The Globalization of Software R&D: The Search for Talent[1]

Avron Barr and Shirley Tessler
Stanford Computer Industry Project

While technological developments have enabled global distribution of software development teams, and wage differentials have stimulated offshore software services export industries such as India's, the overriding factor in the growth of globalization in this industry is the search for talented software people. The best software talent is in increasing demand as the software industry itself grows, and as software systems become a competitive tool in many other industries. Unfortunately, global supplies of the best talent are already tapped and there are no significant new sources readily available in the near future. The implications for US trade and public policy are far-reaching.

Software R&D. Software is defined as the instructions that a computer follows to perform a specified task. Software research in a corporate context is usually applied research, encompassing the design and prototyping of new products or systems, as well as the initial conception of them. Software development is the process of understanding and enumerating the requirements, translating those specifications into instructions for the computer, testing to make sure the specifications and their translation are correct, and documenting and maintaining this "program" as the people using it request modifications. The software R&D process differs from other technology R&D in that there is no tooling or manufacturing phase of product development; rather, when R&D is finished, the program is ready to copy, ship and use.

This paper focuses on corporate applied research and development. We have not addressed the "basic research" component, because it represents a truly small part of industrial software R&D activity. Software R&D spans a set of tasks including conception, design, specification, code development, testing, and documentation. In the past decade, most software outsourcing projects have

---

[1] A position paper for the December 12, 1996 meeting of the Council on Foreign Relations' Study Group on the Globalization of Industrial R&D. The authors are independent software industry analysts working with Prof. William F. Miller, Graduate School of Business, Stanford University, on a Sloan Foundation grant to study the global software industry. http://www-scip.stanford.edu/scip/

focused primarily on development and testing from clearly-defined and well-specified requirements provided by the outsourcing organization. In the more cutting-edge outsourcing endeavors, which have begun appearing more regularly in recent years, all parties to the project are involved with all stages, including the design, since it necessarily evolves iteratively with development, and is therefore much less amenable to formal specification.

Software R&D culminates in a finished program or system, not in an input that gets combined with other inputs in some proprietary way, and certainly not a discovery or invention whose commercial impact then depends on a secret process or methodology in the manufacture of the final product. Thus, when we talk about the foreign outsourcing of software R&D, we are really discussing the offshore production of a finished product in one of the most important industries for the US economy. Our task in this paper is to explore the unique characteristics of the software industry, especially the US software industry, that push firms to source R&D abroad, as well as the factors that constrain such an approach to software development. We will also examine some of the trade and public policy implications of the growth in offshore software R&D.

The importance of the software industry. The most visible segments of the software industry, the software products publishers and services firms, grossed worldwide in 1995 approximately $92B and $170B, respectively.[2] The US software products and services industries, at $71B and $92B in annual revenues, dominate the world picture.[3] With current growth rates of 12-15% expected to continue, some analysts predict that the industry will represent 10% of the GDP by early next century.

The true impact of software on the economy, however, goes well beyond product and services sales. In terms of dollar value of activity, more software is written by companies for their own operations (finance, administration, manufacturing automation, etc.) than the value of software products and

---

[2] **Software products:** Companies like Microsoft, Oracle, Novell and SAP. Sometimes sold shrink-wrapped, sometimes leased, sometimes bundled with hardware or with consulting services. Seventy-seven percent of the world's packaged software is sold by US firms.

Software services: Firms like IBM/ISSC, Anderson Consulting and EDS. Software-related services include consulting, systems design, systems integration, contract programming and maintenance.

[3] The reasons for the US dominance of the software industry, especially of packaged software products, are significant, and neither transient nor based on some "head start" advantage. In fact, US market share for software products increased between 1994 and 1995 from 74% to 77%. The US advantage stems from cultural attitudes toward software and programmers and the relative speed of change of industrial institutions in response to rapid technology change. See Edward Feigenbaum, *Where's the Walkman in Japan's Software Future,* in The Future of Software, edited by Derek Leebaert, MIT Press, 1995. Also, Barr & Tessler, *The US Domination Of Worldwide Software Products Sales Increased In 1995,* SCIP Research Note, October 1996.

services combined. Corporate operations software development is approximately a $500B expense worldwide. These "information systems" were originally deployed to gain efficiency in the financial and administrative data processing of large companies, in which role they have become essential. Today their impact is much deeper: home banking, self-service package tracking, frequent flyer programs and data mining for customer preferences are a few examples of the now-pervasive use of automation in the "front office" to attempt to achieve competitive advantage.

The biggest industrial impact of software mostly likely results from the use of embedded chips to achieve product functionality in everything from cellular telephones to automobiles. As the cost of computer chips drops, computers and the software that runs them have become as important a part of everyday products as they are of aerospace and defense systems. Companies in industries like telecommunications (cellular phones), aircraft manufacturing (on-board avionics), consumer electronics (programmable VCRs), and automobiles (air bags and anti-lock brakes) now have products whose competitive features depend on the software they've embedded in them. Some of these companies have indicated that 70% of their product development costs are now in software development. There are no estimates of the total amount of this "embedded" software in the products of so many industries, but we estimate that it is even larger than the $500B in-house software segment. Furthermore, software-based product functionality is a relatively new phenomenon and there is every indication that it will continue to grow rapidly, in the amount of software and in its complexity.[4]

[4] Reasons for the rapid trend toward more embedded software of increasing complexity: (1) Constantly cheaper chips mean more computers and more digital control systems in all kinds of devices, equipment and other products. (2) Putting increasingly complex functionality in software is cheaper and more flexible than mechanical or electrical solutions. (3) Once there is a computer in the product, there are all kinds of features that might have competitive advantage that can be "programmed in" in subsequent designs. (4) Once there are two or more computers in a product (there are over a dozen in many automobiles), the inter-processor communication requirements immediately makes the software much more complex. Added to feature expansion and maintenance "patches" over a multi-year, multi-developer lifetime, this embedded software will eventually become as complex as any.

Forces Supporting Globalization of Software R&D

The paths of evolution of off-shore sourcing of software R&D are as varied as the types of software people write. We outline here the major business influences that have moved us toward globalization.[5]

Downsizing and outsourcing of corporate IS activity. By far the largest consumers of offshore software services are the Information Systems (IS) departments of the major corporations. The reasons for the downsizing of IS departments during the 1985-1992 time period, conducted under the banner of "core competency," mostly reflected political battle lines that had been formed inside of these firms a decade earlier but came to a head with the advent of the PC over the issue of departmental authority to buy information technology. At the same time, there was a general misunderstanding, reflected in the business press, of the value of IT, focusing solely on productivity and ignoring strategic, competitive applications.[6,7]

The timing was right — offshore software services firms appeared during the period that IS departments had downsized, laying off the software people most experienced with their systems, and were beginning to outsource big contracts for operations and software development. At the time, these offshore service firms were offering low rates, based on wage differentials, as we will discuss below.

Technology change and shortages of specific skills. Another change in corporate information systems during this same period was the shift to client/server architectures, originally motivated by savings on hardware, and eventually necessitated by the movement to distributed computing. The conversion of existing systems and development of new systems that took advantage of the new technology required up-to-date skills that were not

---

[5] We will not discuss in depth, but mention here the familiar list of enablers that underlay the dramatic growth in globalization in the last 10 to 15 years: (1) growth of the packaged software industry, especially PC software; (2) global digital networks, which allow offshore workstations to appear as if they were on the office network; (3) e-mail, as the primary mechanism for interproject communication, which is blind to distance; (4) English as the international language of software; (5) common software tools and platforms accessible to development teams worldwide (such as C++, Visual Basic, etc.); (6) affordable computing equipment; (7) newly-enlightened government policies in countries around the world, in an attempt to nurture a domestic software capability.

[6] Erik Brynjolffson. *The Productivity Paradox of Information Technology*. Communications of the ACM. December, 1993.

[7] This period of downsizing caused painful and damaging dislocation to the many companies and industries. On the other hand, the resulting reorganization of the software industry around strategic partners and outsourcing, which never happened in Japan, for example, has been on the whole a competitive advantage for US firms, fostering the software publishing industry and the independent software services firms.

available to the shrinking IS organizations. (For example, making applications on desktop PCs look like other PC applications, with scrolling windows, graphics and multiple fonts, for example, instead of mainframe applications.)

Technology change is a constant in the world of computing. It is also the single most important factor in the US's dominance of this industry, since the relative flexibility of US companies and culture are an advantage in speedy response to changing conditions.[8] As a result of constant technology change, new skills are always required. The list of currently "hot" skills includes: Year 2000 programming, webmaster, Internet, Java, Visual Basic, PowerPoint, object-oriented, SAP, Oracle, and Windows NT. Finding programmers with these specialized skills has always forced firms to search broadly for talent.

Global markets for software publishers. In the software products segment, global markets reflect an opportunity to recoup increasingly large development costs with minimal additional development expenditures.[9] Software publishing is a numbers game, maximizing return on a fixed investment, and market share is the guiding principle.

The one development expense that publishers do incur is "localization," modifying the product to accommodate local language, practices, and hardware. Very often, this development work is done locally, and this activity might have been a starting point for some companies' global sourcing of R&D.

Furthermore, several publishers have found that they can produce products specifically for foreign markets, like Asian-language word processors or Intuit's product for a German's personal finances, which are very different from those typical for an American. Again, some or all of the R&D might well be sourced locally, as was indeed the case for Intuit.

Customer proximity. Software publishers are also required to support their products, and for some, like Oracle databases or Windows NT, this support requires local service technicians, not just a telephone hotline. Support of this type is usually part of their sales office, and often employs local people. Here the publishers share a motivation to develop offshore capabilities with the software services firms that custom-build, customize, install and maintain systems for local companies. These "local services" operations depend very much on frequent contact with customers to determine requirements, understand the systems environment, and keep up with changing business demands. They very often bring local talent into the company, and sometimes acquire local services firms to build this capability.

---

[8] See William F. Miller. *The Art of Rapid Change.* SCIP Research Note, 1995.

[9] See *The Importance of Being American*. The Economist. May 25, 1996.

Mergers, acquisitions and alliances. One of the most common reasons a software firm finds itself with an offshore development team is that it has acquired or partnered with a foreign software firm. Many foreign software firms use US firms as their primary distribution channel, and many US publishers partner with small foreign firms to obtain particular technologies or to establish a value-added channel abroad for their products. Acquisitions are a natural development from such relations, and acquisitions are rampant in the cash-rich, marketshare-conscious software industry.

Lower development costs by exploiting wage differentials. In many cases, reducing the cost of software projects was the driving factor in the movement of software R&D offshore five years ago, but we believe it is no longer a major factor. IS managers are so concerned about delays and quality problems in software development projects, both in-house and outsourced, that when they find a reliable, timely, high-quality source, they are willing to pay competitive rates. We discuss this subject later in connection with the Indian software industry.

Foreign government actions. Modeling the Indian success, governments of many countries are taking measures to encourage development of their local software industries, including incentives for foreign firms to open subsidiaries. The measures include educational initiatives (Singapore); changes in import restrictions and automation policy (Brazil, Russia); tax incentives (Ireland); and building the communications infrastructure (Malaysia).

Finding the talent. Demand for software is growing beyond the labor resources that we have locally to produce it, and the search for talent is driving the movement of software R&D offshore more than any other factor. Both for specific skills, as we discussed above, and for quality software people generally, software groups in all segments of the industry are expanding their search efforts.

The Increasing Demand for Software Talent

As a global company, IBM's "laboratories," located around the world, where both software products and embedded systems were developed, most likely led the way in the globalization of R&D. Global sourcing of R&D was not common, however, until the mid-1980's, when the independent software industry began to take off.

During the last 10 to 15 years, the transition from mainframe to desktop to network-focused computing brought about many changes in labor requirements in corporate IS departments. The proliferation of computing devices and their increasing importance to business and product strategy created constantly increasing demand for new software. Software got more complex every year, as a result of: the accumulated size and repeated

modifications to the installed base (to keep up with changing business requirements and to incorporate to hardware technologies), the requirements for systems to communicate with each other, and the constant invention of new ways to use all this technology.

At the same time, the tremendous growth of the packaged-product and services segments created additional jobs for programmers. Since the firms in these industries made a living on software, they knew from the beginning that they needed the best software talent. Furthermore, many of these firms were managed by software professionals who were better prepared to hire, nurture and reward the most talented analysts, systems architects, programmers, project managers, testing engineers, documentation writers and all of the other professionals that make up the software team. During this period, software people, who had once been no more than clerks-turned-coders in IS departments, became professionals, and eventually, folk heroes.

While we believe that the demand for software professionals has been increasing at an increasing rate during this entire period of 10 to 15 years, some major labor dislocations masked the increasing demand. These events were, of course, the massive downsizing of corporate IS departments in many industries and the post-cold-war defense spending cuts and subsequent layoffs in the aerospace segment. Both of these events peaked during the 1988-1993 period and resulted in the availability of several hundred thousand programmers. (At one time, there were 75,000 unemployed engineers in Orange and Los Angeles counties alone.) Many were re-absorbed immediately by software services firms who took outsourcing contracts from the same IS departments that had been downsized, but the trailing effects of these layoffs on the supply of programmers continued well into mid-decade.

In the last couple of years, a talent shortage has become apparent. We believe this shortage will soon become a serious issue in the software industry and in the many other industries that depend critically on software (computer hardware, telecommunications, banking, trading, etc.), and will eventually influence business and product strategy in many more industries. Competitive advantage, product differentiation, project delays, and systems errors and device failures are all at stake.

According to the Business Software Alliance,[10] US software employment has grown 9.6% annually from 1987-1994, and then surged to 11.5% in 1995. (This growth rate compares to a 1.6% rate for employment overall.) The demand for programmers is likely to continue to increase steadily for years to come because:

- the double-digit growth of the software products and services segments;

[10] In Kathy Rebello. *We humbly beg you to take this job, please.* Business Week. June 17, 1996.

- the retooling of corporate IS systems, and the expanded use of information technology for strategic, front-office automation; and
- the ever-increasing use of software to achieve product functionality in so many industries.

Furthermore, as we have noted, the programs that will need to be written in the future will continue to increase in complexity, requiring higher levels of software skills.

New tools or development methodologies will not reduce demand for software talent. Before we go on to look at the supply of programmers and their sourcing, we should ask whether this picture of constant growth in demand will continue unchanged. Perhaps, like telephone switchboard operation, future technologies will allow dramatic increases in programmer productivity and facilitate do-it-yourself programming by "end-users," forestalling the growth in demand.

Despite marketing campaigns over the years for a variety of technologies, like CASE (Computer-Assisted Software Engineering), knowledge-based programming, object-oriented programming, and software reuse, we are aware of no technological solution that will dramatically change the way most software is developed over the next ten years. Object technology, with its the notion of assembling software from off-the-shelf components, holds promise, but will require a generation to have widespread impact. For the foreseeable future, software development will remain a craft, dependent on the skill and experience of talented programmers.

Supply Has Not Kept Up With Demand

The US has approximately 2 million programmers, or twice the next largest country, Japan, with one million. The worldwide talent pool is currently estimated at about 6 million. However, not all programmers are created equal:

Good programmers are hard to find. Unlike most other engineering and "product development" disciplines, software has remained an art, the creative output of individuals with unique skills. Some studies show that the best programmers are 20 times better than average programmers, whether you measure speed, error rate, or overall output.[11] Thus, there is a "talent" element in the labor resource for this industry — not everyone can become a good programmer. The number of good programmers a nation produces is limited by raw talent, educational resources, and competition from other possible occupations.

---

[11] For example, see G. Edward Bryan, *Not All Programmers Are Created Equal,* IEEE, 1994. The worst programmers are two orders of magnitude worse than the best!

There are at present tens of thousands of unfilled positions for specific software skills ranging from SAP configuration specialist to Oracle consultant to webmaster. The US Department Labor does not collect detailed data on software professionals and software jobs, but we can present some anecdotal evidence based on interviews with software industry insiders here in Silicon Valley:

- Pro-sports-style, multi-million-dollar signing bonuses for key technical gurus are occurring more frequently.
- The increase in the number of programmers who go out on their own as contractors or form "boutique" software services firms also indicates a seller's market for software talent.
- Industry executives and technical managers express a growing concern about the availability of software professionals. Even well-funded start-ups and the major software publishers are beginning to see hiring delays, rising salaries, and increased mobility among software professionals.

We hypothesize that, in other important parts of the software development world, the "top half of the class" is no longer available at all, at any price. The best talent in each year's crop of newly graduating programmers are much less likely to head for a corporate IS department than they are to join a software publisher, even if they don't start their own company. Software publishers and software services firms are able to reward talented programmers more readily, both with salaries and with equity. Moreover, our interviews indicate that good programmers want to work where the best programmers are already working and in environments that cater to the needs (e.g., the best computing equipment, flexible hours, free midnight pizza). Furthermore, they often express a great deal of satisfaction in seeing the product of their labors on the store shelves.

Our educational system cannot produce enough of the right kind of people. Can we educate a new cadre of talented software people? Programs in software engineering and other efforts to train thousands of new programmers with the needed analysis, design, development, and project management skills are only beginning to be offered in the university and post-graduate education systems. For example, the University of California Extension Program for continuing education produced a curriculum of this nature just this year.

Regional software industry associations in Massachusetts, Minnesota, Silicon Valley and elsewhere have also begun to work with corporations and educational institutions to address local shortfalls in skilled manpower. Other organizations, like the newly-formed National Software Council, have called for certification programs to establish standard levels of professional competence, but these efforts are barely begun and not likely to be embraced quickly by the notoriously individualistic software community.

The software labor shortage is not a phenomenon particular to US. In fact, the shortage has been felt longer and more acutely in other parts of the world where software professionals were not as highly valued as in the US. Reports from the UK indicate that foreign outsourcing started about 15 years ago and today is the norm for large corporate IS organizations. Both the labor shortage and wage differentials are given as the main reasons. The UK and much of Western Europe look to India, Ireland, and increasingly, to Eastern Europe to fill their labor requirements. From the supplier side, outsourcing countries such as the Philippines also complain of labor shortages. There they cannot train enough new programmers or retain enough graduates (who migrate to higher paying countries) to accept all of the outsourcing work offered to them.[12]

The Indian Software Industry: A Misunderstood and Isolated Example

In the early 1990's, talented Indian programmers became available in the US in an organized fashion from intermediary software services firms.[13] Based on dramatic wage differentials, these firms were able to underbid domestic outsourcing firms for software development projects during the height of IS downsizing and the growth of the outsourcing industry in the US. Telecommunications technology, Indian government incentives, and clever business strategies helped make some of these early efforts successful. But the situation changed quite rapidly.

First, Indian programmers' wages increased from $1/15^{th}$ of US wages to $1/6^{th}$ of US wages over 5 years. At the same time, US consumers of these services found that getting good software delivered on time was "worth any amount of money," as one IS manager put it. It had become almost impossible in many firms to get any productivity out of an IS organization that was burdened by maintenance of existing systems and paralyzed by the continued threat of downsizing. Follow-on projects were bigger and less peripheral, and as the word spread, the cost differential decreased.

As a result, the Indian software export industry, which started out as a cost-savings alternative, has become a major source of talented programmers and software development services. In just 6 years, India has reached $700M in annual services exports by filling a growing skill shortage in IS departments in

---

[12] Antonio Lopez. *Goodby, Philippines: Can Manila Stem a Computer Brain Drain?* Asiaweek. October 25, 1996.

[13] Note that we are not talking about Indian nationals working as independents in the US labor market, but about an organized industry involving both client-site participation of (English-speaking) Indian programmers and outsourced development abroad. The familiarity phase of a project, in which Indian programmers might come to the client site to become familiar with the systems and the personnel for a year or more, is critical to the success of the offshore team.

the US, Japan and Europe. Indian businessmen have reduced the discount they need to offer to get repeat business from customers, and have grown their business by word-of-mouth based on successful projects. At the same time, US software and high-tech firms have opened subsidiaries in Bangalore to tap into this talent pool directly and attempt to realize increased cost savings.

Contrary to popular expectation, the Indian phenomenon will not be repeated in other countries in the near future. In India, the combination of the enormous population, the excellent engineering education system and the fact that there are few alternative careers for engineers created a giant pool of software talent (approximately 200,000 in 1995). Clever entrepreneurs and effective government policies turned that resource into an important export industry for India. However, although there are talented programmers all over the world, there are no other countries in which we find an enormous and underutilized engineering talent pool combined with English competency, systems and communications infrastructure, and the entrepreneurial energy that made India's software industry such a success. We believe that there will be no more Indias for at least the next 10 years — not China, not Russia, not Eastern Europe.[14]

Hurdles, Constraints, and Limiting Factors

To balance the picture, the globalization of software R&D is not altogether straightforward or natural. Some factors to consider:

1. The sheer complexity of the software requires extensive periods of familiarization, which in turn requires communication with experts. Communication and training is a problem in any event when the experts are so busy. Distance, cultural differences, and language all contribute to communications problems.[15]

2. Rapid technology changes, market movements, and business requirements require a high degree of coordination and common understanding to achieve product cycles which are very short.

3. For software publishers, piracy is a major, bottom-line issue. As a result, off-shore development on new products is less of a trend in this segment. In fact, only in recent years has there been any movement software publishing firms to source development offshore, and here too, the motivations are

---

[14] The subject of the Indian software industry in particular will be revisited in Tessler & Barr's paper in CFR sessions 4-5, "Comparing Budapest, Bangalore, Santiago and Singapore."

[15] See also Erran Carmel. *Thirteen Assertions for Globally Dispersed Software Development Research.* To appear in Proceedings of the 30th Hawaii International Conference on Systems Science. January, 1997.

complex. A Stanford Computer Industry Project survey of software development practices among software publishers showed only one firm going outside of the country for any type of software development and, in that instance, they engaged an Indian firm only for Web page development related to their new product.[16]

4. Any country besides India which is developing fast enough to produce lots of good programmers is going to have a domestic demand for those programmers in their own banks, phone companies and manufacturing firms. In most cases this will preclude the development of a large software services export business in the short run.

5. Many of the isolated examples of offshore sourcing of R&D are instances when a foreign group has developed one particular technology which is being exploited by a US firm. It is not easy for these foreign groups to expand their capabilities or markets.

6. Restrictions on technology exports, like encryption, also creates obstacles to offshore sourcing of R&D.

Public Policy Considerations

There is a growing movement in the US to cut down on immigration, both legal and illegal. Several engineering professional organizations, including the American Electronics Association, have pointed to immigrant engineers with lower wages as a threat to American jobs. We are concerned that proposed immigration policy changes, if applied without exceptions for software professionals, would cut the flow of the world's best programmers to the States. These measures are unnecessary, since there is a shortage of skilled labor in software, and could be damaging to the software industry and to the economy generally. The US is still the best place for good programmers to work, and we should encourage them to come here while they still have fewer domestic alternatives. Moreover, as suggested by the National Academy of Engineering,[17] "...the economic benefits of R&D activity tend to be much more localized than is commonly assumed." Clearly the US should continue to import scientists and foreign graduate students rather than impose restrictions which would force US companies to search abroad for even more of its labor requirements.

---

[16] See Shirley Tessler and Avron Barr. *A Pilot Survey of Software Product Management.* To appear in the proceedings of the Software Engineering Process Group Conference. San Jose, 1997.

[17] National Academy of Engineering, *Foreign Participation in US Research and Development,* 1996.

The issue of software quality will also reach the attention of policy planners in the next few years, perhaps rather dramatically on January 1, 2000.[18] Business and government organizations have no choice but to continue to build software that is critical to their operations and to our lives. If they cannot find enough good people, somewhere on this planet, to build these systems right, the result will be systems delays, failures, buggy software products and devices that don't work. Inevitably, there will be loss of life.[19, 20]

If outsourcing of R&D effectively means outsourcing the creation of a finished product, then foreign contractors have both possession and control of valuable intellectual property. The software industry's low barrier to entry, combined with the ability to duplicate and distribute identical copies of programs at no cost, make offshore sourcing of software R&D riskier than the foreign outsourcing of other types of industrial R&D. This risk has confined most outsourcers to contracting out the more mundane R&D tasks that carry fewer consequences for the firm than outsourcing key pieces of their core business products or services. However, as the labor shortage becomes more severe, the industry may have to lower its guard and outsource far more proprietary components of its technology. The government's continued commitment to negotiating and enforcing *reasonable* intellectual property controls worldwide will become even more crucial.

Another area of public policy concern is the anti-competitive implications of concentration of software talent. As we mentioned, some software firms are already quite predatory in identifying and hiring the best talent. Since software is a critical resource for firms in other industries, we expect that some banks, airlines, insurance companies, and others will build technical teams as strategic business weapons, which in turn might dramatically change the competitive balance of those industries. We have also begun to see serious efforts appearing at the national policy level in competing nations to ensure an adequate supply of software talent.[21]

---

[18] See, for example, Peter De Jager's Millennium predictions, including the global stand-down of all air traffic on New Years Eve, 1999. Computer Weekly, August 15, 1996.

[19] The Standish Group International estimates that in 1995 US companies and government agencies will have spent $81 billion for canceled software projects and an additional $59 billion for software projects that exceed their original time estimates.

[20] Note also that movement of programmers between companies, which is symptomatic of shortages and salary ramping, also has a devastating effect on project deadlines and potentially on software quality. See Shirley Tessler and Avron Barr. *A Pilot Survey of Software Product Management.* To appear in the proceedings of the Software Engineering Process Group Conference. San Jose, 1997.

[21] Nagy Hanna describes efforts by the Japanese government, including a 3-year training program in Japanese language and software engineering, to enlarge the supply of overseas software talent. *Exploiting Information Technology for Development: A Case Study of India.* World Bank Discussion Papers. 1994.

The training of new programmers, and the re-training of existing programmers, are more complicated issues. Efforts to develop a curriculum to produce more qualified software professionals should be accelerated. While the best software engineers seem to keep themselves abreast of new technologies, efforts should be made to encourage firms to help their software professionals keep current with technology, to avoid finding themselves obsolete as technology advances. The biggest market for Indian programmers is in staffing rapidly-growing specialties like Visual Basic programming.

We also note that in software R&D, foreign outsourcing does not engage just a few scientists and other very high-level professionals, but provides employment to a larger number and wide variety of skilled professionals who derive valuable skills and experience that might be useful in the development of a domestic software industry. Clearly, we are training our competition. We feel, however, that this is not an issue of concern for the foreseeable future. The labor requirements in those countries may soon become as problematic as they are in the US.

At the present time, government data about the software industry and the software labor pool are inadequate.[22] The increasing importance of software in so many industries, in addition to the growing importance of software products and services themselves, speaks to better surveillance on this industry. Specifically, more and better categories of software and software professionals should be tracked in both the business and labor censuses.[23]

One final concern about the impact of a software labor shortage: the world of software and IT generally will slow down if projects are delayed or not even attempted because of lack of adequate development resources. Business software publishers depend on their customers' programmers and systems integrators to adopt their latest offerings. They will delay releases if corporate customers are unable to integrate new products into their operations. Innovation will be hampered for the first time in the history of computing.

---

[22] Avron Barr, Shirley Tessler & Brian Lent. *Revising the Classification of the Software Industry*. Memo to the US Office of Management and Budget, Economic Classification Policy Committee, March, 1995.

[23] In fact, most companies we talk to do not know how much they spend on software (purchase/lease, upgrades, installation, training, outsourced development, in-house development, maintenance, and embedded). They do not know how many of their employees are involved in developing and maintaining software. And they cannot account for their software assets with any accuracy. The few firms that have investigated this issue have been appalled at the size of the expense and the criticality of their software effort. Policy changes and a new respect for the importance of software to their business are positive outcomes.

Before we express relief over the thought of a pause in the rapid and relentless rate of change of the last 40 years, we should consider this: the advantage of the US in software, and thus in information technology generally, is based on a differential capability to accommodate rapid change. When the pace of change slows down, everyone else will catch up. True, this technology revolution will slow down eventually, when inventions reach the physical limits or some utility plateau. But a premature braking of the rate of innovation because of a severe labor shortage will have serious negative consequences

Concluding Remarks

Driven by the increasing demand for software of all types, US firms both in the software industry and in other industries dependent on software, will continue to seek software R&D resources offshore. By and large, the US software industry recognizes that it is exporting projects and jobs overseas at an increasing rate. While most of the work is currently in the less creative areas of software development, such as Year-2000 retrofitting and the rewriting of legacy systems, more important and cutting-edge software R&D will increasingly be outsourced overseas to take advantage of scarce talent.

The industry may be inevitably leveling the worldwide playing field for software development in its relentless drive to find new software talent and open new markets. Yet, two factors mitigate US software industry concern. First, as a result of long-term domestic problems such as poor software training, high rates of piracy, or counter-productive government policies, most of these countries are not likely to become either competitive software publishing centers or software services exporters, at least not in the next decade. In fact, in most cases, they are not likely to catch up to their own increasing domestic demand for software. Secondly, until the rate of technology change slows significantly, current US policy and cultural attitudes will continue to give this country an advantage in the software industry as it continues it rapid evolution. There are policy changes that might improve the situation, including increased basic research funding, targeted training programs to produce good software professionals and enhance the skills of existing programmers, and especially, an open immigration policy for skilled programmers. The software industry will thrive, however, under current policy as long as computing technology continues to advance at breathtaking speeds.

Sources

Asia's Top Software Firms Count on the Net to Become World Players. Asiaweek. October 25, 1996.

Barr, A., Tessler, S. and Lent, B. *Revising the Classification of the Software Industry*. Memo to the US Office of Management and Budget, Economic Classification Policy Committee, March, 1995.

Barr, Avron and Tessler, Shirley. *Good Programmers are Hard to Find: An Alternative Perspective on the Immigration of Engineers.* Stanford Computer Industry Project, Research Note. October, 1996.

Barr, Avron and Tessler, Shirley. *The US Domination of Worldwide Product Sales Increased in 1995.* Stanford Computer Industry Project, Research Note. October, 1996.

Barr, Avron. IS 2000: A New Role for Corporate IS Departments and People. An address to the Software Development Conference. San Francisco, 1995.

Bellinger, Robert. *A Vintage Year.* EETimes. September 30, 1996.

Brynjolffson, Erik. *The Productivity Paradox of Information Technology*. Communications of the ACM. December, 1993.

Carmel, Erran. *Thirteen Assertions for Globally Dispersed Software Development Research.* To appear in Proceedings of the 30th Hawaii International Conference on Systems Science. January, 1997.

Computer Software. US Global Trade Outlook, 1995-2000. US Department of Commerce. 1995.

Dave, Rishi. *Patterns of Success in the Indian Software Industry*. Stanford University. May, 1996.

Global Competitiveness of the US Computer Software and Service Industries. Office of Industries, US International Trade Commission. June, 1995.

Goff, Leslie. *Yeehah!!* Computerworld. September 23, 1996.

Hanna, Nagy. *Exploiting Information Technology for Development: A Case Study of India.* World Bank Discussion Papers. 1994.

The Hitchhiker's Guide to Cybernomics. The Economist. September 28, 1996.

The Importance of Being American. The Economist. May 25, 1996.

Lopez, Antonio. *Goodby, Philippines: Can Manila Stem a Computer Brain Drain?* Asiaweek. October 25, 1996.

O'Riain, Sean. *An Offshore Silicon Valley? The Emerging Irish Software Industry.* Unpublished report. 1996.

Offshore Assets. Computer Weekly. July 8, 1996.

Rebello, Kathy. *We Humbly Beg You to Take This Job Please.* Business Week. June 17, 1996.

Reid, Proctor and Schriesheim, Alan, ed. *Foreign Participation in US Research and Development: Asset or Liability?* Washington, DC: National Academic Press, 1996.

Schmid, Gregory. *The Future of America's Research-Intensive Industries.* Institute for the Future. August, 1995.

The Standish Group International, Inc. *The Chaos Report.* Dennis, Massachusetts, 1995.

Tessler, Shirley and Barr, Avron. A Pilot Survey of Software Product Management. To appear in the proceedings of the Software Engineering Process Group Conference. San Jose, 1997.

Zachary, Paul. *US Software: Now It May Be Made in Bulgaria.* Wall Street Journal. Feb. 21, 1995.