lines of Japanese IT," Japan Inc, Part II, (Nov. Dec.):1-4.

[28] JISA, 2011. Survey on Factors in the Diffusion of the Development of Non-Waterfall Models and the Expansion of their Fields of Application," (Hi Wotaforu Kata Kaihatsu no Fukyuu Yoin to Tekiyou Ryouiki no Kakudai ni Kansuru Chosa), *Software Engineering Center*, Information Technology Promotion Agency.

[29] Interview with Shinya Fushimi, Group Vice President, Mitsubishi Electric, March 23, 2013.

[30] The Japan Institute for Labor Policy and Planning, 2012. *Japanese Working Life Profile 2011/2012,* The Institute, 10, 14.

[31] F. Brooks, 1995. The Mythical Man-Month, Addison Wesley, Longman, Reading, MA.; K. Clark and T. Fujimoto, 1991. Product Development Performance, Harvard Business School, Boston: 71-76.

[32] D. Lippoldt and P. Stryszowski,op.cit., 23.

[33] W. Steinmueller, 1996. "The U.S. Software Industry: An Analysis and Interpretive History," in David Mowery (ed.), The International Computer Software Industry, New York: Oxford University Press: 42.
[34] Y. Nakata and S. Miyazaki, 2011. "The Labor Market for Japanese Scientists and Engineers: Is the Labor Market Externalized? What has Happened at their Workplace?" Japan Labor Review (Summer): 100
[35] J. Dedrick and K. Kraemer, 1998, Asia's Computer Challenge, Oxford University Press, 100.
[36] D. Costa, 2012. "STEM Labor Shortages," Economic Policy Institute, (November 19):4, http://www.epi.org/publication/pm195-stem-labor-shortages-microsoft-report-distorts/.Accessed April; 15, 2013. Original source is unpublished data from National Science Foundation's 2003 National Survey of College Graduates provided to the Economic Policy Institute.
[37] Ibid.
[38] In 2012, those receiving PhDs in CS taking jobs in industry (61%) or government (4%) together accounted for 65% of all CS PhDs taking jobs from North America in PhD granting departments. This compares to 29%, taking jobs in education from PhD. granting institutions, of which 14% were postdocs. Computing Research Association, 2013. 2012 Taulbee Survey http://www.cra.org/uploads/documents/resources/crndocs/2012_taulbee_survey.pdf
[39] E. Blum and Alfred Aho, 2011. CS: The Hardware, Software and Heart of IT," Springer, New York: 423.
[40] http://www.shanghairanking.com/SubjectCS2012.html.
[41] E-mail communication from Prof. Tetsuo Tamai, Hosei University, Nov. 8, 2012.
[42] Ibid.
[43] Email communication from Prof. Tetsuo Tamai, Nov. 20, 2012.
[44] N. Takahashi, 2001. "Information Processing Society of Japan Accreditation Committee Activities," (Part 1 of 2), (Jouhou shori gakkai ni okeru akuredeiteshiyon iinkai katsudou), Bit, (March):8.
[45] J. Dedrick and K. Kraemer, 1998, op.,cit., 48.
[46] Email communication with Michael Cusumano, April 17, 2013. Further confirmation comes from Masato Takeichi, e-mail communication April, 18, 2013. .
[47] Op.cit., N. Takahashi,1.
[48] Ibid., 5.
[49] M. Takleichi, 2003. Concerning Education of Personnel in Information Field in Japan's Universities (Waga kuni ni okeru jōhō bunya no jinzai ikusei ni tsuite), "2003 Information Studies Symposium" (2003 nen Jōhōgaku Shinpojiumu), Science Council of Japan (Nihon Gakujutsu Kaigi), Information Studies Research Contact Committee, Organizer: 14.
[50] Interview with Prof. Masato Takeichi, National Institution for Academic Degrees and University Evaluation, Tokyo, March 15, 2013 .
[51] F. Kimura, 2013. Symposium on IT and Monozukuri: Complexity of Artifacts and Control Structures Among Mechanical, Electrical and Software Sub-systems," Manufacturing Management Research Center, University of Tokyo, (March 18).
[52] P. Denning, 2003. "CS: The Discipline," in A. Ralston and. E. Reilly (Eds.), 2nd edition,Encyclopedia of CS and Engineering, New York: Van Nostrand and Reinhold.
[53] Op., cit. Interview with Masato Takeichi.
[54] http://www.t.u-tokyo.ac.jp/etpage/introduction/history.html.
[55] Op., cit. F. Kimura, 2013
[56] Op., cit. Interview with Masato Takeichi.
[57] Computing Research Association, 2013. 2012 Taulbee Survey, http://www.cra.org/uploads/documents/resources/crndocs/2012_taulbee_survey.pdf
[58] Email communication Abbas El Gamal, Chair EE, Stanford University, April 9, 2013
[59] C. Ness, 2013." From geeky to cool: Statistics is Berkeley's fastest-growing major" *Berkeleyan* (April 6). http://newscenter.berkeley.edu/2013/04/16/from-geeky-to-cool-statistics-is-berkeleys-fastest-growing-major/.

[60] Email communication, David Hodges, former Dean UC Berkeley, College of Engineering, April 13, 2013.
[61] Computing Research Association, 2013, op.cit..
[62] E-mail communication from David Hodges, 4/14/13.
[63] Op., cit. Interview with David Hodges.
[64] E-mail communication with Masato Takeichi, 4/13/13.
[65] Unfortunately the data do not break out the actual number of graduating majors in the two areas. http://www.t.u-tokyo.ac.jp/etpage/introduction/statistics02.html.
[66] Email communication Daichi Horie, Information Processing Advancement Section, METI, March 1, 2013.
[67] Y. Baba, S. Takai, and Y. Mizuta, 1997. "The User-Driven Evolution of the Japanese Software Industry: The Case of Customized Software for Mainframes," in David Mowery (ed.) *The International Computer Software Industry*, New York: Oxford University Press.
[68] JPSA,2007. *Package Software Market Expansion Research Forum Activity Fiscal Year 2004 Report,(Heisei 16 Nendo Pakkeeji Sufutowea Shijou Kakudai Kenkyukai Katsudou Houkokusho), Tokyo: Japan Personal Computer Software Association: 16.*
[69] Interview with Junichi Eguchi, Director, Information Services Industry Division., METI, March 28, 2013.
[70] Interview with Toshihisa Hirose, Senior IT Services Platform Specialist at NEC, March 22,2013
[71] J. Mok, 2008. "My Struggles on the Front Lines of Japanese IT," Japan Inc., Part III, (March 4): 1-2. http://www.japaninc.com/mgz_mar-apr_2008_japanese-enterprise-itAccessed Aug. 2, 2012.
[72] J. Dedrick and K. Kraemer. Op.cit., p.100.
[73] U.S. data based on the "2012 National Venture Capital Association Year Book, Thompson Reuters, New York. Data for Japan based on responses from survey questionnaire distributed by Japan Venture Capital Association members. Responses received from 32 companies. http://www.jvr.jp/sites.default/files/press_releses/2011FundReport.pdf. Accessed 10/14/11.
[74] Teikoku Databank, 2011. COSMOS2 database, 2011 edition.
[75] Murayama, Keiichi, 2011 "Entrepreneurs turning away from Japan," *Nikkei Weekly, (*Oct. 31): 43.
[76] S. Gopalakrishnan, 2011. "The T-Shirt and the Kimono," McKinsey and Co. (ed.) *Reimagining Japan*, Viz Media, San Francisco: 293.
[77] E-mail communication from Katsu Shintani 11/6/12.
[78] J. Mok, 2007. "My Struggles on the Front Lines of Japanese IT," *Japan Inc*, Part I, (March 25): 1-3. http://www.japaninc.com/mgz_spring_2007_frontline_japanese_it, Accessed Aug. 2, 2012.
[79] M. Cusumano, 2003. "Software Development Worldwide: The State of the Practice," *IEEE Software*, (November/December): 2-8.
[80] M Cusumano, "The Puzzle of Japanese Software," *Communications of the ACM*, (July, 2005):25-27.
[81] R.Cole and S. Fushimi, 2011. "The Japanese Enterprise Software Industry," in in H. Miyoshi and Y. Nakata (eds.), Have Japanese Firms Changed, Palgrave McMillan, Houndmills, England: 41-69.
[82] Ibid., 59.
[83] Information Technology Promotion Agency (IPA), 2012, op cit., 64.
[84] J. Peppart, 2011. "Unlocking the Potential of *the* Chief Information Officer (CIO)," *California Management Review*, Summer: 73-99.
[85] Ministry of Internal Affairs and Communications, Japan, *2008 Information and Communications Industries, 2007 White Paper.* http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2007/chapter1-2.pdf.
[86] This account relies heavily on the prior work of Cole and Fushimi. It particularly draws on the observations of Shinya Fushimi. Cole and Fushumi, op.cit. Hitachi, Toshiba and Mitsubishi Denki have announced that their primary business focus going forward will be large social infrastructure projects

(energy, transportation and utilities). These firms see software more a glue holding different project components together rather than as primary drivers.

[87] J. Kokuryo, 1997. "Information Technologies and the Transformation of Japanese Industry," *Paper presented at the Pacific-Asia Conference on Information Systems,* Brisbane, Australia: 1-15.

[88] O. Lorenzo, P. Kawalek, B. Ramdani, 2009. "The Long Conversation: Learning How to Master Enterprise Systems," *California Management Review*, (Fall): 166.

[89] J. Lincoln and M. Gerlach, 2004. *Japan's Network Economy*, Cambridge University Press.

[90] http://www.jetro.org/salesforce_how_japan_came_to_be_the_second_largest_market_for_this_billion_dollar_company Accessed May 7 ,2013.

[91] R. Nelson, 1995. "Recent Evolutionary Theorizing About Economic Change," *Journal of Economic Literature,* (March): 73-74.

[92] Lippoldt and Stryszowski, op.cit., 23.

[93] Y. Kane and P. Dvorak, 2007. "Caught Between Two Worlds, The Sony Chief Tightens His Grip. Will it Work?" *Wall St. Journal*, March 3. ttp://online.wsj.com/article/SB117285593145024927.html

[94] W. Mossberg, 2006. "Shaking Up Sony," *Wall St. Journal,* June 6http://online.wsj.com/article/SB114955974505772215-email.html, . Accessed April 5, 2012.

[95] M. Gerlach and J. Lincoln. 2000. "Economic Organization and Innovation in Japan: Networks, Spinoffs, and the Creation of Enterprise," in G. Von Krogh, I. Nonaka, and T. Nishiguchi (eds.): *Knowledge Creation: A New Source of Value*, London: Macmillan:151-198.

[96] H. Chesbrough, 1999. "Arrested Development: The Experience of European Hard Disk Drive Firms in Comparison to US and Japanese Firms", *Journal of Evolutionary Economics*, 287-330.

[97] W. Starbuck, 1988. "Executives' perceptual filters: what they notice and how they make sense," in D. C. Hambrick (ed.) *The Executive Effect: Concepts and Methods for Studying Top Managers*: Greenwich, CT: JAI Press: 35-65; Ibid, 1995. "How Organizations Channel Creativity," in C.M.Ford and D.A. Gioia (eds.), *Creative Action in Organizations,* Greenwich. CT: Sage: 106-114.

[98] T. Fujimoto and Y. Park, 2012. "Complexity and Control: Benchmarking of Automobiles," *Benchmarking: An International Journal*, 502-516.

[99] M. Fukuzawa, 2008. "A Generation-Selection Process of Product Architecture: A Case of Development of Firmware in Digital MFP," Annals of Business Administration Science, Tokyo, (December): 1-18; T. Fujimoto and Y. Park, 2012. Op.cit.

[100] Interview with Sungcheol Kim, Manager at the Chief Technology Office of Samsung Electronics' Advanced Institute of Technology), Seoul, Korea, April 2, 2012

[101] More specifically, the best jobs were picked according to the following methodology: Researchers created an industry screen using Bureau of Labor Statistics growth forecasts for 7,000 jobs, and identified industries with the biggest increases in jobs requiring bachelor's degrees. They then ranked them by 2008-18 expected growth and pay. They deleted jobs if at least 20% of workers were not in a high-growth industry. They kept fast-growing jobs not specific to any single industry. They then created a job screen and selected jobs that required at least a bachelor's degree, and eliminated those projected to grow less than 10% from 2008 to 2018. They cut jobs with median pay below $60,000 for experienced workers (except certain high growth job titles). They then eliminated jobs with fewer than 10,000 positions nationwide. The final ranking incorporated data from a PayScale Survey in which 40,000 workers rated their jobs on quality-of-life factors such as stress, job security, and flexibility and short- and long-term employment outlook. They ranked job titles using median pay for seasoned workers, industry and job growth, ease of entry, and estimated employment. http://money.cnn.com/magazines/moneymag/bestjobs/2010/index.html . Accessed May 5, 2012.

[102] Information Technology promotion Agency, op. cit., 22.

[103] Email communication July 23,2012.
[104] Email communication July 22,2012.
[105] Nikkei Computer Staff, 2006a. "Future Vision of Measures against 3k: Insecurity is the Problem for Motivation," (3k Taisasku yori Shourai Bijon o: Fuan ga IT Engjinia no Yarigai wo Sogu), *Nikkei Computer*, (December): 40-43;
2006b, "Stop Losing Engineers by Supporting their Childcare," (Shussan Ikuji Shien de Jinzai Ryushitsu wo Tomeru), *Nikkei Computer*, (December): 44-47; 2006c. "This is the Kind of Company I'd Like to Work at: Candid Answers from 2200 IT Engineers" (Konna Kaisha de Hatarakitai: IT Enjinia 2200 Nin no Honne,"), *Nikkei Computer*, (December): 48-53.
[106] The respondent and corporate makeup of this Internet survey are as follows: 65% work for IT vendors, 49% are employed in small and medium sized enterprises, 46% are middle managers, 31% are system engineers, 13% project managers, and 12% application and maintenance engineers.. Average age of sample is 39.3, 1,935 are male and 238 female and. 33% are full time employees.
[107] Nikkei Computer, 2006c, op.cit.,48-53
[108] D. Costa, op. cit.
[109] V. Wadhwa, A. Saxenian and E. Siciliano, 2012. *America's New Immigrant Entrepreneurs: Then and Now*, Kauffmann Foundation.
http://www.kauffman.org/uploadedFiles/Then_and_now_americas_new_immigrant_entrepreneurs.pdf Accessed March 2, 2013.
[110] http://www.universumglobal.com/ Accessed Nov 23, 2012.
[111] Unfortunately, there was no comparative data for engineering students.
http://www.universumglobal.com/IDEAL-Employer-Rankings/The-National-Editions/American-Student-Survey
[112] http://www.universumglobal.com/IDEAL-Employer-Rankings/The-National-Editions/Japanese-Student-Survey
[113] Peppard, op cit., 74.
[114] Y. Nakata and S. Miyazaki, op.cit., 103.
[115] MITI, 1987. Software Personnel in 2000 (Nisennen No Software Jinzai), Computer Age Ltd.,Tokyo, Japan, p.62.

[116]Interview with Junichi Eguchi, op.,cit.
[117] M. Fransman, 1995. Japan's Computer and Communications Industry, Oxford University Press, Oxford: 188.