

The Offshoring of Engineering: Facts, Unknowns, and Potential Implications

Committee on the Offshoring of Engineering

ISBN: 0-309-11484-5, 240 pages, 8 1/2 x 11, (2008)

This PDF is available from the National Academies Press at:

<http://www.nap.edu/catalog/12067.html>

Visit the [National Academies Press](http://www.nap.edu) online, the authoritative source for all books from the [National Academy of Sciences](http://www.nap.edu), the [National Academy of Engineering](http://www.nap.edu), the [Institute of Medicine](http://www.nap.edu), and the [National Research Council](http://www.nap.edu):

- Download hundreds of free books in PDF
- Read thousands of books online for free
- Explore our innovative research tools – try the “[Research Dashboard](#)” now!
- [Sign up](#) to be notified when new books are published
- Purchase printed books and selected PDF files

Thank you for downloading this PDF. If you have comments, questions or just want more information about the books published by the National Academies Press, you may contact our customer service department toll-free at 888-624-8373, [visit us online](#), or send an email to feedback@nap.edu.

This book plus thousands more are available at <http://www.nap.edu>.

Copyright © National Academy of Sciences. All rights reserved.

Unless otherwise indicated, all materials in this PDF File are copyrighted by the National Academy of Sciences. Distribution, posting, or copying is strictly prohibited without written permission of the National Academies Press. [Request reprint permission for this book.](#)

Implications of Globalization for Software Engineering

Rafiq Dossani
Stanford University

and
Martin Kenney

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

ABSTRACT

The offshoring of software engineering, which is more than three decades old, has been at the leading edge of the offshoring of information-technology services. Over the past decade, the pace of offshoring has increased dramatically. This has been due in large part to new communications technologies and the emergence of India as an offshore location. This report describes the evolution of the globalizing software supply chain. We predict that higher value-added work will be an increasing component of offshored software and discuss its implications for employment and innovation in developed countries.

INTRODUCTION

By the end of 2005, 2.9 million people (2.2 percent of the U.S. workforce) were employed in the software industry. The annual growth rate was 7 percent over the previous decade, well ahead of average workforce growth of 1 percent.¹ The Bureau of Labor Statistics (BLS) predicts that the software industry will be among the fastest growing employers in the coming years. Six of the 20 most rapidly growing jobs from 2004 to 2014 are likely to be in high-value software work, including network systems, data-communications analysis and administration, software applications, and systems engineering.

The significant exception to high growth within software

is programming, where employment decreased from 570,000 in 1995 to about 450,000 persons in 2005. Programming requires less training than some other software work, and programmers, on average, earn less than software engineers and computer scientists (Table 3). Whereas software engineers and computer scientists should see job growth of over 45 percent and 27 percent respectively between 2004 and 2014, the Bureau of Labor Statistics forecasts less than 5 percent job growth for computer programmers. This rate is below even the economy's average job growth.

This reflects two trends. First, much routine programming is now automated. This has both reduced the programmers' share of work in software creation and increased the average sophistication of the work. Second, even as this has happened, the growth of online collaboration via the Internet and higher capacities at lower costs offshore has increased the offshoreability of programming.

This may be seen from the following information on India, which is now the largest exporter of software after the United States, accounting for 60 percent of non-U.S. software exports. Programming accounts for 60 percent of Indian software exports, down from 90 percent in 1995. Programming is, of course, not a stand-alone function. The work done by the Indian software industry is part of a supply chain, with most of the components still being fulfilled in the developed world.

Indian software employment has grown by 35 percent per annum over the past decade. Software-exporting firms located in India employed 706,000 people in 2006, up from 513,000 in 2005. In 1995, the comparable numbers for the Indian and American software industry were 27,500 and 1.5 million (1.3 percent of the U.S. workforce of 118 million).

¹Data for this section is from Bureau of Labor Statistics <http://www.bls.gov/oco/oco1002.htm> and http://www.bls.gov/oes/current/oes_nat.htm#b15-0000; GAO, 2005; Heeks, 1996; Nasscom, 2006; Ellis and Lowell, 1999.

Two-thirds of India's software exports are to the United States, a share that has remained nearly steady over the past decade.

The impact is perhaps better appreciated by calculating the Indian share of employment within the American supply chain of software. The share of Indian employment has risen from 3 percent of the programmer pool used in American software production in 1995 to over 30 percent in 2005.²

Meanwhile, work besides programming has also been offshored. Some of this newer work is even lower-end work than programming, such as installation of software and maintenance of software programs. This has happened largely because of the Internet. However, as will be shown below, new tasks, hitherto considered both difficult to offshore and high value-added relative to the programming function, such as product development and contract R&D for the software industry, have been offshored over the past decade, particularly to India. For example, as of 2006, the world's largest contract R&D firm in software, employing 14,000 persons, is the Indian firm, Wipro. A decade ago, Wipro, like others in the Indian software industry, did not do such work.

This paper fulfills two objectives. First, it explains the genesis of software offshoring. This includes a consideration of why programming was the function that was most commonly offshored right from the earliest stages. Second, it examines the scope for offshoring software work other than programming. This includes a consideration of whether the additional scope is higher or lower value-added, how it is linked to the earlier phase of programming offshoring, and its likely evolutionary trajectory.

The paper proceeds as follows: in the next section, we discuss the current status of the debate on software offshoring. The following section provides a historical overview of developments that led to offshoring in the software industry, with a focus on developments in India. This is followed by a theoretical framework for analyzing how skilled work may be offshored. We conclude with a discussion of the impact of software offshoring on employment and innovation in the United States and other developed countries, and the implications for policy on education.

THE CURRENT STATE OF THE DEBATE

A lively discussion is under way about the impact of globalization on employment and productivity in the American software industry. An assessment published in 2006 by the Association for Computing Machinery (ACM) 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 offshor-

²This has happened even as the number of programmers in the total software pool has stayed relatively steady (rising from just under 600,000 in 1995 to 650,000 in 2005), while declining in share of software employment from 38 percent to 21 percent.

ing software and software-service work to . . . low-wage countries." The report concludes that "the globalization of, and offshoring within the software industry will continue and, in fact, increase" (ACM, 2006).

As Bhagwati et al. (2004) and Mankiw and Swagel (2006) have pointed out, the offshoreability of the software industry means, first, that software services are now tradable, whereas in the past they were not. Second, given that international trade is usually beneficial to both trading partners, they conclude, ipso facto, that globalization will have positive implications for the U.S. economy. They argue that workers in the services sector of developed nations will shift to jobs in which they have a comparative advantage, thus ensuring full employment in the long run. As Mankiw and Swagel (2006) note, "Economists see outsourcing³ as simply a new form of international trade, which as usual creates winners and losers, but involves gains to overall productivity and incomes." By contrast, Samuelson (2004) has cautioned that these gains may largely be captured by developing countries; and Gomory and Baumol (2000) have argued that nationally located high-growth industries are important for national growth because of their spillover effects on overall productivity.

To some, these latter cautions suggest potentially dramatic negative impacts for software-related employment in developed countries. These argue that if software development overseas increases in quantity and, especially in scope, to include the most highly skilled work, the result may be unemployment, even for the most highly skilled software engineers in developed countries (Hira and Hira, 2005). The ACM report and other evidence points to the fact that higher skilled work is already being moved offshore in some fields of software, such as computing research (ACM, 2006; Dossani, 2006; Sridharan, 2004).

There is no comprehensive empirical evidence on software offshoring, primarily because of the poor quality of primary data. See Figure 1 for an example of contradictory data reported by the U.S. Bureau of Economic Analysis and the Indian software industry association, Nasscom. As far as we can tell, there is no systematic evidence yet of significant losses of high-value jobs in the United States to services offshoring. As noted in NAPA (2005), "The number of jobs impacted (by services offshoring in general) appears relatively small, when compared to total annual job losses in the United States."

Other empirical studies offer indirect evidence in support of the NAPA findings. For example, Mann (2006) shows that the elasticity of demand for U.S. exports of services is lower than for U.S. imports of services. If this finding is applicable to software, it would imply that globalization could have positive implications for the U.S. balance of payments.

Landefeld and Mataloni (2004) show that the share of

³Technically, the correct term is "offshoring."

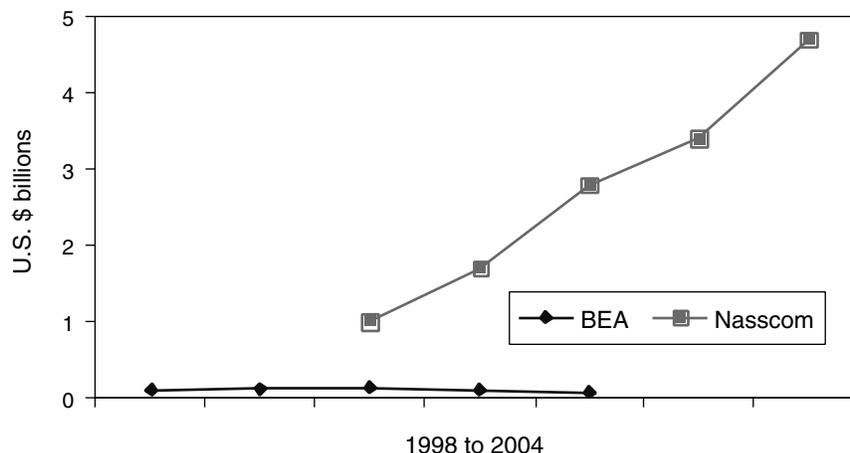


FIGURE 1 BEA and Nasscom figures for software sales from India to the United States (\$ billions). Sources: *www.bea.gov* and Nasscom (various years).

imports from subsidiaries of U.S.-based multinationals to the parent country (as a percentage of sales) did not increase from 1997 to 2001. They also find that job creation by the expansion of multinationals overseas is no different from overall job creation. Both findings imply that multinationals that offshore work to their subsidiaries are not responsible for job losses in the United States. Of course, the destination for offshored work might be unaffiliated firms, for which these data have no implications.

According to Hanson et al. (2001), the evidence of offshoring of manufacturing has shown a positive, complementary effect on American jobs from high-value offshoring and a negative, substitution effect from low-value offshoring. In the software industry, the lower value work consists of programming and the higher value work consists of design, consulting, system integration and managed services (Table 3). Hanson's findings—if applicable intra-sectorally to software—imply that the export of low-end work, such as programming, could reduce industry jobs. As Table 2 shows, this is the field with the highest market share in India, suggesting by extrapolation that job losses in the United States may indeed occur as a result.

This kind of indirect evidence has obvious limitations. Quite simply, we do not know if it is applicable to software. From our interviews with firms that have offshored work, we learned that fulfillment of various aspects of software development can be accomplished in spatially distant locations. Many of these firms state that they will increasingly shift their operations to lower cost countries like India and China. This suggests that the logic for software development at any particular location may be being eroded. The data we provided in the introduction on the programming function may be only the first wave of software offshoring.

This does not, however, mean either that the most skilled work will shift from the United States or that American software employment will decline. More than one type of outcome is possible. First, the capacity of other countries may be constrained by the quality of their educational systems or other factors that hinder labor supply; by their infrastructure, such as telecommunications; or by institutional barriers, such as weak intellectual property laws. Second, the history of technological change suggests that new opportunities will emerge. The software industry in the United States might discover higher value-added opportunities, even as existing operations are increasingly offshored. In an era of high rates of technological change, both offshore and domestic software work can become more highly skilled. Third, assuming that the first and second outcomes are both true, developed countries other than the United States could capture the new opportunities. This possibility is not investigated in this paper.

The actual outcomes of offshoring will, therefore, depend on the evolving capabilities of developed countries vis-à-vis the capabilities of developing countries. New opportunities will depend on the pace and location of innovation, which could be affected by the development of clusters of technical excellence offshore, such as those in Bangalore, Beijing, and Shanghai. Or, perhaps, as Apte and Mason (1995) and many others have argued, the need for proximity to consumers to determine their needs will be the determining factor in the location of innovation. Perhaps the open economy and excellent educational system in the United States will enable American firms to innovate at a pace that keeps them ahead of China and India. If so, the American software industry, though possibly not specific groups of workers, may thrive by keeping the innovative, highest value-added work onshore and offshoring the rest.

Predicting the outcome of offshoring requires an understanding of (1) the software industry and its evolving supply chain and (2) the ecosystem for innovation in the United States vis-à-vis other countries. To simplify our task, we have focused on India and the United States. Data on other countries are used primarily to illustrate the challenges and opportunities in these two countries. We have chosen India as the alternative to the United States for the following reasons: first, because of its position as the largest exporter of software after the United States; second, it has the size of labor force that can pose the most significant threat to U.S. employment; third, its current stage of overall economic development is likely to keep labor costs low for several years, thus adding to its attractiveness as an offshore software destination; and, finally, because as our case studies, presented below, show, Indians have the capability of doing highly skilled work.

HISTORICAL OVERVIEW OF THE SOFTWARE INDUSTRY

Product and Custom Software

Software is usually classified either by its uses or its degree of customization. We use the North American Industry Classification System (NAICS) definitions to differentiate product software and custom software. The attributes, the size of the market, and the market shares of the two key players other than the United States, India, and Israel are summarized in Tables 1 through 3.

Types of software defined by usage are listed below:

- system-level software (i.e., programs that manage the internal operations of the computer, such as operating-system software, driver software, virus-scan software, and utilities)
- tools software (i.e., programs that make applications work better, such as database-management software)
- applications (i.e., programs that deliver solutions to the end user, such as word-processing software, search-engine software and financial-accounting software)

We define two categories of software by their degree of customization: (1) publishers of packaged software (NAICS 5112) and (2) computer systems design and related services

TABLE 1 Uses of Product and Custom Software

	Product Software	Custom Software
Operating system	All users	None
Tools	Most users	Some users
Applications	Small and large users	Large users

TABLE 2 Global Spending on Software Products by Categories of Work and Israel’s Market Share, 2004

Revenue Category	Global Spending on Software Products (\$ billions)	Israel’s Share of the Global Product Software Market (percentage)
Systems and tools software	\$93.7	1.1
Application software	\$120.0	1.3
Total	\$213.7	1.2

Sources: U.S. and global data: <http://www.siia.net/software/resources.asp#stats>. Data for Israel <http://www.iash.org.il/content/SoftwareInds/IsraeliSectors.asp>. Israel’s share of global markets are estimated from data for Israel for 2000 and comparable data for the United States for 2001.

(NAICS 5415). Software publishers such as Microsoft fit under the NAICS 5112 description of publishers of packaged software, “establishments primarily engaged in computer software publishing or publishing and reproduction. Establishments in this industry carry out operations necessary for producing and distributing computer software, such as designing, providing documentation, assisting in installation, and providing support services to software purchasers. These establishments may design, develop, and publish, or publish only.”⁴ Similar in some respects to mass manufacturers, enterprises in this category create software products or packages for the general consumer market and capitalize on economies of scale. Software products may be shrink-wrapped and transported physically or made available for downloading over the Internet.

The second category, computer systems design and related services (NAICS 54151), comprises “establishments primarily engaged in providing expertise in the field of information technologies through one or more of the following activities: (1) writing, modifying, testing, and supporting software to meet the needs of a particular customer; (2) planning and designing computer systems that integrate computer hardware, software, and communication technologies; (3) on-site management and operation of clients’ computer systems and/or data processing facilities; and (4) other professional and technical computer-related advice and services.”⁵

In contrast to the one-size-fits-all software products in the first category, custom software is used when no packaged software products are available, as in highly specialized processes, or to integrate disparate software products into a cohesive system. The latter process is common when large software products, such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM) suites, must be integrated into already existing enterprise systems. Custom software may be constructed by using traditional programming languages and tools or proprietary scripting or configuration languages. Because custom software is made-

⁴See <http://www.census.gov/epcd/naics02>.

⁵ibid.

TABLE 3 Spending on Global Software Services by Categories of Work and India’s Market Share, 2003

	Global Spending on Software Services (\$ billions)	India’s Global Market Share (percentage)	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 operating systems	62.4	< 1	40
IT education and training	18.5	0	40
Managed services	124.9	1.6	60–120
Total	357.4		

Definitions:

Consulting includes IT strategy, system conceptualization, information systems (IS) consulting, architecture, design, and network consulting and integration. These services require the highest level of skills, including system design and understanding of clients’ requirements.

Applications development includes creating applications programs. These require programming skills.

System integration: hardware and software deployment and support includes making software and hardware components compatible and interoperable, hardware deployment and support, and software deployment and support. The skills required vary, but are not as high-level as programming or consulting skills.

System integration: applications, tools, and operating systems includes the integration of software components (both products and custom software) in a software project. The required skills include understanding clients’ requirements and programming skills.

Managed services include managing applications either on site or remotely over the Web, managing networks, applications management, IS outsourcing, network and desktop outsourcing, applications service provision, and systems-infrastructure service provision. The skills required vary greatly.

Sources: Nasscom, 2004 (pp. 19, 36, 106) for columns 1 and 2; Nasscom, 2001 (p. 24) and authors’ interviews for column 3.

to-order, it is more geographically constrained than product software. Proximity to the stakeholder is often crucial, especially if tacit (uncodified) knowledge is involved. Thus, software products are more readily exportable than custom software.

Nearly every computer needs systems software, and the mass market provides very favorable conditions for creating systems software as packaged products. Hence, systems software is now marketed almost exclusively as packaged products. And, over time, the need for compatibility among operating systems has become a critical requirement of both enterprise and retail users; this need has increased with the advent of the Internet. As a result, a few operating systems now dominate the computing landscape and have considerable pricing power. Compared to the demand for applications software, the demand for systems software has relatively little “give” in terms of pricing. Consumers of systems software, such as high-availability server-operating systems and real-time embedded operating systems, are willing to pay high prices for quality and interoperability. Consequently, the producers of systems software are less sensitive to production costs than product quality and the need for people with highly specialized skills.

Although product software is designed to meet a wide range of customer requirements, it can incorporate only a limited number of variations. Beyond this limit, software must be written to a customer’s specifications. Industries such as banking, in which customer requirements vary significantly, need custom software. In general, the more varied the needs of different end-users, the more likely software is to be customized. And, because needs vary most at the applications stage, most customized software is applications

software. Table 1 compares the uses of product and custom software.

The United States is the market leader in software product development, accounting for 41 percent of the total.⁶ The U.S. share of exported software products is probably even higher because many countries only produce software products for protected local markets. For instance, data on Brazil and Japan (Table 6) show that while Brazil’s annual output of product software earns revenue of about \$3 billion and Japan’s annual output earns about \$21 billion, these products are only available to domestic markets. Western Europe and Israel, like the United States, develop product software for global markets.

Custom software is part of a larger category called software services, as defined in NAICS 54151. Software services are described by type and size in Table 3.

Independent Software Vendors

The independent software-vendor (ISV) industry was created by two events, both related to market leader IBM. First, in 1956, IBM settled a long-standing antitrust suit by the federal government by agreeing, as part of a consent decree, to stop offering computer-consulting advice (McKenna, 2006).⁷ With IBM out of the picture, leading accounting firms, such as Arthur Andersen, then began offering computer consulting services. Second, in 1969, IBM

⁶See www.siaa.net.

⁷When the consent decree was lifted in 1991, IBM immediately created an IT consulting group, which, within five years, had annual revenues of \$11 billion (McKenna, 2006).

decided to unbundle its mainframe operating system, applications software, and hardware by creating open standards. Subsequently, some end-user firms set up in-house software development and maintenance operations and some began outsourcing work. As a result, ISV businesses were created (Table 4).

The columns in Table 4 do not describe mutually exclusive choices. For example, a firm might purchase system-level software products and develop 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 in-house hardware at the beginning of the decade. Initially, firms developed their own software (B), but as hardware and software became more complex, in-house software development and management became increasingly difficult. This led to the outsourcing of system integration (C) and then system-level and applications products (D). The outsourcing of customized applications (E) was an indication that industry-specific products did not meet the needs of sophisticated users, particularly large banks (Steinmuller, 1996).

In the 1980s, the IBM PC was introduced, but within a decade, IBM had lost control of the operating system to Microsoft Windows, which combined with the Intel microprocessor (Wintel) to become a market-created standard by the late 1980s. The result was a decline in hardware prices and an increase in demand for applications. Unlike mainframes, PCs were made for individual users who relied on product software. PCs in the 1980s had neither the programming capacity nor the performance capabilities necessary for mid-sized and large enterprises. Hence PCs did not impact the custom software business. However, they did create a mass market for retail product software.

The workstation, which was introduced in the early 1980s, provided many end uses for enterprises but could also be used for stand-alone programming for mainframes. The adoption of Unix as the operating system for all computers, combined with the workstation (in short, the U-W standard), revolutionized the ISV industry. An ISV could now own a workstation made by any manufacturer and write programs for a client with a different brand of installed hardware (including a mainframe). In other words, software creation became modularized, or platform independent.⁸

With the simultaneous widespread adoption of Unix/C as the programming language, other functions of software creation, such as system architecture, design, and integration, could be done separately from programming, thus modularizing the programming component. Programming could now be done anywhere in the world by programmers whose

only raw material, apart from a workstation, was a specified software system. Programmers did not even have to know which firm's hardware a program would work on or the type of application the program would support.

The workstation also had sophisticated graphics and enough computational capacity to satisfy the needs of small enterprises, which now shifted from outsourcing data-processing services to running their own workstations. In the early 1980s, the first workstation-based local area networks were established, increasing the demand for more sophisticated software for running these networks and for applications compatible with networked users.

In the 1990s, the success of database software packages further simplified the creation of applications software. Platform independence, combined with the rise in demand for custom software by small firms, resulted in the growth of a large custom software industry.

Also in the 1990s, PCs with more computing power were able to process programs written in Unix/C, thus making them more acceptable to small enterprises. As costs for PCs fell in the mass market, PCs superseded workstations as the hardware platform for programming. Later in the decade, PC-based networks made applications accessible to many more users in an enterprise.

The spread of the Internet beginning in the mid-1990s was accelerated by declining costs for bandwidth and storage. The Internet provided a platform for networked development of software and software installation, hosting, and maintenance. At this point, data no longer had to be on servers located on the premises of an enterprise but could be housed in remote data centers. The Internet also significantly reduced the cost of collaboration among remote teams. These factors further reduced the need for the proximity of user groups or of developers and users.

With the establishment of the Internet, several new models of preparing and delivering software 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 nonproprietary code) developed by voluntary contributions of networked developers. With the exception of the Linux open-source operating-system software, which is believed to have about a one-third share of the server market (although less than 2 percent of all operating systems), the new models described above have not impacted the spatial distribution of software development.⁹

The first three columns in Table 5 show the major changes and driving forces in the software-services industry in the United States described above. The two right-hand columns show (for later reference) developments in the Indian and

⁸Modularization 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.

⁹See www.idc.com/getdoc.jsp?containerID=202388.

TABLE 4 Independent Software-Vendor Industry, 1970–1979

	External Data Processing	Clients That Own Hardware			
Clients' Options	Managed services	Develop and maintain software	Buy bundled software and outsource maintenance services	Buy software products from ISVs	Buy custom software services
ISV Services	Managed services Electronic data processing	None	Integration of hardware and software Software maintenance	Systems-level and applications products	Custom applications software
	A	B	C	D	E

Source: Adapted from Steinmuller, 1996.

Israeli software industries. Note that this table does not include information on the product-software industry.

Offshoring of Software Development

American IT firms began to offshore software development to India, Ireland, and Israel (the 3 I's) in the 1970s, about a decade after the offshoring of IT hardware manufacturing. Siwek and Furchgott-Roth (1993) argue that the lag between hardware and software offshoring was because software development, unlike hardware manufacturing, required close coordination with clients throughout the process.

A widespread knowledge of English and relatively low labor costs were common attractions of the 3 I's. Small domestic markets and the lack of domain knowledge (less so for Israel) were common disadvantages. Beginning in the 1990s, many other countries, including China, several countries in Eastern Europe, Brazil, Mexico, Russia, the Philippines, and Vietnam, began exporting software to developed countries (Table 6).

As Table 6 shows, China and Brazil sell software services mostly to their domestic markets. Ireland develops software products and services for Europe, mostly by customizing U.S. software products. This should properly be included in

the category of software services. Russia, the Philippines, and Vietnam, like India, primarily export software services. Countries in Eastern Europe and Russia export mostly to Europe. Other countries export mostly to the United States. Israel is the only significant non-American producer of software products for the U.S. and other global markets.

As Table 6 shows, the most significant producers of offshored software for global markets are India and Israel. Israel focuses on software products for the global market and India on custom software for the global market. Ireland is the largest provider of localized products and services for Europe.

Ireland

Hardware offshoring began in Ireland after policy makers offered export incentives following Ireland's entry into the EU in 1973 (Enterprise Ireland, <http://www.enterpriseireland.com>, downloaded 1/20/2007; Torrissi, 2002). Software offshoring, which began in the 1980s, followed hardware offshoring (Torrissi, 2002). The main clients initially were, and continue to be, American transnational corporations (TNCs). These use Ireland to localize their software products for European markets (Torrissi, 2002). American TNCs account

TABLE 5 New Work Types and Driving Forces in the U.S. Software-Services Industry and Their Impact on the Software Industry in India and Israel

	New Work Types in the U.S. ISV Industry		Market Change	Technology Change	New Work Types in the Indian ISV Industry	New Work Types in the Israeli ISV Industry
1960–1970	Software maintenance Electronic data processing			Mini-computers	Electronic data processing	Software maintenance exports Electronic data processing
1971–1980	Custom applications		IBM unbundles software and hardware		Programmers exported	No change
1981–1990	Software system integration		Increased complexity of applications	Unix-Workstation (U-W) standard adopted	Custom applications exports	Custom applications for domestic market
1991–2004	Managed services			Internet, database management systems, PC-based networks	Managed services, contract R&D exports	Contract R&D exports, products for global markets

Source: Adapted from Steinmuller, 1996; Mowery, 1996; and <http://www.siiia.net/software/resources.asp#stats> for columns 1–4. Columns 5 and 6 are the authors' analyses.

TABLE 6 Software Exports from Developing Countries, 2001

Country	Sales (\$ billions)	Exports (\$ billions)	Labor Force (2000)	Sales per Employee (\$)	Primary Work Type
Brazil	7.7	0.1	220	35	P/S = 40/60 ^b
China	7.4	0.4	186	40	Domestic
	<i>(15.0)^a</i>	<i>(2.0)</i>	<i>(750)</i>	<i>(20)</i>	
EE5 (Bulgaria, Czech Republic, Hungary, Poland, Romania)	0.6	0.5	75	8	Services to Western Europe
India	8.2	6.2	350	23	Services to U.S.
	<i>(22.3)^a</i>	<i>(17.1)</i>	<i>(878)</i>	<i>(25)</i>	P/S = 25/75
Ireland	7.7	6.5	24	160	Localization of U.S. product software for Western Europe
Israel (2000)	3.7	2.6	35	106	P/S = 70/30
Japan	85.0	0.07	535	159	P/S = 25/75
Philippines	0.2	0.15	0.05	12	Services to U.S.
Russia	0.2	0.1	0.1	13	P/S = 30/70
United States (2002)	200.0	n/a	2,600	77	P/S = 40/60

Notes:

^aFigures in italics are for 2005.

^bP/S = the ratio between revenue from software products and revenue from software services.

Sources: Arora and Gambardella, 2005 (pp. 45, 77, 101); Sahay et al., 2003 (p. 17); Nasscom, 2006 (pp. 46, 47).

TABLE 7 Software Exports from India, Ireland, and Israel (in \$ millions, except where otherwise noted)

	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
Number employed (2003)	260,000	23,930	15,000
Revenue/employee (2003)	33,076	493,988 ^a	273,000
Number employed (2005)	513,000	24,000	n/a
Revenue/employee (2005)	33,333	776,000 ^a	n/a

^aNote: Sands (2005, p.45) argues that the revenue/employee for Ireland is overstated because of in-country transfers and should be about \$160,000. If so, total exports in Table 7 are overstated by a factor of three.

Source: Data for India are from Heeks (1996) and Nasscom (2003–2006). Data for Ireland are from <http://www.nsd.ie/html/ssii/stat.htm>, downloaded September 26, 2006. Data for Israel are from <http://www.ias.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. Data for Ireland prior to 2003 are in euros (converted at 1 euro = \$1.043, rate on January 5, 2003). From 2003 on, data are converted at 1 euro to \$1.26, the rate in January 2004. Most recent figures for Israel are for 2001.

for about 90 percent of Ireland’s software exports (Arora and Gambardella, 2005).¹⁰ Since the 1990s, an indigenous software sector has developed in Ireland, initially providing support services for TNCs but subsequently developing

¹⁰By contrast, in India, only 15 to 20 percent of the work since 1990 is estimated to be done by TNCs. According to Enterprise Ireland, the official state website, <http://www.nsd.ie/html/ssii/stat.htm>, Irish-owned companies generated about 11 percent of software exports in 2002, with the rest coming from TNCs.

products for the European telecom and financial sectors. In 2003, the indigenous sector in Ireland employed 40 percent of the total software workforce (Sands, 2005).

Israel

As in Ireland, though a decade earlier, hardware firms were established in Israel during the 1960s first in response to export incentives.¹¹ Software TNCs followed in the early 1970s (Torrissi, 2002). These initially undertook software product maintenance and, later, R&D.

In the 1980s, domestic firms were established, funded by government research contracts. They initially provided software services to the defense industry. Key labor was drawn from the Israeli defense industry. In the 1990s, with support from global venture capitalists, security product firms were established. These offered products for global markets (Teubal, 2002, see also Table 5). TNCs currently account for about 25 percent of total employment in the Israeli IT industry and focus on R&D, but growth is being driven by local firms producing software products for export markets (Torrissi, 2002). The three largest software firms in Israel are product firms that jointly account for 60 percent of the industry’s revenue (Bresnitz, 2005).

India

From Tables 6 and 7 above, we note that the most significant increase in offshoring to global markets is in India. Unlike in Ireland and Israel, where fiscal incentives were

¹¹For example, Motorola’s first offshore manufacturing subsidiary was set up in Israel in 1964 (Ariav and Goodman, 1994; Sahay et al., 2003).

critical for private-sector entry, the software industry in India began when government policy was hostile to all private industry. State policy at that time was appropriately described as “statist, protectionist and regulatory” (Rubin, 1985). An industrial licensing regime and state-owned banks strictly regulated private-sector activity. In IT, the state was the main producer of products and services. The strategy was to create “national champion” state-owned enterprises, which were granted monopolies (Sridharan, 2004).

A key protectionist policy was the Foreign Exchange Regulation Act of 1973 (FERA-1973), under which a foreign firm could only have a minority interest (up to 40 percent) in a company operating in India. Many foreign firms, including IBM, closed their Indian operations, citing concerns about the protection of intellectual property (IP). FERA-1973 effectively closed the door to software development by TNCs in India.

Domestic firms found an innovative way to benefit from global opportunities for ISVs. Because software development could not come to India, Indian programmers were sent to developed countries. This began in 1974 when Burroughs, an American mainframe manufacturer, asked its Indian sales agent, Tata Consultancy Services, to supply programmers for installing system software for a U.S. client (Ramadorai, 2002). Other firms followed suit, including foreign firms in joint ventures with Indian firms.¹²

Initially, the exported Indian programmers worked for global IT firms. Later in the decade, as IBM gained a larger share of the total global market, end-users such as banks hired Indian firms to convert existing applications software to IBM-compatible versions.

The state remained hostile or, at best, indifferent to the software industry throughout the 1970s. Import tariffs were high (135 percent on hardware and 100 percent on software). Software was not considered an “industry,” which meant that exporters were not eligible for bank financing. Even overseas sales offices were disallowed until 1979 (Ramadorai, 2002).

Such protectionism interfered with learning and prevented Indian-based programmers from moving up the value chain. Programmers returning from overseas assignments were the main source of learning about new opportunities, but because of their short assignments overseas—typically less than a year—their learning was also limited (Ramadorai, 2002). In addition, many chose to remain overseas after completing their assignments. As a result, the software industry during its first decade was mostly limited to the recruitment of engineers.

It being easier for established private conglomerates than for small firms to navigate anti-private-sector policies, large firms became the dominant players in the industry. Mumbai,

the country’s commercial and industrial capital, became the center of the business. In 1980, five of the top eight exporters (including the top four) had large-firm pedigrees. Seven of the eight, all headquartered in Mumbai, had a 90 percent market share (Table 8).

The industry changed when the global industry adopted the U-W standard in the 1980s and, as we discussed earlier, software creation and, within it, programming were modularized. Beginning at that time, coincidentally, the state gradually abandoned its protectionist, anti-TNC stance. The New Computer Policy of 1984 (NCP-1984) reduced import tariffs on hardware and software to 60 percent; reclassified software exports as a “delicensed industry” eligible for bank financing and not subject to the intrusive licensing regime (Heeks, 1996); gave foreign firms permission to set up wholly owned, export-dedicated units; and initiated 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 encouraged TNCs to introduce new businesses and new business models. Some TNCs (e.g., Texas Instruments and Hewlett Packard) did R&D and wrote product software using cross-country teams; others (e.g., ANZ Bank and Citigroup) wrote custom software for in-house use, again using cross-country teams. Thus TNCs used approaches that had been successful in other environments, such as Ireland and Israel.

Although the initial entrants, such as Texas Instruments, persuaded the government to improve the infrastructure,¹³ TNCs still faced daunting communications costs and intrusive regulation (Parthasarathy, 2000). Thus product-focused TNCs remained small. Domestic firms (e.g., Wipro) that tried to imitate the TNC product-software model also failed because (1) the domestic markets could not supply adequate domain expertise (Athreya, 2005), and (2) there was no venture capital industry to speak of.¹⁴ By 1990, product development accounted for less than 5 percent of exports (Heeks, 1996), and, by 1999, it had only increased to 8 percent (Nasscom, 2002).

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 foreign co-vendors for program design and specification. This approach succeeded because it matched the expertise of Indian firms (programming) with the expertise of overseas vendors (client understanding, design, and integration) and because it reduced costs by keeping programmers at home—although

¹²These included Datamatics (a joint venture between Wang, the U.S. minicomputer maker, and ex-employees of TCS), Digital, and Data General.

¹³According to Naidu (2002), Texas Instruments’ decision to enter India was conditional on the state providing adequate power and telecommunications bandwidth.

¹⁴Through the 1980s, domestic venture capital was concentrated in state-run firms. Two of today’s leading IT firms, Wipro and Infosys, were both turned down by state-run venture capital firms in the 1980s.

TABLE 8 Top Eight Indian Software Exporters

Rank	Firm, HQ 1980	Firm, HQ 1990	Firm, HQ 2004	Founder, Education, Experience
1	TCS, Mumbai	TCS, Mumbai	TCS, Mumbai	Kanodia (MIT)
2	Tata Infotech, Mumbai	Tata Infotech, Mumbai	Infosys, Bangalore	Murthy (U. Mysore, IIT Kanpur)
3	Computronics, Mumbai	Citibank, Mumbai	Wipro, Bangalore	Premji (Stanford) and Soota (IISc)
4	Shaw Wallace, Kolkata	Datamatics, Mumbai	Satyam, Hyderabad	Raju (Loyola College, Chennai; Ohio U)
5	Hinditron, Mumbai	Texas Instruments, Bangalore	HCL, Delhi	Nadar (PSG College, Coimbatore)
6	Indicos Systems, Mumbai	Dell, Mumbai	PCS, Mumbai	Patni (MIT)
7	ORG, Mumbai	PCS, Mumbai	i-Flex, Mumbai	Hukku (BITS, Pilani) (TCS, Citicorp)
8	Systime, Mumbai	Mahindra-BT, Mumbai	Mahindra-BT, Mumbai	Mahindra (Harvard)
Total Market Share	90%	65%	38%	

Notes:

1. IBM was probably in the top eight firms in 2004 (it was ranked 6th in 2002), but the company has not given 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).

2. Column 5 data is for firms listed in Column 4.

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

TABLE 9 Exports of Indian Software

Year	Total Exports (\$ millions)	Number of Firms	Average Revenue per Firm (\$)	Average Revenue per Employee (\$)	Exports/Total Revenue (percentage)
1980	4.0	21	190,476	16,000	50.0
1984	25.3	35	722,857	18,741	50.0
1990	105.4	700	150,571	16,215	n/a
2000	5,287.0	816	7,598,039	32,635	71.8
2004	12,200.0	3170	7,003,154	35,362	73.9

Notes:

1. Data for 1980, 1984, and 1990 are from Heeks, 1996 (pp. 72, 73, 87, and 88).

2. Data for 2000 (financial year ended March 2001) are from Nasscom, 2002, and Nasscom, 2004 (pp. 23, 26, and 64).

3. Data for 2004 (fiscal year ended March 2005) are from Nasscom, 2005 (pp. 75–76). 2004 data for number of firms and average revenues are based on figures for software, software services, and IT-enabled services combined because disaggregated data are not available.

4. Number of employees for 1980, 1984, 1990, 2000, and 2004 was 250, 1,350, 6,500, 162,000, 260,000, and 345,000, respectively. Data for 1980–1990 are from Heeks, 1996. Data for 2000 and 2004 are from Nasscom, 2004 and 2005.

the number of personnel dispatched overseas declined slowly at first.¹⁵

Thus Indian firms gradually shifted from exporting programmers to programming outsourced custom software in India. The shift, though gradual, induced many domestic firms to enter the market. The number of software firms increased from 35 in 1984 to 700 in 1990, and the share of smaller firms also rose (Table 9).

This shift raised the standards required for physical infrastructure in India. It also marked a turning point in the role of Bangalore, where real estate was cheaper than in Mumbai,

as a center for software development. Several new firms, including Infosys and Wipro decided to locate their facilities in Bangalore (Premji, 2003). The first software technology park under NCP-1984, with a reliable supply of electricity and telecommunications bandwidth, was also located in Bangalore. Another advantage of Bangalore over competing locations was low labor costs. Unlike Mumbai and Delhi, which had histories of large firms and militant labor unions, small companies in Bangalore had relatively few problems with unions (Heitzman, 1999).

In addition, Bangalore, the capital of Karnataka, is located at the center of the four southern states, Karnataka, Tamil Nadu, Andhra Pradesh, and Kerala, which together produce 52 percent of India's engineering graduates. Bangalore's best known academic institution, the elite Indian Institute of

¹⁵By 1988, 10 percent of the Indian software industry's labor force was located in India; this had risen to 41 percent by 2000 and 71 percent by 2004 (Nasscom, 1999, 2002 [p.28], 2005 [p.58]).

Science (IIS), was established in 1909. Most IIS graduates and most research were directed toward the public sector, but some indirectly supported Bangalore's development in software. This was because the government had decided to locate several high-technology state-owned enterprises there, thus creating a trained labor force (Balasubramanyam and Balasubramanyam, 2000). However, according to some industry observers, the quality of that labor force was dubious and could meet only a small part of the software industry's needs (Ramadorai, 2002). The biggest success related to IIS, Wipro Technologies (India's third largest software exporter), was founded at IIS by a group of engineers working under Ashok Soota (Parthasarathy, 2003).

Policy reforms in the 1990s and 2000s reduced import tariffs to near zero¹⁶ and regularized foreign ownership, intellectual property protection, venture capital, stock market 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 attracted a new round of TNCs, particularly foreign outsourcers and U.S.-based start-ups, and provided new opportunities for existing firms in remote software services, such as e-mail management and remote software maintenance (Table 4).

Interestingly, TNCs initially focused on programming only, which was the approach adopted by domestic firms. The TCS remote-programming method was used for in-house product development by Texas Instruments, Agilent, Hewlett Packard, Oracle, and General Electric, as well as for services by ANZ Bank, ABN Amro Bank, Accenture, IBM, and Dell. During this phase, TNCs and foreign start-ups overwhelmingly chose Bangalore for their IT operations (Naidu, 2002).

Over time, the level of sophistication of work done in India rose. As Table 10 shows, routine programming work and maintenance accounted for 68.9 percent of total export revenue in 2001, but fell to 58.5 percent by 2005. During this period, foreign firms earned 14.5 percent of total revenues in 2001 and 31 percent in 2005. We believe that there was a causal relationship between the declining share of routine work and the entry of foreign firms doing more sophisticated work.¹⁷ Data provided by Sridharan (2004) supports this inference; he notes the presence of 230 TNCs in Bangalore

¹⁶The reduction of import tariffs was a key feature of the 1990s reforms. These tariffs had risen to 110 percent by 1991 but were reduced to 85 percent in 1993, 20 percent in 1994 for applications software and 65 percent for systems software, and to 10 percent for all software in 1995 (Heeks, 1996). Duties on hardware ranged from 40 percent to 55 percent in 1995, but by 2000 they had come down to 15 percent for finished goods, such as computers, and had been eliminated for components (microprocessors, storage devices, ICs, and subassemblies, display screens, and tubes, etc) (Indian Ministry of Finance, 2000).

¹⁷Unfortunately, data on employment in foreign firms is not available, so causality cannot be proved. In 2001, the only year for which data are available, foreign firms employed 13 percent of the workforce (Nasscom, 2002).

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

Financial Year	2001	2002	2003	2004	2005	2006 (E)
CAD and AM (\$ billions)	3.65	4.40	4.87	5.98	7.67	10.16
Total software exports (\$ billions)	5.3	6.16	7.1	9.8	13.1	17.1
Share of CAD/AM (percentage)	68.9	71.4	68.6	61.0	58.5	59.6
Share of foreign firms' revenue (percentage)	14.5	22.0	26.0	31.0	31.0	n/a

Notes: CAD = custom application development. AM = applications management.

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

employing about 25,000 engineers in R&D work by 2001 and an estimated 30 to 40 chip-design start-up firms all over India between 1999 and 2002.

Of course, several domestic firms also do high-end work. Wipro, the third largest domestic firm, with 14,000 employees, provides contract R&D services and filed 68 U.S. patents on behalf of overseas clients in 2005 (Premji, 2006).

As the share of routine programming work declined, the share of engineering services, R&D, and product development rose from 8 percent in 1999 to 23 percent in 2005 (Nasscom, 2002, 2006).

Case Studies of Software Products Offshoring

Although a comprehensive study of value-added work in offshored software development is not presented here, evidence from case studies is provided to support the sectoral shift discussed above. In this section we present some examples based on our interviews. From these descriptions, the key constraints in performing higher value-added work appear to be the recruitment and retention of qualified persons and the small size of domestic markets.

Problems with recruitment and retention derive from earlier problems with educational policy and minimal interactions between universities and industry (Parthasarathi and Joseph, 2002). Until recently, faculty at even the best engineering institutions, almost all of which are public universities, were not required to conduct research. Those who 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 indus-

try . . . the curriculum offered is outdated and does not meet the needs of the labor market” (Indian Ministry of Human Resource Development, 2001). Until very recently, nearly all of the best students migrated (Siwek and Furchtgott-Roth, 1993), although this may already be changing as opportunities at home increase.

Small domestic markets have also limited the ability of Indian engineers to move up the value chain. As Rosenberg and Mowery (1979) have argued (in a more general context), vendors become technologically sophisticated through understanding customer preferences. D’Costa (2002) has criticized the dependence of the Indian software industry on exports. He argues that international outsourcing of software, although lucrative, discouraged domestic firms from doing more complex projects at home because “excessive dependence on outsourcing limits the synergy between vibrant domestic and foreign markets.”

For purposes of this discussion, we consider software product development by two types of firms, start-ups and established firms. The former are dependent on venture capital and tend to be staffed very tightly. For start-ups, coordination costs are a large share of total costs. Established firms have sources of revenue, a more reliable labor pool, and, perhaps, an interest in establishing a base in China or India for accessing domestic markets. In consequence, established firms may use offshoring as a non-integral part of product development, for purposes such as product upgrades and second-generation product maintenance.

Both types of firms also are known to use outsourcing as a strategy rather than doing work in house, 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 in house was estimated to be an \$8 billion industry in 2005 (Nasscom, 2006), about 4 percent of the software product industry. In 2005, India was the largest participant, generating revenue of \$3.9 billion in this segment. Israel came next, with \$750 million.¹⁸

Case Study: Agilent Technologies¹⁹

Agilent Technologies, which produces test and measurement equipment, chose India as a base for software development in 2001. India offered a potential talent pool, a mature judicial system, favorable protections for intellectual property compared with other developing countries in Asia, and mature management talent. Nevertheless, because of some concerns about intellectual property protection and managerial control, the company decided to do most of the work in

house rather than outsourcing it (although some software maintenance and programming work was outsourced). To address these concerns and concerns about reversibility in the event of failure, there was a six-month overlap in staffing between the United States 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 was providing parts lists to customers worldwide and data entry for the CAD group in the United States. Over time, most support services were moved to India.

In early 2002, the second Agilent user, the communications-solutions group, established a 10-person team to automate test suites for Netexpert, one of Agilent’s projects. However, a lack of coordination between the Indian and U.S. teams led to the initial failure of this experiment. The situation improved after the time allocated for coordination was increased and a quality-enhancement program was introduced in the Indian operations. By 2005, the development and maintenance of Agilent’s EDA software products were being done jointly by multicountry teams located in both countries.

Case Study: Broadcom

Broadcom, a Silicon Valley-based fabless chip firm, acquired an India operation through the acquisition of Armedia Labs, another Silicon Valley-based company founded in 1997 to develop a single-chip (popularly, system-on-a-chip [SOC]) for high-definition TV. From its inception, Broadcom’s work in Silicon Valley was tightly integrated with work at its Bangalore subsidiary, except for market development, for which the Silicon Valley team took responsibility (Khare, 2006). All other work, such as the design and development of embedded software and libraries was shared.

When Broadcom acquired Armedia in 1999, its 25-person Indian subsidiary became Broadcom India. Broadcom subsequently 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. Employees were, as in the firm’s San Jose offices, divided into functional teams, each of which was part of a global team consisting of engineers in San Jose and Irvine, California; Israel; Andover, Massachusetts; and Singapore.

As of 2006, product development was driven by the engineering director of the project, based in San Jose, and the marketing team, based in Irvine. The team might consist at any one time of more than 100 people located in various places who travel, as needed, from one location to another. The final chip-integration design (tapeout), which may take as long as two months, is always done at one location because of the need for close coordination. Tapeout was initially done either in San Jose or in Irvine, but is increasingly being done

¹⁸Sources: Nasscom, 2006 (p. 47), and Torrissi, 2002 (pp. 9 and 18). Torrissi’s data are extrapolated for Israeli exports in 2005 and may not be entirely accurate.

¹⁹Based on Dossani and Manwani, 2005.

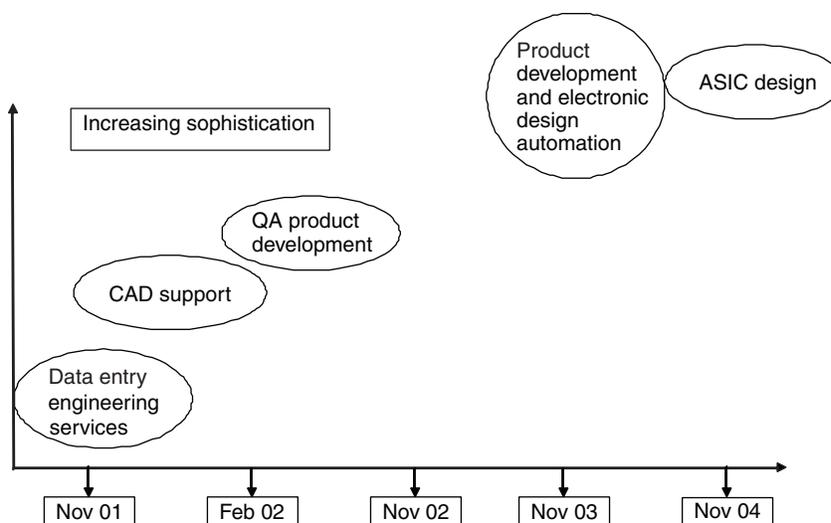


FIGURE 2 Activity transfer to Agilent's Indian operation, by date. Source: Dossani and Manwani, 2005.

in Bangalore. Early in the chip-development process, one of these three locations takes the lead.

The logic for Bangalore sharing the lead position in product development, a status not granted to other locations (such as Andover, Israel, and Singapore), is a logic of scale and capability. From 2003 to 2005, the Indian team had filed for 140 U.S. patents and been granted 10. From 2006 onward, the firm expected that the Indian team would be granted 25 to 30 patents annually. According to the CEO of Broadcom India, these numbers are comparable to U.S. patent rates (Khare, 2006).

Despite the progress of the Bangalore team, proximity still matters in some cases. Once a chip has been fully designed (after tapeout), software libraries and firmware are necessary to accommodate the specific requirements of customers, which may change considerably after the product is released. Understanding customer needs turned out to be difficult from Bangalore. Hence, in the event that the project is led by Bangalore, one member of the Bangalore team is sent to the United States for an eight-week rotation after the first release and until maturity (Khare, 2006).

The CEO also noted that the main challenges to having operations in different locations is the time it takes to establish respect among teams and to build a large enough team with the high level of skills necessary for chip development. By comparison with Silicon Valley, where putting together a 100-person skilled team of ASIC designers might take up to 18 months, putting together a similar team in India might take a good deal longer. To improve skill levels, Broadcom India recruits engineers from the United States, mostly of Indian origin, as a result of which about 5 percent of its Indian workforce is Indian expatriates. Initially, the Indian recruits were experienced engineers who were hired away

from competitors. Because of low attrition rates, however, the average work experience of engineers at Broadcom India is now more than nine years. Thus the company can now recruit from universities and offer internships to university students.

This hybrid approach has two major payoffs. First, despite the recruitment of expatriates, costs in India average one-third of costs in the United States. Second, the center of expertise is growing not only in Broadcom India, but also in Bangalore generally, in embedded software and very large chip development.

Case Study: Hellosoft

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

Case Study: Ketera Technologies²⁰

Ketera Technologies, headquartered in Santa Clara, California, provides inventory-management software on demand

²⁰Information based on a case study compiled by Shah (2005).

(i.e., software-as-a-service). As of 2005, the company had 150 employees worldwide. Its objectives for having subsidiaries in India was to cut costs and speed up time-to-market. In 2002, the company established a relationship with an Indian vendor, which had a peak of 105 workers in June 2004. The engineers in the India operations worked on software development and mundane tasks, such as configuring software for customers and other support services.

The relationship with the vendor turned out to be unsatisfactory because the engineers there were relatively unproductive and attrition rates were high. In addition, the U.S. operation was understaffed as a result of the 2001–2003 downturn. For example, there was only one architect for about 80 engineers, less than half the norm.

In late 2004, the firm created its Indian subsidiary and transferred the work in phases, beginning with software programming. The company also decided to shift its product management to India. To ease coordination problems, staff was added in the United States.

It took about nine months for Ketera to hire 75 engineers in Bangalore. Close coordination was essential to the company's success; product management was divided between the U.S. and Indian teams, with the U.S. team taking responsibility for market requirements and the Indian teams converting those into product specifications.

A key challenge in new-product development is measuring team productivity. Unlike well specified software, for which productivity can be measured by error rates or lines of code, a "new level of complexity" (Shah, 2005) is always associated with the release of a new product, which makes measuring productivity difficult.

*Case Study: Netscaler*²¹

Netscaler was founded in 1998 to redesign a specific component of infrastructure used in regulating traffic flow on the Internet. After Netscaler had developed the product, the company realized some functionality had to be added to attract customers who were wary of moving from legacy products to the Netscaler product. 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 of this contracting arrangement convinced the company to establish Netscaler India, which was staffed by many of the developers from NodeInfoTech (Tillman and Blasgen 2005). In 2004, Netscaler India employed approximately 60 engineers to develop other features and planned to grow to 200 employees by 2005 (Hindu Business Line, <http://www.thehindubusinessline.com/>, downloaded 1/13/2006). At that point, however, it was purchased by Citrix Systems for \$300 million.

²¹This discussion of Netscaler is based on Tillman and Blasgen (2005) and Jagadeesh (2006).

The reason Netscaler formed a subsidiary rather than continuing to outsource 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 U.S. 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.

*Case Study: Tensilica*²²

Tensilica is a Silicon Valley start-up established in 1997. The company, which has 120 employees worldwide, develops and licenses its embedded processor technology to SOC suppliers. The downturn of 2001 affected demand for Tensilica's products and led the firm to consider shifting second-generation work, such as adding features and improving product reliability, to India, thus freeing up expensive U.S.-based engineers for new-product development. To save on initial setup costs, and because the firm did not have a brand name in India to help recruit the best talent, Tensilica decided to begin working with a vendor, eInfochips, and then transfer to a subsidiary over time.

The initial work involved adding features to an existing product, such as improving the graphical user interface. An experiment with quality assurance was unsuccessful because it required too much U.S. management time. In general, coordination costs were much higher than expected. e-Infochips agreed to let Tensilica handle recruitment, but this turned out to be much more difficult than expected because the level of skills available was too low. In addition, some qualified engineers were unwilling to work for an outsourcer.

In January 2006, Tensilica transferred engineers from e-Infochips to its own subsidiary, which, as of September 2006, employed 15 persons, or 12 percent of Tensilica's workforce. Without the veil of an outsourcer, recruitment became much easier, and attrition rates have fallen. After working with the India team for a year, the company has also greatly reduced coordination costs. The company now does work that involves much more complexity in India.

Case Study: SAP

SAP, a large German applications software firm, began its offshoring operations to Bangalore in 2000. Initially, a CRM project was supported from India. About 40 percent of the programming work for the project was done in Bangalore. The work was done on an ad hoc basis. Project managers based in SAP's German offices would request programming

²²Based on Dixit (2005, 2006).

support from the Bangalore operations when needed, on a short-term basis.

Despite the success of this approach, SAP found that attrition rates in its Bangalore operations rose to over 30 percent. A workforce analysis revealed that its Bangalore team would have to be given more responsible and long-term work in order to induce them to stay on with SAP. The firm responded in 2003 by shifting all the programming work for selected projects to Bangalore, while retaining the management of the project in Germany. This approach enabled Bangalore-based engineers to offer all the programming support for a project through the life of the project.

While this approach led to a reduction in attrition, the coordination required to manage complete projects globally was proving to be very high. In 2004, SAP shifted the work of some project and sub-project (component) managers to Bangalore in order to ensure that engineers only reported locally. This approach proved to be so successful that, by 2006, SAP had grown to 3,200 persons in Bangalore. The Bangalore operations were given the status of a “Global Development Center” (i.e., it had achieved across-the-board capabilities to support any of SAP’s projects globally). This is a status hitherto granted by SAP only to its operations in Germany, Palo Alto in the United States, and Tel Aviv, Israel. SAP Bangalore was also designated as SAP’s center of excellence for several verticals, including oil and gas, steel and telecommunications. Attrition rates by the end of 2006 were at industry-standard rates of 12 percent.

Lessons from the Case Studies

Extrapolating from this admittedly small base of information, we found two basic models: (1) offshoring as a supplement to onshore operations (i.e., the purpose of the offshore facility is to lower costs and/or accelerate product or product-line extensions); and (2) offshore operations as an integral part of the business model. The ultimate goal in

both models is for the India business to become an integral part of the company.

Interestingly, both start-ups and established firms often begin by using an outsourced provider rather than establishing their own facilities. One advantage of outsourcing is that operations can be ramped up quickly. In addition, the company may learn about the Indian environment through the operation of the outsourcer, thereby facilitating the later establishment of a subsidiary.

There are also risks to this approach. First, as a company cedes control over the labor force to an outside vendor, it risks losing control of its intellectual property and also its ability to respond directly to attrition. Second, because the ultimate goal for both new and established firms appears to be that the India operations become integral to the business, a subsidiary must be established at some point. Integration into the company may sound like an irrevocable end point, but we have observed cases of firms that later contracted out routine in-house work. Established firms have less critical cost concerns and are, therefore, more likely to create a subsidiary and begin in-house work right away. Third, in all cases, coordination costs have been surprisingly high, not because of inadequate communications facilities, but because of the complex nature of the work. Fourth, finding and retaining qualified persons for higher value-added work is difficult, most likely because of the small size of India’s domestic markets and its inadequate educational system.

Table 11 provides a summary of the stages of offshoring described in the case studies. Undoubtedly, evolution will continue. For example, Agilent India plans to increase outsourcing once the offshoring process is stabilized.

THEORETICAL FRAMEWORK

A framework for offshoring of software services in international trade requires some definitions, some as basic as a definition of “service.” Most people agree that “manu-

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

Firm	Type of Work	Initial Stage Onshore	Offshoring Stage 1	Reason for Stage 1 Offshoring ^a	Offshoring Stage 2	Reason for Stage 2 Offshoring
Agilent	Embedded software	In house	In house, not integral	Control	In house, integral	Coordination stabilized in Stage 1
Broadcom	Chip design	In house	In house, integral	Scale		
Hellosoft	IP development	Offshoring operations from the start	Integral			
Ketera	Software-as-a-service	In house	Outsource, not integral		In house, integral	To improve coordination and resolve labor-quality issues
Netscaler	Router software	In house	Outsource, not integral		In house, integral	To undertake more complex product development
Tensilica	Embedded processor	In house	Outsource, not integral	Rapid ramp-up	In house, not integral	To improve coordination and resolve labor-quality issues
SAP	Applications development	In house	In house, not integral	Cost and scale	In house, integral	To improve coordination and resolve labor-quality issues

^ain addition to labor cost arbitrage

facturing” is a process that involves the transformation of a tangible good. Most people also agree that, in many cases, manufacturing does not require face-to-face contact between the 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 as the opposite of manufacturing in many respects. Services are transactions that involve intangible, non-storable goods, and client and vendor must be face-to-face when 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 is obviously true when the service requires customization, such as receiving a haircut, but is also true when the “service experience” does not require customization, such as when a bank client wants to check the bank’s home loan offering, or even proximity, as when a customer wants to check a bank balance.

Thus certain services are intrinsically more difficult to offshore than manufactured goods. When a service activity is considered as a totality, it indeed 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 interaction, either among coworkers or with persons outside the organization, such as vendors and clients.

Following Bhagwati’s (1985) framework, we divide services that require proximity between user and provider into three categories:

1. Mobile user-immobile provider (e.g., a cell-phone user who visits a service center for a software upgrade).
2. Immobile user-mobile provider (e.g., a software consultant who visits a client prior to designing an IT system to understand the information flows in the client’s business).
3. Mobile user-mobile provider (e.g., two delegates at a conference who exchange information through Bluetooth-enabled laptops).

For software services, the required interaction between seller and consumer has been substantially reduced. Advances in information technology have made possible the parsing of the provision of certain services into components requiring different levels of skill and interactivity. Besides the standardization of hardware and software platforms and the reduced cost of computing power, new language-structuring mechanisms, such as object orientation, have been developed. In addition, the Internet allows for the standardization of data-transmission platforms. As a result, certain portions of serviced activities—that might or might not be skill-intensive and that require little face-to-face interaction—can now be relocated offshore. Digital technology has made this possible.

The first fundamental change with digitization was that service flows could be converted into stocks of information, making it possible to store a service. For example, a consultant’s assessment that once had to be delivered to a client in person could now be prepared as a computer document and transmitted via e-mail or, better yet, encoded into software. Easy storage and transmission allowed for the physical separation of client and vendor, as well as their separation in time. In addition, services could be separated into components that were standardized and could be prepared in advance (such as a template for the assessment) and components that were customized for the client (such as the assessment itself), which were non-storable. By taking advantage of the subdivision of tasks and the economies of the division of labor, costs could be reduced by having lower cost laborers prepare the standardized components, possibly at another location.

The second fundamental change was the conversion of non-information service flows into information service flows. For example, the assessment of information-technology needs for an automobile assembly line, which had required a site visit to make the assessment, can now be made through virtualization models of the assembly line delivered over the Internet. Once converted to an information flow, the service may then be converted into a stock of information, which can reduce costs through the standardization of components and remote production.

Third, by enabling the low-cost transmission of digitized material, digitization accelerated the offshoring of services. Early on, services, such as the writing of 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. Later, in the 1990s, as the cost of digital transmission fell, even non-storable services, such as customer care, could be offshored.

The events that enabled software offshoring did not happen all at once and may not even have happened in the same way in every country. Israel, for example, was able to move quickly to product development for global markets by domestic firms. India, by contrast, until a few years ago, had offered only routine programming work for more than two decades. As of 2006, there was no evidence of successful product development that originated in India, although work to support product development conceived in developed countries was being done.

Thus moving to higher stages of work is not automatic, sequential, or time bound. Based on the available evidence, we cannot specify the conditions for movement to higher stages or predict that an exporter will capture a rising share of the economic rents (income in excess of cost).

At the very least, our case studies suggest that one factor that can hinder movement to higher stages is the cost of global coordination, whether it be between a developing country vendor and a developed country consumer or a team of vendors located across the world. For this reason, the developed-country firm can be compensated for being the middleman. Much of the market-related coordination and

networking requires developed-country institutions, enabling the capture of value by the developed-country firm. However, competition is likely to force price compression on developed-country firms, especially if it comes from developing countries. This is happening now with major Indian software services firms, which are evolving into systems integrators as they develop the requisite skills and customer confidence.

The inference is that certain aspects, such as deciding on a product and its specification, design, marketing, and sales, are usually retained by the importer. But there is no guarantee that developed-nation firms will continue to maintain this privileged position. For the time being, however, the exporter's ability to rise to new stages of growth is limited, and developed-country buyers will continue to reap the rewards.

CONCLUDING THOUGHTS

It is tempting to view software offshoring as the cause of unmitigated job losses for U.S. workers. Software offshoring raises fears that, as a result of digitization, skilled jobs will rapidly disappear from U.S. shores. This would not only leave the United States digitally divided from other countries, but would centralize demand for U.S. workers in non-offshorable jobs. In software, the argument is often made that U.S. workers will ultimately do only those jobs that are impossible to offshore, a few of which will undoubtedly be highly skilled but most of which will require lesser skills, such as information-technology training and hardware and software systems integration.

Our analysis of the software industry shows that the effects of offshoring on employment in developed nations vary, even though the impact of software offshoring on developing countries is to generate increasingly high levels of employment. The kinds of work initially offshored typically have low entry barriers and are subject to automation. Thus services exported from developing countries initially lack brand value and thus are very different from services exported from developed countries. In consequence, there is likely to be competition and price compression in these sectors.

However, over time, the level of sophistication of work being done offshore has risen rapidly. This can be a subtle process. As Shah (2005) notes in his discussion of Katera's offshoring, "The primary challenge [of offshoring most of the head count to India] was the lack of informal communication in our Silicon Valley office. We missed the informal hallway and coffee station side chats. We missed going to the white-board and brainstorming an idea." After observing the progress of the Indian operation, he concluded, "We then realized that the hallway discussions and white-board brainstorming are still happening [in our firm], but in India."

In summary, there is little doubt that work that is modularized and standardized and does not require regular customer contact is more likely to be moved offshore. This was evidenced by the rapid offshoring of the programming function. As our case studies show, the digital revolution (a catch-all

term for a series of changes) has increased the scope of work in the software supply chain that can be spatially disaggregated and outsourced. Even when a customer interface is necessary, it is possible (as the case study of Broadcom India showed) to manage customer interfaces remotely through "body-shopping" that focuses on understanding customers rather than, as in the old days, accessing customers' software and hardware. In the case of Broadcom India, offshore workers are substitutes for U.S. workers.

Lowering the costs of some aspects of software development lowers total costs and makes a company more competitive globally. It can also make possible the creation of new firms that would otherwise not be economically viable, as the case study of Netscaler showed. Jobs created by this entrepreneurship can be counted against jobs lost to offshoring. As 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 United States today than we did when we set up the India operations" (Tillman and Blasgen, 2005). In this case, offshore workers are complements to U.S. workers.

Ongoing technological development typical of the software industry can both speed up and 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 on-site technician. In most cases, these tasks can now be done remotely, thus reducing the need for on-site 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.

At the same time, the Internet enables access to many more software applications that are developed elsewhere, including open-source applications. Raza (2005) notes that chip designers who 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 (although this does not increase demand for U.S. software developers).

An alternate view of the impact of technological change is that, because the developers of new technology are mostly in 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. From this point of view, anything that helps developed-country engineers innovate more quickly and efficiently is a plus for the developed country. Hence, offshoring software development that is a step behind the work being done in the developed country enables engineers in developed nations to innovate even more and is good for both developing and developed nations.

As we noted in our introduction, scholars concede that the effects of offshoring on the quality of work done in developed nations are uncertain, because we do not know whether

the productivity gains will be captured by the developing country or the developed country. This depends on their relative productivity gains. Hence many would concede that the jobs left for workers in developed nations will certainly include low-wage work that cannot be done remotely (such as the physical installation of a hard-wired network). Many would also agree that short-term unemployment is possible. However, they also argue that most of the new work will require higher skill levels than are available in developing countries, will pay more, and will even leverage work being done in developing countries.

Based on the experience of offshoring in the manufacturing sector, a second issue is the speed with which services offshoring takes place. The decline in manufacturing in the United States happened gradually and was accompanied by rising revenue per employee, reflecting in part that, as 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 still being done onshore (Figure 3 and Table 12). The slow pace of manufacturing offshoring also gave displaced workers time to acquire skills to shift to other occupations.

As the rate of offshoring in the Indian software industry shows, some aspects of software offshoring may be rapid, leaving little time for labor-force adjustment. The reason for the rapid rate can be attributed to digitization, which has been firmly established since the mid-1990s (the Telecom Regulation Act of 1996 is often considered a turning point). Digital technology has been crucial to the rapidity of services offshoring. Unburdened by the need for large factories, off-

TABLE 12 Share of Employment in Manufacturing Employment in the United States

	1970	1980	1990	2004
Employment in manufacturing	18.9%	19.8%	18.7%	14.1%

Source: BLS statistics (<http://www.bls.gov>) accessed 10/6/05.

shored services can be set up almost as rapidly as workers with the requisite skills can be hired. Certainly the growth rate of the Indian information-technology industry has been much, much faster than in manufacturing offshoring.

This raises the question of whether the digital revolution has done more than provide a one-time boost for Asian competitors. Apart from the labor-cost advantage, developing countries will continue to have a comparative advantage for two reasons: (1) economies of scale and scope, and (2) specialization.

Countries such as India have large labor pools that could offer significant economies over smaller labor pools or country-specific labor pools. In addition, by locating software developers in India, the vendor can supply services for clients in different time zones, thus making efficient use of capital and real estate. Or, vendors can manage episodic peak requirements, such as when a new upgrade of software is released, more efficiently.

Many efficient practices for offshore software development that resulted from the remote software-programming businesses were developed in India. Thus remote management is emerging as a specialized skill that is applicable in a variety of other offshoring situations, such as providing

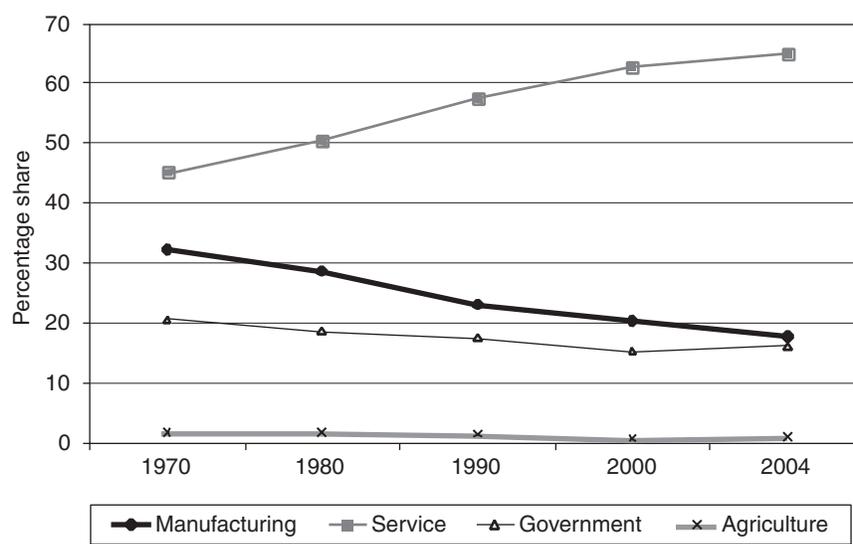


FIGURE 3 Share of employment for various economic sectors in the United States, 1970–2004. Source: BEA Statistics (<http://www.bea.gov/bea/dn/nipaweb/>) Table 6.5, accessed 10/6/05.

R&D and product-development services. Of course, Indian firms with these specialized skills must compete with the remote project-management skills developed by global firms in other environments (e.g., Accenture's skills in system integration).

At the beginning of this paper, we suggested two trajectories in offshoring that might protect employment in developed countries. The first was that constraints on capacity (both educational and infrastructural) in low-cost countries might limit the scale of offshoring. Based on the evidence we have presented, this is unlikely to happen. The second trajectory was that developed nations would reinvent themselves to a higher value-added path. It appears that the only viable strategy for developed nations is to develop the capacity to generate continuous high-value new opportunities that cannot be immediately offshored, which will require ongoing innovation. Although there is no guarantee that a developed country will have the capacity for continuous innovation, a country with an open economy that invests in education has a better chance than others. We can be hopeful that the United States will continue to demonstrate the truth of this proposition.

REFERENCES

- ACM (Association for Computing Machinery). 2006. W. Aspray, F. Mayadas and M. Vardi (eds.). *Globalization and Offshoring of Software: A Report of the ACM Job Migration Task Force*. New York: ACM.
- Apte, U., and R. Mason. 1995. Global disaggregation of information-intensive services. *Management Science* 41(7): 1250–1262.
- Ariav, G., and S. Goodman. 1994. Israel: of swords and software plowshares. *Communications of the ACM* 37(6): 17–21.
- 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, U.K.: 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 *Regions, Globalization and the Knowledge-Based Economy*, edited by J. Dunning. Oxford, U.K.: Oxford University Press.
- Bhagwati, J. 1985. Why Are Services Cheaper in the Poor Countries? In *Essays in Development Economics, Volume 1: Wealth and Poverty*, edited by J. Bhagwati and G. Grossman. Cambridge, Mass.: MIT Press.
- Bhagwati, J., A. Panagariya, and T. Srinivasan. 2004. The muddles over outsourcing. *Journal of Economic Perspectives* 18(4): 93–114.
- Bresnitz, D. 2005. The Israeli Software Industry. Pp. 72–98 in *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel*, edited by A. Arora and A. Gambardella. Oxford, U.K.: Oxford University Press.
- D'Costa, A. 2002a. Software outsourcing and development policy implications: an Indian perspective. *International Journal of Technology Management* 24(7/8): 705–723.
- Dixit, A. 2006. VP, Tensilica. Personal Interview, 9/20/06.
- Dossani, R. 2006. Globalization and the Offshoring of Services: The Case of India. Pp. 241–267 in *Offshoring White-Collar Work*, edited by S. Collins and L. Brainard. 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 Conference on Globalization of Services, Stanford University, June 2005.
- Ellis, R., and C.L. Lowell. 1999. Core Occupations of the U.S. Information Technology Workforce. IT Workforce Data Project 1.
- Gadrey, J., and F. Gallouj. 1998. The provider-customer interface in business and professional services. *Services Industries Journal* 18(2): 1–15.
- GAO (Government Accountability Office). 2005. *International Trade: U.S. and India Data on Offshoring Show Significant Differences*. Report #GAO-06-116. Washington, D.C.: U.S. Government Printing Office.
- Gomory, R., and W. Baumol. 2000. *Global Trade and Conflicting National Interests*. Cambridge, Mass.: MIT Press.
- Hanson, G., R. Mataloni Jr., and M. Slaughter. 2001. Expansion Strategies of U.S. Multinational Firms. In *Brookings Trade Forum*, edited by S. Collins and D. Rodrik. Washington, D.C.: Brookings Institution.
- Heeks, R. 1996. *India's Software Industry*. New Delhi: Sage Publications.
- Heitzman, J. 1999. Corporate strategy and planning in the science city: Bangalore as Silicon Valley. *Economic and Political Weekly*, January 30, 1999.
- Hira R., and A. Hira. 2005. *Outsourcing America: What's Behind Our National Crisis And How We Can Reclaim American Jobs*. New York: American Management Association.
- Indian Ministry of Finance. 2000. Finance Minister's Budget Speech. New Delhi: Ministry of Finance.
- Indian Ministry of Human Resource Development. 2001. *Technical Education Quality Improvement Project of the Government of India*. New Delhi: Ministry of Human Resource Development.
- Jagadeesh, B. 2006. CEO, Netscaler. Personal Interview.
- Khare, R. 2006. CEO, Broadcom, India. Personal Interview, 2/1/06.
- Landefeld, S., and R. Mataloni. 2004. *Offshore Outsourcing and Multi-national Companies*. Working Paper. Washington, D.C.: Brookings Institution.
- Mankiw, G., and P. Swagel. 2006. *The Politics and Economics of Offshore Outsourcing*. Working Paper. Washington, D.C.: American Enterprise Institute.
- 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, U.K.: Cambridge University Press.
- Mowery, D., ed. 1996. *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure*. New York: Oxford University Press.
- Mukherji, A. 2006. Senior VP, Wipro. Personal Interview, 8/1/06.
- Naidu, B. 2002. Director, Software Technology Parks of India, Bangalore. Personal Interview, 2/15/02.
- NAPA (National Academy of Public Administration). 2005. *Department of Commerce Offshoring Study*. Available online at http://www.napawash.org/pc_management_studies/index.html, accessed 5/12/07.
- Nasscom. 1999, 2001–2006. *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.
- Parthasarathy, B. 2000. *Globalization and Agglomeration in Newly Industrializing Countries: The State and the Information Technology Industry in Bangalore, India*. Ph.D. thesis, University of California, Berkeley.
- Parthasarathy, S. 2003. CEO of Aztec Software. Personal Interview, 3/2003.
- Premji, A. 2003. CEO of Wipro. Personal Interview, 12/2/2003.
- Premji, A. 2006. CEO of Wipro. Personal Interview, 2/10/2006.
- Ramadorai, S. 2002. CEO of TCS. Personal Interview, 11/29/02.
- Raza, A. 2005. CEO of Raza Foundries. Personal Interview, 12/15/2005.
- 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, U.K.: Cambridge University Press.

- Samuelson, P. 2004. Where Ricardo and Mill rebut and confirm arguments of mainstream economists supporting globalization. *Journal of Economic Perspectives* 18(3): 135–146.
- Sands, A. 2005. The Irish Software Industry. Pp. 41–71 in *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel*, edited by A. Arora and A. Gambardella. Oxford, U.K.: Oxford University Press.
- Shah, R. 2005. Kettera India Case Study. Paper presented at Conference on Globalization of Services, Stanford University, June 2005.
- Siwek, S., and H. Furchtgott-Roth. 1993. *International Trade in Computer Software*. Westport, Conn.: Quorum Books.
- Sridharan, E. 2004. *Evolving Towards Innovation?: The Recent Evolution and Future Trajectory of the Indian Software Industry*. Pp. 27–50 in *India in the Global Software Industry: Innovation, Firm Strategies and Development*, edited by A. D'costa and E. Sridharan. New York: Palgrave Macmillan.
- Steinmuller, W. 1996. The U.S. Software Industry: An Analysis and Interpretive History. Pp. 15–52 in *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure*, edited by D. Mowery. Oxford, U.K.: 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): 151–187.
- Tillman, J., and N. Blasgen. 2005. Case Study of Netscaler. Paper written for CRD 199 Special Study Course, University of California, Davis, June 16, 2005.
- Torrisi, S. 2002. *Software Clusters in Emerging Regions*. Working Paper. University of Camerino. Italy.
- Yarlagadda, K. 2005. Founder, President, and CEO, Hellosoft. Personal Interview, 2005.

BIBLIOGRAPHY

- Arora, A., and S. Athreye. 2002. The software industry and India's economic development. *Information Economics and Policy* 14(2): 253–273.
- Collins, S., and L. Brainard. 2006. *Offshoring White-Collar Work*. Washington D.C.: Brookings Institution.
- Correa, C. 1996. Strategies for software exports from developing countries. *World Development* 24(1): 171–182.
- D'Costa, A. 2000. *Technology Leapfrogging: The Software Challenge in India*. Pp. 183–200 in *Knowledge for Inclusive Development*, edited by P. Conceicao, D. Gibson, M. Heitor, G. Sirilli, and F. Veloso. Westport, Conn.: Quorum Books.
- 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 Conference on Globalization of Services, Stanford University, June 2005.
- Feller, J., B. Fitzgerald, S. Hissam, and K. Lakhani. 2005. *Perspectives on Free and Open Source Software*. Cambridge, Mass.: MIT Press.
- NSF (National Science Foundation). 1999. *Assessing the demand for information technology workers*. IT Workforce Data Project: Report 1. Washington, D.C.: NSF.
- Scholte, J. 2000. *Globalization: A Critical Introduction*. Basingstoke, U.K.: Palgrave Publishers.
- Schwartz, R. 1992. Software industry entry strategies for developing countries: a "walking on two legs" proposition. *World Development* 20(2): 143–164.