Software Engineering: Globalization and Its Implications

Rafiq Dossani
Stanford University

&

Martin Kenney
University of California, Davis, and
Berkeley Roundtable on the International Economy

This paper has been prepared for the National Academy of Engineering 'Workshop on the Offshoring of Engineering: Facts, Myths, Unknowns and Implications', October 24-25, 2006, Washington DC.

Contents

1. Introduction, p.1
2. The Historical Development of the Software Industry, p.3
2.1 The Offshoring of Software Production, p. 7
2.2 Case Studies of Offshoring Product Development by U.S. Firms to India, p.12
3. A Theoretical Framework of the Drivers of Software Offshoring, p.17
4. Concluding Thoughts on the Impact of Software Offshoring on Employment, Innovation and Policy, p.19.

Abstract

The offshoring of software development is over three decades old and has been at the leading edge of the offshoring of information technology services. Over the past decade, the pace of offshoring has dramatically increased due in large part to new communication technologies and the opening up of India. This report discusses the evolution of software's globalizing supply-chain. It argues that high value-added work will be an increasingly large component of offshored software development in the future. We discuss the implications of this for employment and innovation in developed countries.

Section 1. Introduction.

Software exporting firms located in India employed 513,000 persons as of 2005; in the US, 2.6 m persons were employed in software production and related occupations. In 1995, the comparable numbers were 27,500 and 1.5 m.[1] Even as the centrality of information technology in the operation of the US economy is evident from the rise in US software employment, these statistics are telling of the substantial globalization of the software industry that is under way. Further, while its end state is unknown, it is clear that developing countries are integral to the globalization process.

A lively discussion about globalization's impact on the software industry's employment and productivity is under way. A report published in 2006 by the Association for Computing Machinery (ACM, 2006) notes that "attracted by available talent, work quality and most of all low cost, companies in high-wage countries, such as the United States and United Kingdom, are increasingly offshoring software and software service work to these low-wage countries." (p.19) The report concuded that "the globalization of, and offshoring within, the software industry will continue and, in fact, increase." (p.13).

These findings suggest dramatic impacts for software-related employment in developed countries. If software development overseas increases in quantity and, especially, in scope to cover the most highly-skilled work, unemployment of skilled software engineers in developed countries is a likely outcome (Hira and Hira 2005). The ACM report notes that higher-value work is already shifting in some fields of software, such as computing research (p.155, also Dossani (2006)). The report's primary recommendation is that developed nations urgently develop policies to foster innovation through continuing investment in education and research and development (p.12).

For theorists such as Bhagwati and Mankiw (eg, Bhagwati et. al. (2004), Mankiw and Swagel (2006)), the globalization of the software industry is little other than evidence that many services are now tradeable, whereas earlier they were not. Like all trade, they argue that this has positive implications for the US economy. They argue that workers in the services sector of developed nations will shift to work in which they have a comparative advantage, so that full employment in the long-term is assured. As Mankiw and Swagel (2006, p.4) note, "Economists see outsourcing (actually, offshoring: authors' note) as simply a new form of international trade, which as usual creates winners and losers, but involves gains to overall productivity and incomes."

In contrast, as Samuelson (2004) has noted, these gains can be captured by developing countries; and Baumol and Gomory (2000) have argued that nationally located high-growth industries are important for national growth because of their spillover effects on overall productivity.

There is no comprehensive empirical evidence on software offshoring, in part because, as is widely conceded, the quality of data is poor (See Figure 1). From what we know of macroeconomic data, it appears that there is no systematic evidence yet of significant high-value job losses due to services offshoring. As noted in NAPA (2005), "The number of jobs impacted (by services offshoring) appears relatively small, when compared to total annual job losses in the United States." (p.50).

Other empirical work offers indirect evidence in support. Mann (2006) shows that elasticity of demand for US exports of services is lower than for US imports of services (p.115). If applicable to software, this would imply that the industry's globalization could have positive implications for the US balance of payments. Landefeld and Mataloni (2004) show that the share of imports from subsidiaries of US-based multinationals to the parent (as a percentage of sales) did not rise between 1997 and 2001. They also find that job creation by multinationals expanding overseas is no different from overall job creation.

According to Hanson et. al. (2001), the evidence from manufacturing's offshoring shows a positive, complementary effect on American jobs from high-value offshoring and a negative, substitution effect from low-value offshoring. If applicable to software, the export of low-end work, such as programming, could reduce industry jobs because the largest number of man hours in any software project is in programming. Mann (2006), citing BLS data, notes that 6 of the 20 most rapidly growing jobs in the decade 2004-2014 are in software (including network systems and data communications analysts and administrators, and software applications and systems engineers). For computer programmers, however, the rate of growth is projected to be below that of the economy as a whole. This may not be all bad news for employment in the software industry. If programmers can shift, with some training, to computer engineering, then they can leverage the programming work being done overseas. This appears to be happening (Dossani, 2006).

These sectoral arguments provide little insight into the developments within software and provide no evidence that the software industry will continue at full employment despite offshoring. Nor does it tell us which types of jobs in software will remain onshore and their impact on productivity. Meanwhile, what is clear at the level of the firm is that the necessity of undertaking software development at a particular location is, in many cases, being eroded. This could mean that, over time, the software industry will shift to lower-cost countries like India or China, which will capture the economic gains of software offshoring.

Nevertheless, what is true for existing firms is not necessarily true for the evolution of an industry. The history of technological changes suggests that new opportunities will emerge and the software industry in developed nations could reinvent itself to a higher value-added path, even as existing firms move increasing portions of current operations offshore. These new opportunities depend on the pace and location of innovation. This could be affected by the development of clusters of technical excellence offshore, such as may be happening in Bangalore, Beijing of Shanghai. Or, it may be, as many scholars have argued (eg., Apte and Mason (1995)) that the need for proximity to understand consumer needs will be the determining factor in locating innovation. Or if, because of their more open economies and better educational system, US firms innovate at a pace that keeps them ahead of China and India, then the American software industry can thrive by keeping the highest value-added work onshore and offshoring the rest.

We, therefore, argue that to predict the ultimate outcome requires an understanding of the software industry and its evolving supply-chain. This is the task of the present chapter, which proceeds as follows:

Section 2 provides a historical overview of the developments that led to the offshoring of the software industry, with a focus on the developments in India. Section 3 provides a theoretical framework with which to analyze how higher value-added work may be offshored. Section 4 discusses the impact of software offshoring on employment and innovation in the US and other developed countries and the policy implications.

Section 2 The historical development of the software industry

Software may be classified by type of use and by customization.

Types of software by usage:

1. System-level software: programs that manage the internal operations of the computer, such as operating system software, driver software, virus scan software and utilities.

2. Tools software: programs that help applications to work better, such as database management software.

3. Applications: programs that deliver solutions to the end-user, such as word-processing software and financial accounting software.

Types of software by customization:

Software is either (1) written for general use and replicated in its original form across many users, or, (2) written for a specific user. The former is termed a software product or package. It may be shrink-wrapped and transported physically or over the Internet. The latter is termed custom software. Being made-to-order, custom software is more geographically constrained that products, i.e., proximity to the user is more important. Because of this, software products are more readily exportable than custom software.

System-level software is the most complex as it manages the interfaces with both hardware and higher level software; applications software is the least complex. Nowadays, all system level software are products. The more varied an end-user's needs from another end-user, the more likely is the software to be customized. Since variations in needs appear most at the stage of applications, most customized software is applications software. These attributes, the size of the market and the market shares of two key players other than the US, India and Israel, are summarized in Tables 1-3 below.

The independent software vendor (ISV) industry was created by two events, both related to market leader IBM: (1) In 1956, in order to settle a long-standing antitrust suit by the federal government, IBM agreed to cease offering computer consulting advice, as part of a consent decree (McKenna, 2006, p.20).[2] Leading accounting firms, such as Arthur Andersen, then began computer consulting services. (2) In 1969, IBM decided to unbundle its mainframe operating system, applications software and hardware by creating open standards. Subsequently, some end-user firms set up inhouse software development and maintenance operations (column B below) while others outsourced the work (columns C – E). The resulting ISV businesses are shown in Table 4 below.

Columns A to E above are not intended to describe mutually exclusive choices. For example, a firm might purchase system level software products while developing its own applications.

The columns are arranged by sequentially dominant work-types over the decade, starting with the shift from external data processing and managed services (Column A) to inhouse hardware at the start of the decade. Initially, firms developed their own software (B). As the 1970s progressed, hardware and software became more complex making inhouse software development and management more difficult. This led to the outsourcing of system integration (C) and then to the sourcing of system level and applications products (D). The move to outsourcing customized applications (E) was due to the failure of industry specific products to meet the needs of the more sophisticated users, particularly the large banks (Steinmuller, 1996, p. 30).

In the 1980s, the PC was invented. The Wintel standard became established by the mid-1980s, leading to a decline in hardware prices and rising demand for applications. Unlike mainframes, the PC was for retail users (both individuals and small firms) who were reliant on product software. The PC of the 1980s lacked both the programming capacity and performance needed by mid-size and large enterprises. Hence, it had no impact on the custom software business. However, it spawned the creation of a mass market for retail product software.

The workstation, introduced in the early 1980s, had many end-uses, but could also be used for stand-alone programming for mainframes. The widespread adoption of Unix as the operating system for all computers, jointly with the workstation (in short, the U-W standard), revolutionized the ISV industry. An ISV could now own a workstation made by any manufacturer, yet write programs for a client whose installed hardware might be of a different brand (even a mainframe). In other words, software creation became platform independent or modularized[3] from the hardware component. With the simultaneous widespread adoption of C as the programming language, the other functions of software creation, such as system architecture, design and integration could be done separately from the programming, thus modularizing the programming component and enabling its remote production. Programming could now be done anywhere in the world by programmers whose only raw material, apart from a workstation, was a specified software system, i.e., the programmers did not need to know which firm's hardware the program would work on and even the type of application the program was intended to support. In the 1990s, the success of database software packages further simplified the creation of applications software. The platform independence that arose from the U-W standard, combined with the rise in demand for custom software by small firms, resulted in a large custom software industry.

The workstation also had sophisticated graphics and the computational capacity needed by small enterprises. Such firms shifted from outsourcing data processing services to running their own workstations. By the end-80s, the first workstation-based local area networks were established and increased the demand for more sophisticated software for running such networks and for applications compatible with networked users.

In the 1990s, the rising computing power and declining cost of the PC improved its capacity to program in Unix/C. The PC, therefore, replaced the workstation as the hardware platform for programming. Later in the decade, the success of PC-based networks increased the accessibility of applications to many more users within the enterprise.

The spread of the Internet from the late 1990s was driven by declining bandwidth and storage costs. The Internet provided a platform for networked development of software, and software installation, hosting and maintenance. This influenced the location of data, which moved away from servers located at the premises of the enterprise to remote data centers. The Internet also significantly reduced the cost of collaboration among remote teams. These factors further reduced the need for proximity.

Since the Internet has become established, several new models of preparing and delivering software have appeared. These include service-oriented architecture that provides a standards-based environment for sharing services, independent of development technologies and platforms; network-based access to and maintenance of software (software-as-a-service); and open source software, i.e., software based on non-proprietary code, whose development relies on the voluntary contributions of networked developers. With the exception of the Linux open-source operating system software, whose market share of webserver operating systems is believed to be about a third, although it is less than 2% of all operating systems, these have not impacted the existing models of software development.[4]

The first four columns in the table below summarize the main changes in the software services industry in the U.S. and the driving forces reviewed above. The latter two columns display the parallel developments in the Indian and Israeli software industries.

Section 2.1 The offshoring of software production.

American IT firms began to offshore software development to India, Ireland and Israel (the 3 I's) in the 1970s, about a decade after offshoring IT hardware manufacturing. Siwek and Furchgott-Roth (1993, p. 93-4) suggest that the delay was because, unlike manufacturing, software development was more closely linked to customer requirements and required close coordination within the firm.

The widespread knowledge of English and relatively low labor costs were common attractions of the '3 Is' although small domestic markets and the lack of domain knowledge (less so for Israel, see below) were common disadvantages. Subsequently, starting in the 1990s, many other countries exported software to developed countries, notably China, several countries in Eastern Europe, Brazil, Mexico, Russia, the Philippines and Vietnam (see Table 5 below). (Sahay, et. al, 2003, Arora and Gambardella, 2005). However, China and Brazil mainly produce software services for the domestic market, Ireland develops products and services for Europe, although much of it is localization of US software products. Russia, the Philippines and Vietnam are, like

India, primarily producers of software services for export, although Russia exports mostly to Europe while the others export mostly to the US. Israel is the only producer of software products for the US and global markets. (See Table 6)

Of the above, the significant producers of offshored software, i.e., for global markets, are India and Israel, while Ireland is the most significant producer of localization products and services. Israel focuses on software products for the global market while India focuses on custom software for the global market.

In Ireland, policymakers offered incentives for global software firms only in the late 1970s following Ireland's entry into the EC in 1973 (*Enterprise Ireland*, and Torrisi, 2002, p. 17). Software offshoring began in the 1980s, and followed hardware offshoring, which had begun in the early 1970s (Torrisi, 2002, p.17). It was begun – and continues to be dominated – by American transnational corporations (TNCs) that use Ireland as a low value-added packaging gateway into Europe for their software products (Torrisi, ibid, p.18) and account for about 90% of software exports (Arora et al., 2001, p. 7).[5]

An indigenous software sector has developed since the 1990s, initially providing support services for the MNCs but subsequently developing products for the European telecom and finance sectors. In 2003, the indigenous sector employed 40% of the total (Sands, 2005, p.44).

Israel's software industry grew out of its successful hardware industry that was established by American TNCs, in the 1960s. For example, Motorola's first offshore manufacturing subsidiary was set up in Israel in 1964 (Ariav and Goodman, 1994, quoted in Sahay, et.al., 2003, p.15)

Israeli policymakers offered incentives for global software firms in the early 1970s, helping to induce TNC entry (Torrisi, 2002, p.18). These TNCs used local programmers initially for software product maintenance and later for R&D. Domestic software firms were also established due to a growing domestic market from the local military-industrial complex. The work focused on writing product software and on R&D. Much of the labor force had earlier worked in the defence industry. In the 1980s, domestic software startups, funded by government research contracts, were established. Initially, they provided software services to the defence industry and later developed security software products for global markets. This trend continued into the 1990s with support from global VC firms. (Teubal, 2002, p.166). TNCs currently account for about 25% of total employment in the IT industry and the industry's growth is driven by local firms producing software products for export markets (Torrisi, ibid, p.9 and 18). The three largest software firms are product firms that jointly account for 60% of the industry's revenue (compiled from Bresnitz in Arora & Gambardella, 2005, p.77).

Thus, from the above, it appears that the significant growth in offshoring to global markets is primarily occurring in India. The rest of this section, therefore, focuses on the Indian software industry. (See Table 8.)

Unlike Ireland and Israel, where government policy was an important incentive to induce entry, the Indian software began during a time when Indian policy was appropriately described as "statist, protectionist and regulatory." (Rubin, 1985) An industrial licensing regime and state-owned banks were used to strictly regulate private sector activity. In IT, the state was the main producer of products and services. Its strategy was to create 'national champion' state-owned enterprises (Sridharan, 1996). These were granted monopolies.

A key protectionist policy was the Foreign Exchange Regulation Act of 1973 (FERA-1973). Under FERA-1973, a foreign firm could operate in India only with a minority interest (foreign ownership was restricted to 40%). Many foreign firms closed their Indian operations, including firms as diverse as Coca-Cola and IBM (which supplied hardware and software for local markets), citing concerns about the protection of intellectual property. FERA-1973 thus closed the door to software development by TNCs.

Domestic firms discovered an innovative solution to benefit from global opportunities for ISVs. Since software development could not come to India, Indian programmers went to developed countries. It began in 1974 with the mainframe manufacturer, Burroughs, asking its India sales agent, Tata Consultancy Services, to supply programmers for installing system software for a U.S. client (Ramadorai, 2002). Other firms followed, including foreign IT firms that formed FERA-1973 compatible joint ventures.[6] Initially, the exported programmers worked for global IT firms. Later in the decade, as IBM's total global market share increased, end-users such as banks used Indian firms to convert existing applications software into IBM-compatible versions.

The state remained hostile or, at best, indifferent to the software industry through the 1970s. Import tariffs were high (135% on hardware and 100% on software) and software was not considered an "industry", so that exporters were ineligible for bank finance. Even opening an overseas sales office was disallowed until 1979 (Ramadorai, 2002).

Protectionism hampered learning thus preventing movement up the value-chain. Programmers returning from overseas assignments were the main source of learning about new opportunities, but their short duration – typically less than a year – limited the depth of the learning (Ramadorai, 2002). Also, many chose to remain overseas after completing their assignments. As a result, the activity of the industry during its first decade consisted of little other than recruitment of engineers.

The anti-private sector policies of the state actually favored established private conglomerates who could access finance more easily than small firms. As a result, Mumbai, the country's commercial capital, became the center of the business. 7 of the top 8 exporters in 1980 were headquartered in Mumbai with a 90% market share (see the Table 9 below).

The industry's activities changed in the mid-1980s with the global industry's adoption of the U-W standard. As we have discussed above, programming was modularized as a result.

The state also at that time began a gradual abandonment of its protectionist, anti-TNC stance. The New Computer Policy of 1984 was a key element. NCP-1984 consisted of a package of reduced import tariffs on hardware and software (reduced to 60%), recognition of software exports as a "delicensed industry", i.e., henceforth eligible for bank finance but not subject to the intrusive licensing regime (Heeks, 1996, p. 44-5), permission for foreign firms to set up wholly-owned, export-dedicated units and a project to set up a chain of software parks that would offer infrastructure at below-market costs. In 1985, all export revenue (including software exports) was exempted from income-tax.

The new policies led to considerable new entry by TNCs, who introduced new businesses and new business models. Some TNCs did R&D and wrote product software using cross-country teams (such as Texas Instruments and Hewlett Packard), others wrote custom software for inhouse use, again using cross-country teams (such as ANZ Bank and Citigroup). The TNCs thus replicated successful approaches used by them in other environments, such as Ireland and Israel (see above, this section). However, facing daunting communications costs and intrusive regulation (Parthasarathy, 2000), the product-focused TNCs remained small, although the initial entrants, such as Texas Instruments, were important in persuading the government to improve the infrastructure.[7] Domestic firms that sought to imitate the TNC product-software model, such as Wipro, also failed, due to the weakness of domestic markets as a source of domain expertise (Athreye, 2005) and the absence of a venture capital industry. In consequence, product development accounted for less than 5% of exports by 1990 (Heeks, 1996, p. 88-89) and reached only 8% by 1999 (Nasscom, 2002, p.28).

However, the combination of the U-W standard and lower costs engendered a successful new business model, pioneered by TCS. Domestic firms began to supply software programs coded entirely in India, while relying on a foreign co-vendor for program design and specification. This approach succeeded because it matched the expertise of Indian firms (programming) with that of overseas vendors (client understanding, design and intregration) and because it reduced costs by keeping programmers at home – although the decline in the personnel dispatched overseas was initially slow.[8] Indian firms thus shifted from exporting programmers to programming outsourced custom software in India. The shift, though gradual, induced considerable entry by domestic firms. The number of software firms went from 35 in 1984 to 700 in 1990 and the share of smaller firms rose (Table 7).

The shift for the first time raised the amount and standards of the required physical infrastructure in India. This marked a turning point in the development of Bangalore as a center for software development, as real estate costs were cheaper than Mumbai. This induced several new firms, including Infosys and Wipro (Premji, 2003), to locate to Bangalore. The first software technology park under NCP-1984, with assured supply of electricity and telecommunications bandwidth, was located in Bangalore.

Bangalore's other advantages over competing locations included low labor costs and, unlike Mumbai and Delhi, with histories of large firms and labor militancy, Bangalore had small companies that were relatively free of union troubles (Heitzman, 1999, p. 6). Further, Bangalore is located at the center of the four southern states, Karnataka (whose capital is Bangalore), Tamil Nadu, Andhra Pradesh and Kerala, which together produce 52% of India's engineering graduates. Bangalore's best known academic institution was the elite Indian Institute of Science (IIS), which had been established in 1909. While most of its graduates and research were directed towards the public sector, some of these, if indirectly, helped Bangalore's development in software. This was because the government had earlier chosen Bangalore to locate several high technology SOEs, thus creating a trained labor force (Balasubramanyam, et. al., 2000, p. 351) – although, according to some industry observers, the quality of the labor force was dubious and provided only a small percentage of the software industry's needs (Ramadorai, 2002). The biggest success from IIS was Wipro Technologies, India's 3rd largest software exporter. It was founded at IIS by a group of engineers working under Ashok Soota, an academic at IIS (Parthasarathy, 2003).

Policy reforms in the 1990s and 2000 reduced import tariffs to near zero[9] and standardized foreign ownership, intellectual property protection, venture capital, stockmarket listing and telecommunications policies to global best practices. In addition, technological changes during this period, particularly the Internet, led to a sharp decline in data storage and transmission costs. These changes induced a new round of entry of TNCs, particularly foreign outsourcers, and US-based startups and opened new opportunities for existing firms in remote software services, such as email management and remote software maintenance (Table 4).

Interestingly, the TNCs initially adopted the domestic firms' approach of focusing on programming only, i.e., they adopted the TCS-pioneered remote programming method. This was for both inhouse product development by firms such as TI, Agilent, HP, Oracle and GE, as well as services by firms such as ANZ Bank, ABN Amro Bank, Accenture, IBM and Dell. TNCs and foreign startups overwhelmingly chose Bangalore for their IT operations (Naidu, 2006).

The entry of TNCs and overseas startups, over time, led to a rising sophistication of work done in India. As shown in the table below, the share of routine programming work and maintenance accounted for 68.9% of total export revenue in 2001. By 2005, this had fallen to 58.5%. During this period, the share of foreign firms in total revenue rose from 14.5% to 31%. We argue that the relationship was causal, i.e., the declining share of routine work was caused by the entry of foreign firms doing more sophisticated work.[10]

As the share of routine programming work declined, the share of engineering services, research and development and product development rose from 8% in 1999 to 23% in 2005 (Nasscom, 2002, p.28 and Nasscom, 2006, p.47). Of course, several domestic firms also do high-end work. Wipro, the third largest domestic firm, has 14,000 employees providing contract R&D services and filed 68 US patents on behalf of overseas clients in 2005 (Premji, 2006). (See Table 10.)

Although no comprehensive study of value-added work in offshored software development has been done, evidence from case studies supports the sectoral data above. We present some studies below. From their experiences, it appears that the key domestic constraints in doing higher value-added work appear to be the recruitment and retention of qualified persons and small domestic markets. The former in turn derive from earlier problems with educational policy and minimal university-industry interactions (Parthasarathi and Joseph, 2002, p.32). Until recently, faculty at even the best engineering institutions, almost all of which are public universities, were not required to do research. Those that chose to do so faced, according to the government's own reckoning, severe problems: "obsolescence of facilities and infrastructure are experienced in many institutions … the IT infrastructure and the use of IT in technical institutions is woefully inadequate … the barest minimum laboratory facilities are available in many of the institutions and very little research activity is undertaken … engineering institutes have not succeeded in developing strong linkages with industry … the curriculum offered is outdated and does not meet the needs of the labor market" (Ministry of HRD, 2001, Sections

2.1.2-2.1.6). The best students often migrate (Siwek and Furchtgott-Roth, 1993, p.140), although, with increasing opportunities at home, this is likely to lessen.

Small domestic markets have also limited the ability to move up the value-chain. As Rosenberg and Mowery (1978) have argued in a more general context, vendors become technologically sophisticated through understanding customer preferences. D'Costa (2002a, p.705) has criticized the Indian software industry's dependence on exports, arguing that international outsourcing of software, though commercially lucrative, discouraged firms from doing more complex projects at home because "excessive dependence on outsourcing limits the synergy between vibrant domestic and foreign markets".

Section 2.2 Case Studies of Offshoring Product Development by U.S. Firms to India

For purposes of this discussion, we consider software product development by two types of firms, (1) Startups and (2) Established firms. The former are dependent on venture capital and tend to be staffed very tightly. For startups, offshoring leads to high coordination costs as a share of total costs. The latter have established sources of revenue, a more reliable labor pool and, often, an interest in establishing a base in China or India for accessing domestic markets. and may choose to use offshoring as a non-integral part of product development, for purposes such as product upgrades or second generation product maintenance. Both types are known to use outsourcing as a strategy rather than do the work inhouse, despite concerns about the protection of intellectual property, labor force control and management efficiency (Mukerji, 2006). Offshoring of product development (including engineering and R&D services), whether outsourced or done inhouse was estimated to be a $ 8 bn industry in 2005 (Nasscom, 2006, p.96) or about 4% of the software product industry. In 2005, India was the largest participant, generating revenue of $3.9 bn in this segment, followed by Israel, with $750 m.[11]

Case Study: Agilent Technologies[12]

Agilent, a maker of test and measurement equipment, chose India as a base for software development in 2001. Its choice of India was based on the potential talent pool, and on the judicial system, which it deemed mature, favorable intellectual protection rules compared with other developing countries in Asia and mature management talent. Concerns about IP protection and managerial control drove its choice to do most of the work inhouse rather than outsource (some software maintenance and programming work was outsourced). To manage these concerns and also concerns about reversibility of work in the event of failure, all offshored operations had a six month overlap in staffing between the US and India.

The work began with simple activities and moved to more complex activities over time(see Figure 2). The engineering services group was the first user of the Indian operations. The initial work done was to provide parts lists to customers worldwide and data entry for the CAD group in the US. Over time, most support services were moved to the Indian operations.

In early 2002, the second user within Agilent, the communications solutions group, established a 10-person team to automate test suites for one of Agilent's projects, Netexpert. However, a lack of coordination between the Indian and US teams led to the initial failure of this experiment. Increasing the time allocated for coordination and introducing a quality enhancement program in the Indian operations led to an improvement. By 2005, the development and maintenance of Agilent's EDA software products were being done jointly by multi-country teams located in both the US and India.

Case Study: Broadcom

Broadcom, a Silicon Valley based fabless chip firm, acquired an India operation through the acquisition of Armedia Labs. The Silicon Valley based Armedia Labs was founded in 1997 to develop a single-chip (popularly, system-on-a-chip, or SOC) solution for HDTV. From its inception, work in Silicon Valley was tightly integrated with its Bangalore subsidiary, except for market development, for which the Silicon Valley team took responsibility (Khare, 2006). This meant that all other work, such as design and developing the embedded software and libraries were shared. When Broadcom acquired Armedia in 1999, its 25 person Indian subsidiary became Broadcom India. Broadcom expanded the team and brought in complementary technology for SOC work, such as in graphics and digital conversion and processing. By 2006, the team in Bangalore had grown to 190 persons. Employees were, as in the firm's San Jose offices, divided into functional teams and each functional team was part of a global team that consisted of engineers in San Jose and Irvine, USA, Israel, Andover MA and Singapore.

As of 2006, the process of product development is driven by the Engineering Director of the project, based in San Jose, working with the marketing team, based in Irvine. The team members might consist at any one time of over 100 people located in various places. These travel as needed across the various locations. The final chip integration design (tapeout), which may take upto two months is always done at one location because of the need for close coordination. This initially used to be either San Jose or Irvine, but is increasingly done in Bangalore. Early in the chip development process, one of these three locations takes the lead.

The logic for allowing Bangalore to share the lead position in product development, a status not granted to other locations (including Broadcom's offices in Andover, Israel and Singapore), is that of scale and capability. The Indian team had, between 2003 and 2005, filed for 140 US patents and been granted 10 patents. From 2006 onwards, the firm expected that the Indian team would be granted a 25 to 30 patents annually. According to Khare, this compared well with optimal US patent rates.

Interestingly, despite the progress with the Bangalore team, proximity still matters in some cases. Once a chip has been fully designed, i.e., after tapeout, software libraries and firmware are needed to accommodate the specific requirements of different types of customers. These may also change considerably during and after the product's first release. Understanding customer needs was found difficult to undertake from Bangalore (Khare, 2006). Hence, in the event that the project is led by Bangalore, one member of the Bangalore team is located in the US on an 8-week rotation after the first release and until maturity.

Khare notes that the main locational challenge is that it takes time to build respect across teams and that building a large enough team for chip development can take time due to the high level of skills needed. By comparison with Silicon Valley, where building a 100 person skilled team of ASIC designers might take upto 18 months, a similar time frame in India would generate a considerably less-skilled team. To improve skill levels, Broadcom India also recruits engineers from the US, mostly of Indian origin, as a result of which about 5% of its Indian workforce are expatriates of Indian origin. The Indian recruits initially were experienced engineers hired away from competitors, but due to low attrition, Broadcom India's average work experience per engineer has crossed nine years, enabling it to recruit from universities and to offer internships to university students.

The payoffs of this hybrid approach are: (a) lower costs in India despite expatriate recruitment, averaging one third of US costs; (b) growing center of expertise in Broadcom India and in Bangalore in embedded software and very large chip development.

Case Study: Hellosoft

Hellosoft is a Silicon Valley startup established in 2000 and funded by U.S., Taiwanese, and Indian venture capitalists. It provides high-performance communications intellectual property for VoIP and wireless devices. It was intended from the beginning that the firm would use Indian engineers to create the firm's intellectual property. All R&D is undertaken in a subsidiary located in Hyderabad, India that employs over 100 digital signal-processing engineers (Hellosoft 2005). The Hyderabad center develops software for advanced cell phones and networking technologies. Marketing and sales operate from its San Jose headquarters.

Case study: Ketera Technologies[13]

Ketera provides inventory management software on-demand, i.e., software-as-a-service and is headquartered in Santa Clara, CA. As of 2005, it had 150 employees worldwide. Its objective of using India was to both cut costs and speed up time-to-market. In 2002, it began a relationship with an Indian vendor. Headcount with the vendor peaked at 105 in June 2004. The engineers in the India operations worked on software development and more mundane tasks, such as configuring the software for customers and other support services.

The relationship with the vendor was unsatisfactory: the engineers provided were relatively unproductive and had high attrition rates. The US operation added to the problem: it was understaffed as a result of the 2001-2003 downturn, so that there was only one architect for about eighty engineers.

In late 2004, the firm created its Indian subsidiary and transferred the work in phases, beginning with software development. It also decided to undertake product management from India. To ease the coordination problems, it added staff in the US.

It took about nine months for Ketera to hire 75 engineers in Bangalore. Coordination was key to raising value-addition. For example, the work of product management was divided between the US and India teams, the US team taking responsibility for market requirements and the India teams converting those into product specifications.

A key challenge in new product development remains that of measuring team productivity. Unlike well-specified software, where productivity can be measured by error rates or lines of code, in a new product release, there is always a 'new level of complexity associated with it' (Shah, 2005), making measurement difficult.

Case Study: Netscaler[14]

Netscaler was founded in 1998 to redesign a specific piece of infrastructure used in regulating Internet traffic flow. After Netscaler developed the product, Netscaler found that it needed to add other features to attract customers unsure about moving from legacy products to their product. Netscaler understood that it could increase the acceptance of its product if it could offer greater functionality. Because Netscaler was constrained financially and needed to cut costs, in 2001 it hired an Indian outsourcing firm, NodeInfoTech, to help develop the new features.

The success with this contract convinced Netscaler to establish Netscaler India. To staff the new operation, Netscaler hired many of the developers from NodeInfoTech (Tillman and Blasgen 2005). In 2004 Netscaler India employed approximately 60 engineers to develop other features and had plans to grow to 200 total employees in 2005 (Hindu Business Line, 2004), when it was purchased by Citrix Systems for $300 million.

The reason Netscaler formed a subsidiary rather than continue with outsourcing was to increase the number and sophistication of projects done in India and encourage tighter engineering integration (Tillman and Blasgen 2005). After its initial foray into India, Netscaler offshored high value work to its subsidiary, and outsourced some lower-level engineering support to local Indian vendors. Having Indian and US internal engineering teams made it possible for Netscaler to provide all levels of support 24 hours a day. As the Indian team grew, it became feasible to add a technical writer in India to provide software documentation to complement the one in the U.S.

Case Study: Tensilica[15]

Tensilica is a Silicon Valley startup established in 1997. It has 120 employees worldwide. The firm develops and licenses its embedded processor technology to system-on-a-chip suppliers.

The downturn of 2001 in the context of growing demand for Tensilica's products led the firm to consider shifting second-generation work, such as adding features and improving product reliability, to India, thus freeing up expensive US-based engineer resources for new product development. To save on initial set-up costs, the firm decided to begin working with a vendor, eInfochips, and then transfer the work to a subsidiary over time. The firm was also concerned that it lacked a brand name in India to be able to recruit the best talent.

The initial work was the addition of features to an existing product, such as improving the graphical user interface. An experiment with also doing quality assurance work was unsuccessful owing to requiring too much US management time. In general, the coordination costs have been much higher than expected.

Recruitment was much more difficult than expected. Einfochips agreed to let Tensilica drive the recruitment process; Tensilica found that the level of skills available was substandard. Also, some qualified engineers were unwilling to work for an outsourcer.

In January 2006, Tensilica transferred the engineers to its own subsidiary, which, as of September 2006 employs 15 persons, or 12% of its workforce. The experience of working with the India team for a year has greatly reduced coordination costs. Recruitment has also been easier without the veil of an outsourcer and attrition rates have fallen. The firm states that it has been able to increase the complexity of work done in India as a result.

Implications of case studies

Extrapolating from this admittedly small base of information, we have found two basic models: In the first model, offshoring is a supplement to onshore operations, i.e, the offshore facility is to lower their costs and/or to speed product or product line extensions. In the second model, offshore operations are an integral part of the business model. In both cases, the ultimate goal for the India business appears to be to make them an integral part.

Interestingly, both startups and established firms often choose to begin by using an outsourced provider rather than establish their own facilities. The advantage of outsourcing is that it can facilitate a rapid ramp-up. Also, through the operation of the outsourcer it may be possible to learn more about the Indian environment facilitating the establishment of a subsidiary. There are risks to this approach, as one cedes control over the labor force to an outside vendor and there are risks of losing IP. The ultimate goal, as noted above, for both new and established firms, appears to be to establish a subsidiary. Established firms have less critical cost concerns and are more likely to begin with inhouse work right away.

In all cases, coordination costs have been surprisingly high. The fault for this does not lie in inadequate communications facilities but appears to lie in the complex nature of the work. The other primary constraint to doing higher value-added work is the difficulty in finding and retaining qualified persons. As discussed earlier, this likely arises from the small size of domestic markets and India’s inadequate educational system. The following table summarizes the stages of offshoring described in the above cases. Undoubtedly, there will be further stages of evolution. For example, Agilent India plans to increase outsourcing once the offshoring process is stabilized. (See Table 11.)

Section 3. A theoretical framework of the drivers of software offshoring

The novelty of services offshoring in international trade warrants some definitional classifications, as basic as what a “service” is. Most persons would agree that manufacturing is a process that involves the transformation of a tangible good. In terms of the location and discharge of manufacturing, it would also generally be agreed that manufacturing does not require face-to-face contact between buyer and seller. Usually, manufacturing creates a good that can be stored, thereby allowing a physical separation of the buyer and the seller.

Services have been defined by the opposites to these qualitites: they are transactions of intangible, nonstorable goods, requiring that client and vendor be face-to-face while the service is being delivered. For example, Gadfrey and Gallouj (1998) define services as goods that are "intangible, cosubstantial (i.e., they cannot be held in stock) and coproduced (i.e., their production/consumption requires cooperation between users and producers)." This was obviously true when the service required face-to-face experience, such as receiving a haircut (a "customized" service), but also true when the "service experience" did not require customization, such as when a bank's client wanted to check the bank's home loan offering, or did not require proximity, as when she wanted to check her bank balance.

This suggests that services are intrinsically more difficult to offshore than manufactured goods. When a service activity is examined as a totality, indeed it appears to resist relocation. In fact, very few service operations can be done only on the computer (the modern form of "mundane work"). Most services require at least some level of face-to-face interactivity, either among co-workers or with persons outside the organization, such as vendors and clients.

Following Bhagwati (1985), we can classify a service into those that require proximity to a user and those that do not. Services that require physical proximity are of three types:

1. Mobile user-immobile provider, eg., a patient going to a hospital
2. Immobile user-mobile provider, eg., a doctor visiting a patient at home
3. Mobile user-mobile provider, eg., students and professors meeting in a university classroom for lectures.

The new twist in the provision of services is that the required interaction between the seller and the consumer has been substantially limited. The advances in information technology made possible the parsing of the provision of certain services into components requiring different levels of skill and interactivity. As a result, certain portions of the serviced activity – that might or might not be skill-intensive, but required low levels of face-to-face interactivity - could be relocated offshore.

The sequence of events that enabled this process is the following: First, the digital age allowed (or, at least, revolutionized) the conversion of service flows into stocks of information, making it possible to store a service. For example, a legal opinion that earlier had to be delivered to the client in person could now be prepared as a computer document and transmitted to the client over email or, better yet, encoded into software Easy storage and transmission allowed for the physical separation of the client and vendor as well as their separation in time. It also induced the separation of services into components that were standardized and could be prepared in advance (such as a template for a legal opinion) and other components that were customized for the client (such as the opinion itself) or remained non-storable. Taking advantage of the possibility of subdividing tasks and the economies that come with the division of labor, this reduced costs by offering the possibility of preparing the standardized components with lower-cost labor and, possibly, at another location.

The second fundamental impact of digitization was the conversion of non-information service flows into information service flows. For example, the sampling of tangible goods by a buyer through visiting a showroom is increasingly being replaced by virtual samples delivered over the Internet. Once converted to an information flow, the service may also then be converted into a stock of information, as noted earlier, and subjected to the above mentioned forces of cost reduction through standardization of components and remote production.

Third, by enabling low cost transmission of the digitized material, the digital age accelerated the offshoring of services. Services such as writing software programs which were offshored to India in the early 1970s were enabled by digitized storage and, in the 1980s, by the standardization of programming languages. Still later, as digital transmission costs fell in the 1990s (just as digital storage costs had fallen earlier), even non-storable services, such as customer care, could be offshored.

The sequence of events that enabled software offshoring did not happen all at once and may not even have a standardized sequence across countries. Israel, for example, was able to quickly move up to product development for global markets undertaken by domestic firms. India, till a few years ago, offered only routine programming work for over two decades. As of 2006, there was no evidence of successful product development originated from India, although work to support product development conceived in developed countries was happening, as we have evidenced.

Thus, moving to higher stages of work is not automatic, sequential or time-bound. Nor can we, based on the evidence we have, provide conditions for movement to higher stages, or predict that the exporter will capture a rising share of the economic rents.

At the very least, it seems from our case studies, that a factor that can hinder movement to higher stages is that costly global coordination is needed. Such coordination will initially be done by the developed country buyer and enable it to earn a rent for doing so. In addition, much of the market-related coordination and networking will occur through developed country institutions, enabling a further retention of value in the developed country.

The inference is that certain key aspects, such as deciding on the product and its specification, design, marketing and sales are usually retained by the importer. Such retention then appears to hinder the exporter's ability to rise to new stages of growth and to result in the rewards being overwhelmingly gathered by the developed country buyer.

Section 4. Concluding thoughts on the impact of software offshoring on employment, innovation and policy

It is tempting to view software offshoring as the cause of an unmitigated loss of jobs for US workers. In fact, much of the political alarm that software offshoring raises is that skilled jobs will rapidly disappear from US shores due to the digital age. This could not only leave the US digitally divided from other countries, but centralize demand for US workers on non-offshoreable jobs. In services, the argument is made that, ultimately, US workers will only do those jobs that are impossible to offshore, a few of which are highly-skilled, such as wielding the surgeon's knife, and mostly those that require a lower order of skills, such as wielding the butcher's knife. Is there something about software offshoring that suggests a different trajectory?

Our analysis of the software industry shows that offshoring's effects on developed nation employment are varied, even though the impact of software offshoring on developing countries is to generate high and growing levels of employment. The kinds of work initially offshored are typically those with low barriers to entry and subject to automation. Thus, services exports from developing countries initially lack brand value and are quite different from services exports from developed countries. These factors lead to competition and price-deflation in these sectors. However, over time, the sophistication of work being done has grown rapidly, particularly with the advent of the digital revolution. As Shah (2005) notes in his discussion of Ketera's offshoring, discussed above, "The primary challenge (of offshoring most of the headcount to India) was the lack of informal communication. We miss the informal hallway and coffee station side chats. We miss going to the white-board and brain-storming an idea." After observing the progress of the Indian operation, he concluded, "The hallway discussion and white-board brainstorming are actually happening now (in our firm), but in India."

In summary, we found that, in software offshoring, work that is modularized and standardized, and does not require regular customer contact is more prone to offshoring. Even when customer interface is needed, as the case of Broadcom India showed, it is possible to manage customer interface remotely through a new kind of body-shopping, that which is done to understand customers rather than, as in the old days, to access customers' software and hardware. In this case, offshore workers are substitutes for US workers.

However, lowering the costs of some parts of software development lowers total firm costs and raises the firm's competitivity globally. It can also make possible the creation of new firms that would otherwise not be economically viable. The case of Netscaler showed both these aspects. Any jobs created by this entrepreneurship would be counted against those lost to offshoring. For example, Rakesh Singh, Netscaler's General Manager of Asia Operations said, "The cost savings through outsourcing have helped us become more competitive and experience rapid growth as a company. As a result, we have a lot more employees in the US today than we did when we set up the India operations" (Tillman and Blasgen 2005). In this case, offshore workers are complements to US workers.

The ongoing process of technological development that is a feature of the software industry can both speed up as well as slow down job losses. For example, prior to the establishment of the Internet as a reliable medium of digital communication, installing software or fixing a software problem required an onsite technician. Now, this can often be done remotely, thus reducing the need for onsite work and increasing the demand for offshore maintenance. Similarly, the invention of the router led to the creation of remote data centers, thus reducing the need for on-site storage hardware and support services.

On the other hand, the Internet enables access to many more software applications developed elsewhere, including open-sourced applications. Raza (2005) notes that chip designers that used to offshore components of chip development to vendors in India can now usually find some components already available in open source, thus reducing the need for offshoring such work (although this does not add to demand for US software developers).

An alternate view of the impact of technological change takes the view that since the developers of new technology are mostly in the developed nations, a faster rate of technological progress is advantageous to employment in developed nations because it makes it harder for developing countries to catch up. For example, the ability of developed nations semiconductor foundries to stay a generation ahead of those in East Asia means that the location of (and, therefore, employment in) the latest generation of 'fabs' is in developed nations. Of course, the pace of technological development is closely linked to changes in customer requirements.

Under this viewpoint, anything that helps developed country engineers to innovate more quickly and efficiently is a plus for the developed country. Hence, offshoring software development that is a step behind the quality of work done in the developed country enables engineers in developed nations to innovate even more and is good for both developing and developed nations alike.

As we noted in the introduction, equilibrium theorists concede that the effect of offshoring on the quality of work done within the developed nation is ambiguous since it is not known who will capture the productivity gains – the developing country or the developed country. It depends on relative productivity gains in the two types. Hence, most would be ready to concede that some of the work that developed nations workers will be left to do as a result of offshoring will certainly include low wage work in which technology does not enable remote provision (such as the physical installation of a hard-wired network) and also concede the possibility of short-term unemployment. However, they argue that much of the new work will use higher skills than available in developing countries, will pay more and can even leverage the work being done in developing countries.

Related to what we may learn from manufacturing, a second issue is the speed with which services offshoring will happen. Manufacturing's decline in the United States was a gradual process and was accompanied by a rising revenue per employee, reflecting in part that, while the more commoditized parts of manufacturing were being outsourced offshore, the more customized or specialized parts and some service components, such as design and integration, were being done onshore (see Figure 3 and Table 12). The slow speed of manufacturing offshoring also enabled displaced workers to acquire the skills to shift to other occupations.

It has been argued that services' offshoring will take place at a higher speed, leaving less time for labor-force adjustment. This is attributed to the digital age that firmly established itself in the mid-1990s (the Telecom Regulation Act of 1996 is often viewed as a key driver). The effect of the digital age on the speed of services offshoring can be impressive. Unburdened by the need to set up large factories, it is likely that offshored services can be set up as rapidly as workers with the requisite skills can be hired. Certainly, the rates of growth of the Indian IT industry shows trajectories that were never achieved in manufacturing offshoring.

This raises the question as to whether the digital revolution has done more for Asian competitivity than provide a one-time boost to comparative advantage by simply opening a previously closed door. However, there are two reasons, apart from the labor-cost advantage, why comparative advantage will continue:

1. Economies of scale and scope. Countries like India offer large labor pools that could offer significant labor economies over thinner pools or country-specific pools. In addition, locating software developers in Indian programmers allows the vendor to supply services for clients located in different time zones, thus using capital and real estate more efficiently. Or, they could manage episodic peak requirements, such as when a new upgrade of software is released, over many clients more efficiently.

2. Specialization. Many efficient practices for offshore software development that came out of the remote software programming businesses have been developed in India. Such remote management is emerging to be a specialized skill, applicable in a variety of other offshoring situations, such as providing R&D and product development services. Of course, Indian firms with such skills must compete with the remote project management skills developed by global firms in other environments, for example, Accenture in system integration.

It appears that the only viable strategy for the developed nation is to develop the capacity to constantly generate high-value new opportunities that cannot be offshored. This requires innovation. While there is no guarantee that a developed economy will have such a capacity, an open economy that invests in education, has a better chance than otherwise. The US has long demonstrated the truth of this proposition.

References

Apte, U., and R. Mason. 1995. Global disaggregation of information-intensive services, *Management Science,* 41.

Arora, A. and S. Athreye. 2002. The software industry and India's economics development, *Information Economics and Policy,* 14(2), 253–273.

Arora, A. and A. Gambardella. 2005. From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel. Oxford: Oxford University Press.

Athreye, S. 2005. The Indian software industry and its evolving service capability, *Industrial and Corporate Change,* 14: 393–418.

Balasubramanyam, V., and A. Balasubramanyam. 2000. The software cluster in Bangalore, in Dunning (ed.), *Regions, Globalization and the knowledge-based economy*, Oxford: Oxford University Press.

Bhagwati, J. 1985. Why are Services Cheaper in the Poor Countries, in G. Grossman (ed.), *Wealth and Poverty*, Cambridge: MIT Press.

Bhagwati, J., A. Panagariya, and T. Srinivasan. 2004. The Muddles over Outsourcing, *Journal of Economic Perspectives* (Fall).

Collins, S., and L. Brainard. 2006. Offshoring White-Collar Work. Washington DC: Brookings Institution.

Correa, C. 1996. Strategies for software exports from developing countries, *World Development*, 24(1), 171–82.

D'Costa, A. 2000. Technology Leapfrogging: the software challenge in India, in Conceicao et al. (eds), *Knowledge for inclusive development,* Quorum Books: Westport.

D'Costa, A. 2002. Software outsourcing and development policy implications: an Indian perspective, *International Journal of Technology Management,* 24(7/8), 705–23.

D'Costa, A. 2002. Export growth and path dependence: the locking-in of innovations in the software industry, *Science, Technology and Society*, 7(1), 51–87.

Dixit, A. 2005. Tensilica's India Operations: The First Seven Months of Embedded Processor Engineering Offshore. Paper presented at Stanford University Conference on Globalization of Services, June 2005.

Dossani, R. 2006. Globalization and the Offshoring of Services: the Case of India, in Collins, S. and L. Brainard, *Offshoring White-Collar Work*, pp. 241–67. Washington, D.C.: Brookings Institution.

Dossani, R., and A. Manwani. 2005. Agilent's Supply Chain: A Locational Analysis of its Indian Operations. Paper presented at Stanford University Conference on Globalization of Services, June 2005.

Feller, J., B. Fitzgerald, S. Hissam, and K. Lakhani. 2005. Perspectives on Free and Open Source Software. Cambridge: MIT Press.

Gadrey, J., and F. Gallouj. 1998. The provider-customer interface in business and professional services, *The Services Industries Journal*, 18(2), 1–15.

Gomory, R., and W. Baumol. 2000. Global Trade and Conflicting National Interests. Cambridge: MIT Press.

Hanson, G., R. Mataloni Jr., and M. Slaughter. 2001. Expansion Strategies of U.S. Multinational Firms, in Collins S. and D. Rodrik, *Brookings Trade Forum*, Washington, D.C.: Brookings Institution.

Heeks, R. 1996. India's Software Industry. New Delhi: Sage.

Heitzman, J. 1999. Corporate Strategy and Planning in the Science City: Bangalore as Silicon Valley, *Economic and Political Weekly*, January 30, 1999.

IT Workforce. 1999. Assessing the demand for information technology workers', National Science Foundation: Washington, D.C..

Khare, R. 2006. CEO of Broadcom, India. Personal Interview with Rafiq Dossani, 2/1/06.

Mann, C. 2006. Accelerating the Globalization of America: The Role for Information Technology. Washington, D.C.: Institute for International Economics.

Mckenna, C. 2006. The World's Newest Profession. Cambridge, UK: Cambridge University Press.

Indian Ministry of Finance. 2000. Finance Minister's Budget Speech. New Delhi: Ministry of Finance.

Landefeld, S., and R. Mataloni. 2004. Offshore Outsourcing and Multinational Companies. Working Paper, Brookings Institution.

Mankiw, G., and P. Swagel. 2006. The Politics and Economics of Offshore Outsourcing. Working Paper, American Enterprise Institute.

McKenna, C. 2006. The World's Newest Profession. Cambridge: Cambridge University Press.

Mowery, D. (ed.). 1996. The international computer software industry: a comparative study of industry evolution and structure, Oxford University Press: Oxford and New York.

Mukerji, A. 2006. SVP, Wipro Technologies. Personal Interview with Rafiq Dossani, 8/1/06/.

Naidu, B. 2002. Personal Interview with Rafiq Dossani, 2/15/02.

Nasscom. 2002–2006. The IT Industry in India: A Strategic Review. New Delhi: Nasscom.

Parthasarathi, A., and K. Joseph. 2002. 'Limits to Innovation in India's ICT Sector', *Science, Technology and Society*, 7(1), 13–50.

Premji, A. 2003. Chairperson of Wipro. Personal communication with Rafiq Dossani, 12/2/03.

Ramadorai, S. 2002. CEO of TCS. Personal communication with Rafiq Dossani, 11/29/02.

Raza, A. 2005. CEO of Raza Foundries. Communication with Rafiq Dossani, 2/12/05.

Rosenberg, N., and D. Mowery. 1978. The Influence of Market Demand upon Innovation: a critical review of recent empirical studies, *Research Policy*, 8.

Rubin, B. 1985. Economic liberalization and the Indian state, *Third World Quarterly*, 7(4), 942–957.

Sahay, S., B. Nicholson, and S. Krishna. 2003. Global IT Outsourcing. Cambridge: Cambridge University Press.

Samuelson, P. 2004. Why Ricardo and Mill Rebut and Confirm Arguments of Mainstream Economists Supporting Globalization, *Journal of Economic Perspectives* (Summer).

Scholte, J. 2000. *Globalization: A Critical Introduction*, Palgrave Publishers: Basingstoke.

Schware, R. 1992. Software Industry entry strategies for developing countries: a "walking on two legs" proposition, *World Development*, 20(2), 143–164.

Shah, R. 2005. Ketera India Case Study. Paper presented at Stanford University Conference on Globalization of Services, June 2005.

Steinmuller, W. 1996. The U.S. Software Industry: An Analysis and Interpretive History, in D. Mowery (ed.), *The international computer software industry: a comparative study of industry evolution and structure*, Oxford: Oxford University Press.

Teubal, M. 2002. The Indian Software Industry from an Israeli Perspective: A Microeconomic and Policy Analysis, *Science, Technology and Society,* 7:1, 151–87.

Tillman, J., and N. Blasgen. 2005. Case Study of Netscaler. Paper written for CRD 199 Special Study course, UC Davis (June 16).

Torrisi, S. 2002. 'Software Clusters in Emerging Regions,' Working Paper, University of Camerino.

Endnotes

[1] Heeks (1996), p.93, Nasscom (2006), p.145, IT Workforce Data Project: Report I, p.3, *http://www.bls.gov/oes/current/oes_nat.htm#b15-0000*, downloaded 9/21/06.
[22] When the consent decree was lifted in 1991, IBM immediately created an IT consulting group. Within five years, it had annual revenue of $11 billion (McKenna, 2006, p.23).
[3] Modularization is the conversion of a component of the production process with one or more proprietary inputs, design or fulfillment techniques into a component with standardized inputs, design and fulfillment techniques.
[4] *www.idc.com/getdoc.jsp?containerID=202388*
[5] By contrast, in India, only 15–20% of the work is estimated to be done by MNEs, although it was different up to 1990. According to Enterprise Ireland, the official state website, *http://www.nsd.ie/htm/ssii/stat.htm*, Irish-owned companies generated about 11% of software exports, the rest coming from MNEs (data for 2002).
[6] These included Datamatics (a joint venture between the U.S. minicomputer maker, Wang, and ex-exployees of TCS), Digital, and Data General.
[7] According to Naidu (2002), Texas Instruments' decision to enter India was conditional on the state providing adequate power and telecommunications bandwidth.
[8] By 1988, 10% of the Indian software industry's labor force was located in India; this had risen to 41% by 2000 and 71% by 2004 (Nasscom, 1999; Nasscom, 2002, p.28; and Nasscom, 2005, p.58).
[9] Import tariff reduction was a key feature of the 1990s reforms. These had risen to 110% by 1991 but were reduced to 85% in 1993, 20% in 1994 for applications software and 65% for systems software and, in 1995, to 10% for all software (Heeks, 1996, p.49). Duties on hardware ranged from 40% to 55% in 1995, but by 2000 had come down to 15% for finished goods, such as computers, and 0% for components (microprocessors, storage devices, IC s and subassemblies, display screens and tubes, etc) (Ministry of Finance, 2000).
[10] Unfortunately, data on employment in foreign firms is not available, preventing proof of a direct causality. In 2001, the only year for which data is available, foreign firms employed 13% of the workforce (Nasscom, 2002).
[11] Sources: Nasscom (2006), p.47, and Torrisi (2002, pp.9 and 18). Torrisi's data are extrapolated for Israeli exports in 2005 and may not be entirely accurate.
[12] Based on Dossani and Manwani (2005).
[13] Information based on case study compiled by Shah (2005).
[14] The discussion of Netscaler draws upon a case study done by Joshua I. Tillman and Nicholas W. Blasgen (2005) and an interview of CEO B.V.Jagadeesh with Martin Kenney.
[15] Based on Dixit (2005) and Rafiq Dossani's interview with Dixit on September 20, 2006.

FIGURES AND TABLES

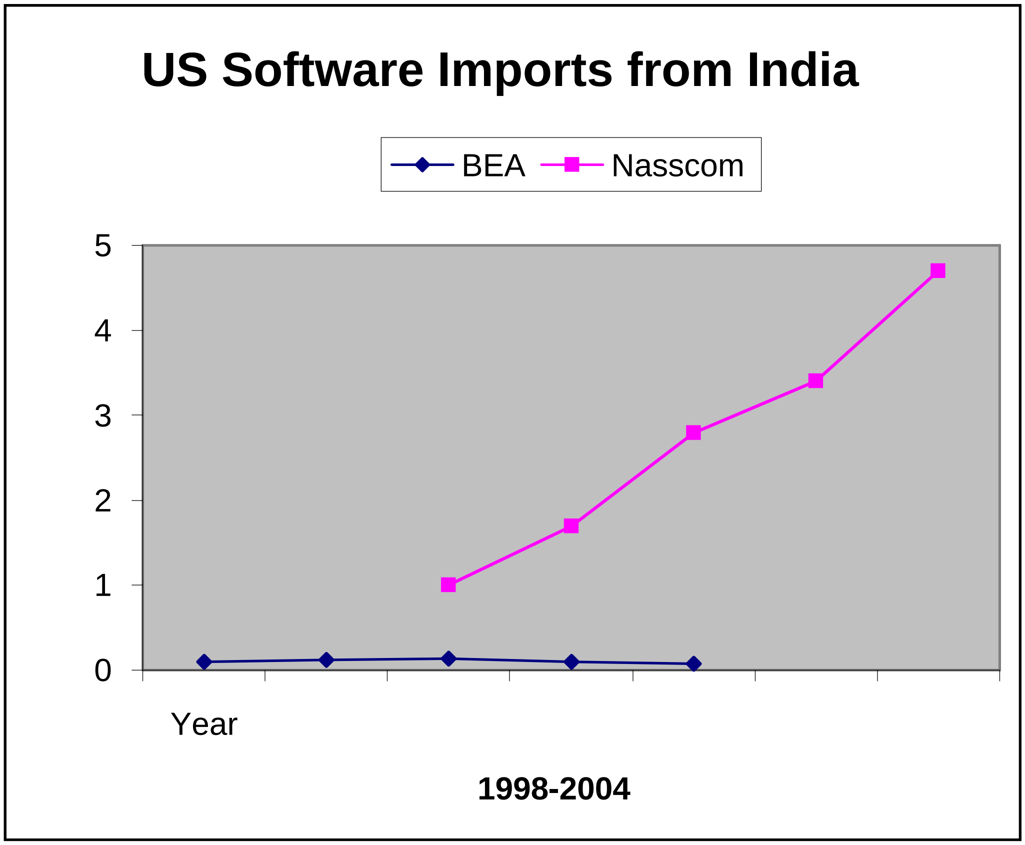

FIGURE 1 BEA and Nasscom figures for software sales from India to the US ($ bn).

Sources: *www.bea.gov* and Nasscom (various years).

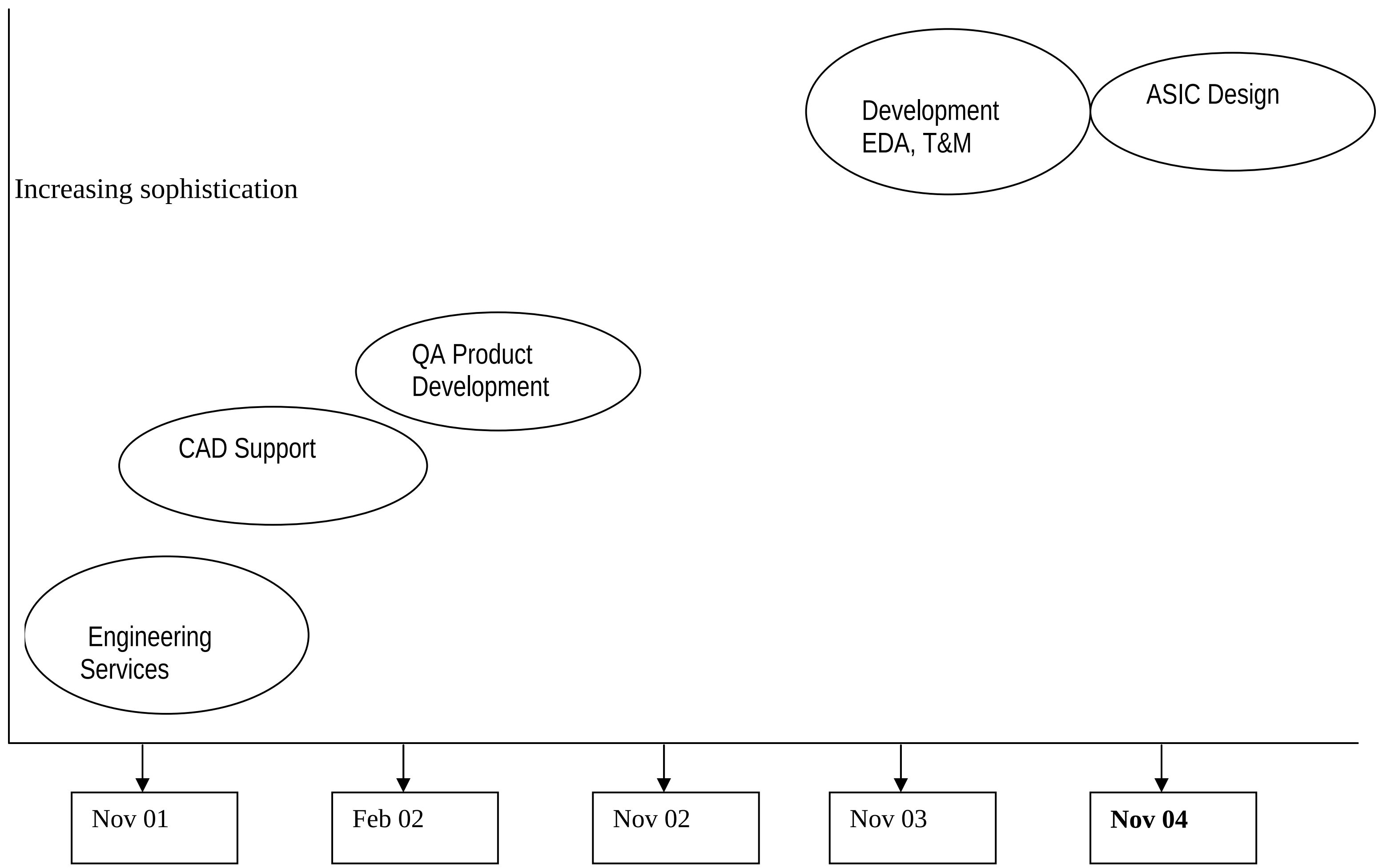

FIGURE 2 Activity transfer to Agilent's Indian operation by date.

Source: Dossani and Manwani (2005).

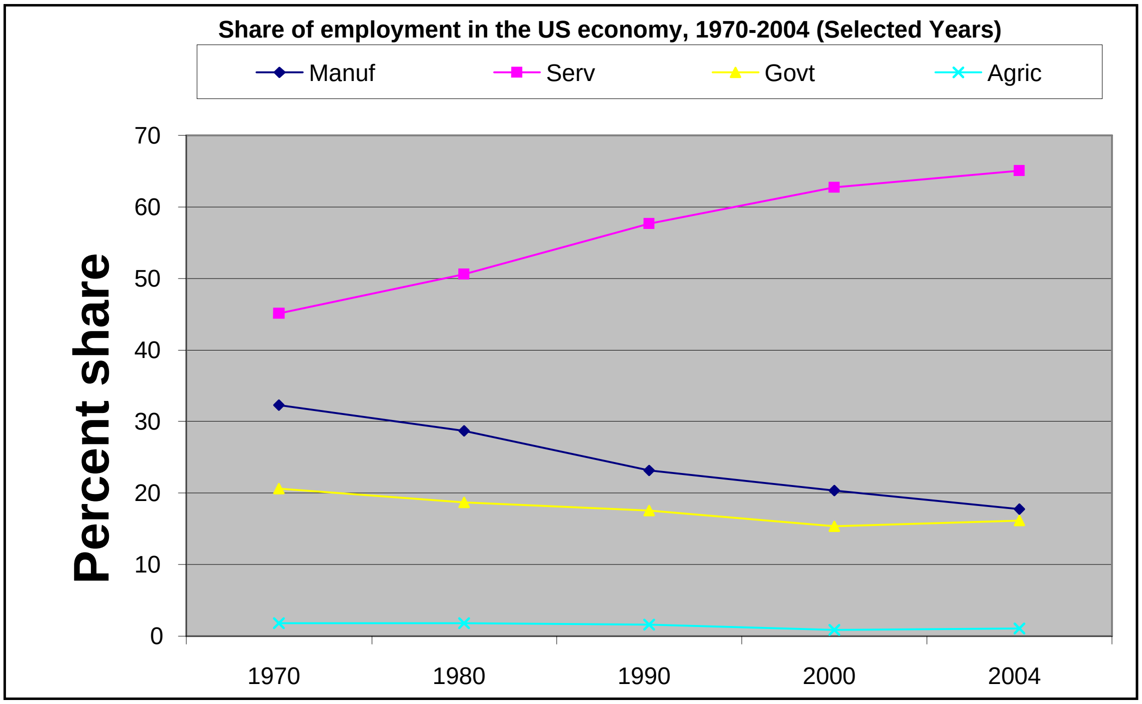

FIGURE 3

Source : BEA Statistics—*http://www.bea.gov/bea/dn/nipaweb/*, Table 6.5, downloaded 10/6/05.

TABLE 1 Software Types and Programs Used

	Product software used by:	**Custom software used by:**
Operating System	All users	None
Tools	Most users	Some users
Applications	Small and large users	Large users

Source: Authors' compilation

TABLE 2 Global Software Services Spending by Categories of Work, 2003, and India's Market Share

	Global software services spending ($ bn)	**India's global market share (%)**	**U.S. Wage rate ($/hour)**
Consulting	41.5	< 1	80-120
Applications Development	18.4	16.4	25
System Integration: Hardware and Software Deployment and Support	91.7	< 1	18-25
System Integration: Applications, tools and O/S	62.4	< 1	40
IT education and training	18.5	0	40
Managed services	124.9	1.6	60-120
Total	357.6		

Sources: Nasscom (2004, p.19, 36, 106) for columns 2 and 3; Nasscom (2001, p.24) and author's interviews for column 4.

Definitions: Consulting refers to IT strategy, system conceptualization, architecture and design. It is comprised of Nasscom numbers for IS consulting and network consulting and integration.

Applications Development refers to creating the applications programs. It is comprised on Nasscom numbers for custom applications development.

Systems integration: Hardware and software deployment and support refers to making the software and hardware components compatible and interoperable. It is comprised of Nasscom numbers for (1) Hardware Deployment and Support and (2) Software Deployment and Support.

Systems integration: Applications, tools and O/S refers to integration of the software components (both products and custom software) in a software project.

Managed services refers to services such as managing applications either onsite or remotely over the Web, managing networks, etc. It is comprised of Nasscom numbers for applications management, IS outsourcing, network and desktop outsourcing, applications service providers and system infrastructure service providers.

TABLE 3 Global Software Products Spending by Categories of Work, 2004, and Israel's Market Share

Revenue Category	Global software products spending ($ bn)	Israel's global market share (%)
System and tools software	$93.7	1.1
Application software	$120.0	1.3
Total	$213.7	1.2

Sources: and *http://www.siia.net/software/resources.asp#stats* and *http://www.iash.org.il/content/SoftwareInds/IsraeliSectors.asp*. Israel share of global markets estimated from 2000 data for Israel and comparable data for the US for2001. The United States' global market share was estimated at 41% in 2005 by the Software Information and Industry Association (www.siia.net). This percentage is used in the table above.

TABLE 4 Client-Vendor Grid During 1970–79

Clients' options =>	External data processing and managed services (A)	Client owns hardware			
		Develop and maintain own software (B)	Buy bundled software and outsource maintenance services (C)	Buy software products from ISVs (D)	Buy custom software services (E)
ISVs' offerings =>	Managed services , EDP	No role for ISVs	Integration of hardware and software; software maintenance	System level and applications products	Custom applications software

Source: Author's compilation, based on Steinmuller (1996).

TABLE 5 Software Industry New Work-Types and Causes: United States, India, and Israel

	U.S. ISV new work type	**Market change**	**Technological change**	**India ISV new work type**	**Israel ISV new work type**
1960-70	Software maintenance, EDP		Minicomputer	EDP	MNE inhouse product support
1971-80	Custom software for applications	IBM separates software and hardware		Export of programmers, primarily for custom software	No change
1981-90	Software system integration	Growing complexity of IT systems	U-W standard	Custom software for applications	Custom software for domestic markets
1991-2004	Managed services		Internet, database platforms	Managed services, R&D and product development	Product software for global markets; MNE inhouse R&D

Source: Author's compilation, based on Steinmuller (1996), Mowery (1996), and *http://www.siia.net/software/resources.asp#stats* for columns 1 to 4. Author's analysis for columns 5 to 6.

TABLE 6 Software Exports from Developing Countries, 2001

Country	Sales ($ bn)	Exports	Domestic	Sales per employee ($,000)	Remarks
Brazil	7.7	0.1	7.6	49	P/S=40/60
China	7.4	0.4	7.0	40	
EE5	0.6	0.5	0.1	8	Exports services primarily to W.Europe
India	8.2	6.2	2.0	23	Exports services primarily to US
Ireland	7.7	6.5	1.1	160	Exports primarily localized US TNC product software to W.Europe
Israel	3.7	2.6	1.1	106	P/S=70/30
Japan	85.0	0.07	84.3	159	P/S=25/75
Philippines	0.2	0.15	0.05	12	Exports services primarily to US
Russia	0.2	0.1	0.1	13	P/S=30/70
USA	200.0	NA	NA	192	P/S=40/60

Notes: P/S = product/services ratio; EE5 = Bulgaria, Czech Republic, Hungary, Poland, and Romania; U.S. data for 2002, Israel data for 2000.

Sources: Arora et.al. (2005), p.45, 77, 101; Sahay, p.17.

TABLE 7 Software Exports from India, Ireland, and Israel ($m)

	India	**Ireland**	**Israel**
1990	105	2,132	90
2000	6,200	8,865	2,600
2002	7,500	12,192	3,000
2003	8,600	11,819	3,000
2005	17,100	18,631	3,000
Employment –2003	260,000	23,930	15,000
Revenue/employee – 2003	33,076	493,988*	273,000
Employment – 2005	513,000	24,000	NA
Revenue/employee – 2005	33,333	776,000*	NA

Source: Indian data from Heeks (1995), Nasscom (2003–2006); Ireland data from *http://www.nsd.ie/htm/ssii/stat.htm*, downloaded September 26, 2006; Israel data from *http://www.iash.org.il/Content/SoftwareInds/SoftwareInds.asp*, downloaded August 31, 2003, and *http://www.israel21c.org/bin/en.jsp?enDispWho=InThePress&enPage=BlankPage&enDisplay=view&enDispWhat=Zone&enZone=InThePress&Date=08/11/05*, downloaded September 26, 2006. Irish data in Euro converted at 1 Euro = $1.043 (rate on January 5, 2003) for years prior to 2003; and then at 1.26 for 2003 onwards (rate in January 2004 was 1.26; for September 2006, the rate was 1.27). Israel's latest figures are for 2001.

*Note: Sands (2005, p.45) argues that the Ireland figures for revenue per employee are overstated due to accounting for in-country transfers and are in reality about $160,000. If so, the figures above overstate total exports by a factor of three times.

TABLE 8 Growth of the Indian Software Industry

Year	**Total exports ($m)**	**No. of firms**	**Average revenue per firm ($)**	**Average revenue per employee ($)**	**Exports/ Total Revenue (%)**
1980	4.0	21	190,476	16,000	50
1984	25.3	35	722,857	18,741	50
1990	105.4	700	150,571	16,215	N/A
2000	5287	816	7,598,039	32,635	71.8
2004	12200	3170	7,003,154	35,362	73.9

Notes:
1. Data for 1980, 1984, and 1990 are from Heeks (1996), pp.72, 73, 87, 88.
2. Data for 2000 (financial year ended March 2001) are from Nasscom (2002) and Nasscome (2004), p. 23, 26, 64.
3. Data for 2004 (f.y. ended March 2005) are from Nasscom (2005), p.75–6. 2004 data for number of firms and average revenue is based on software, software services, and IT-enabled services because disaggregated data is not available.
4. No. of employees for 1980, 1984, 1990, 2000, and 2004 were 250, 1350, 6500, 162,000, 260,000, and 345,000. respectively. Source: Heeks (1996), p.93 for 1980–1990 data and Nasscom (2004, 2005) for 2000 and 2004 data.

TABLE 9 Top 8 Indian Software Exporters

Rank	1980, India HQ	1990, India HQ	2004,India HQ	Founder, education, experience
1st	TCS – Mumbai	TCS – Mumbai	TCS – Mumbai	Kanodia (MIT)
2nd	Tata Infotech – Mumbai	Tata Infotech – Mumbai	Infosys – Bangalore	Murthy (U. Mysore, IIT Kanpur)
rd	Computronics – Mumbai	Citibank – Mumbai	Wipro – Bangalore	Premji (Stanford) and Soota (IISc)
4th	Shaw Wallace – Kolkata	Datamatics – Mumbai	Satyam – Hyderabad	Raju (Loyola College, Chennai; Ohio U)
5th	Hinditron – Mumbai	TI – Bangalore	HCL – Delhi	Nadar (PSG College, Coimbatore)
6th	Indicos Systems – Mumbai	DEIL – Mumbai	PCS – Mumbai	Patni (MIT)
7th	ORG – Mumbai	PCS – Mumbai	i-Flex – Mumbai	Hukku (BITS, Pilani) (TCS, Citicorp)
8th	Systime – Mumbai	Mahindra-BT – Mumbai	Mahindra-BT – Mumbai	Mahindra (Harvard)
Market share of top 8 firms (%)	90	65	38	

Sources: Heeks (1996), p.89 for columns 2 and 3; Nasscom (2005), p.76 for column 4; company websites and author's interviews for column 5.

Note: IBM is probably in the top 8 firms in 2004 (it was ranked 6th in 2002, but has withheld permission for its name to be displayed in subsequent Nasscom rankings: *http://www.nasscom.org/artdisplay.asp?art_id=4413#top20*, downloaded August 26, 2005)

TABLE 10 Share of Foreign-Firms' Revenue and Share of Custom Programming and Applications Management Work in Indian Software Exports

Financial yr ->	2001	2002	2003	2004	2005	2006 (E)
CAD and AM ($B)	3.65	4.40	4.87	5.98	7.67	10.16
Total software exports ($B)	5.3	6.16	7.1	9.8	13.1	17.1
Share of CAD/AM (%)	68.9	71.4	68.6	61.0	58.5	59.6
Share of foreign firms' revenue (%)	14.5	22	26	31	31	NA

Sources: Nasscom (2006), p.47, 59, 60, 70; Nasscom (2005) p.50, 51; Nasscom (2004), p.36, 40; Nasscom (2003), p.39; Nasscom (2002), p.29, 30.

Notes: CAD = Custom application development
AM = Applications management

TABLE 11 Stages of Software Offshoring by U.S. Firms to India

Firm	Work	Initial stage onshore	Offshore Stage 1	Reason for Stage 1*	Offshore Stage 2	Reason for Stage 2
Agilent	Embedded Software	Inhouse	Inhouse; Non-integral	Control	Inhouse; Integral	Stabilized coordination in Stage 1
Broadcom	Product Devt	Inhouse	Inhouse; Integral	Scale		
Hellosoft	IP Devt	Offshore operations from the start	Integral			
Ketera	Software Devt	Inhouse	Outsource; Non-integral		Inhouse; Integral	To improve coordination; to solve labor quality issues
Netscaler	Product Extension	Inhouse	Outsource Non-Integral		Inhouse Integral	To undertake more complex product devt.
Tensilica	Product Devt	Inhouse	Outsource; Non-integral	Speedy ramp-up	Inhouse; non-integral	To improve coordination; to solve labor quality issues

*(in addition to labor cost arbitrage).

TABLE 12 Manufacturing Employment in the U.S. Economy (selected years)

	1970	1980	1990	2004
Employment in manufacturing (m)	18.9	19.8	18.7	14.1

Source: BLS statistics – *http://www.bls.gov*, downloaded 10/6/05.